C-Nash: A Novel Ferroelectric Computing-in-Memory Architecture for Solving Mixed Strategy Nash Equilibrium

Yu Qian[1], Kai Ni[2], Thomas Kämpfe[3], Cheng Zhuo[1,4*] and Xunzhao Yin[1,4*]
[1]Zhejiang University, Hangzhou, China; [2]University of Notre Dame, Notre Dame, USA; [3]Fraunhofer IPMS, Dresden, Germany
[4]Key Laboratory of CS&AUS of Zhejiang Province, Hangzhou, China; *Corresponding author email: {czhuo, xzyin1}@zju.edu.cn

ABSTRACT

The concept of Nash equilibrium (NE), pivotal within game theory, has garnered widespread attention across numerous industries. However, verifying the existence of NE poses a significant computational challenge, classified as an NP-complete problem. Recent advancements introduced several quantum Nash solvers aimed at identifying pure strategy NE solutions (i.e., binary solutions) by integrating slack terms into the objective function, commonly referred to as slack-quadratic unconstrained binary optimization (S-QUBO). However, incorporation of slack terms into the quadratic optimization results in changes of the objective function, which may cause incorrect solutions. Furthermore, these quantum solvers only identify a limited subset of pure strategy NE solutions, and fail to address mixed strategy NE (i.e., decimal solutions), leaving many solutions undiscovered. In this work, we propose C-Nash, a novel ferroelectric computing-in-memory (CiM) architecture that can efficiently handle both pure and mixed strategy NE solutions. The proposed architecture consists of (i) a transformation method that converts quadratic optimization into a MAX-QUBO form without introducing additional slack variables, thereby avoiding objective function changes; (ii) a ferroelectric FET (FeFET) based bi-crossbar structure for storing payoff matrices and accelerating the core vector-matrix-vector (VMV) multiplications of QUBO form; (iii) A winner-takes-all (WTA) tree implementing the MAX form and a two-phase based simulated annealing (SA) logic for searching NE solutions. Evaluations show that C-Nash has up to 68.6% increase in the success rate for identifying NE solutions, finding all pure and mixed NE solutions rather than only a portion of pure NE solutions, compared to D-Wave based quantum approaches. Moreover, C-Nash boasts a reduction up to 157.9X/79.0X in time-to-solutions compared to D-Wave 2000 Q6 and D-Wave Advantage 4.1, respectively.

1 INTRODUCTION

Game theory has seen remarkable success across various applications in industries and businesses [1, 2]. Nash equilibrium (NE), as a fundamental concept within game theory, has garnered significant attention across multiple fields [3, 4], exemplified by its application to the well-known "Prisoner's Dilemma" problem [5]. NE captures a stable solution for the decision-making problem involving multiple players. Each player has a finite set of actions, aiming to find an optimal strategy to choose actions to maximize their individual payoff. At the equilibrium, each player commits to a strategy that no player can unilaterally improve their payoff by changing their chosen strategy, which represents a stable state. NE has two forms: (1) pure strategy NE , where players make deterministic choices, selecting a specific action; (2) mixed strategy NE , where players can choose multiple actions with respective probabilities.

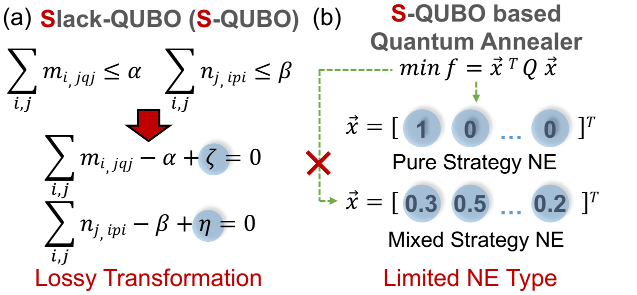
(a) Slack-QUBO (S-QUBO)
$\sum_{i,j} m_{i,jqj} \leq \alpha$
$\sum_{i,j} n_{j,ipi} \leq \beta$
$\sum_{i,j} m_{i,jqj} - \alpha + \zeta = 0$
$\sum_{i,j} n_{j,ipi} - \beta + \eta = 0$
Lossy Transformation
(b)
S-QUBO based Quantum Annealer
$min\, f = \vec{x}^{\,T} Q\, \vec{x}$
$\vec{x} = [\ 1\ \ 0\ \dots\ 0\ ]^T$
Pure Strategy NE
$\vec{x} = [\ 0.3\ \ 0.5\ \dots\ 0.2\ ]^T$
Mixed Strategy NE
Limited NE Type

Figure 1: Existing approaches for addressing NE. (a) The Slack-QUBO conversion (S-QUBO) performs a lossy transformation to handle inequalities at the cost of extra slack variables; (b) S-QUBO based quantum annealers, such as D-Wave, solve only limited type of NE.

Verifying the existence of NE has been proven to be NP-complete [6], placing it among the most intricate computational challenges within the NP realm. Several NE solvers has been proposed to find NE solutions. [7] has proposed a method to reformulate the task of finding an NE in a game into a quadratic optimization problem. Upon that, two quantum-based NE solvers have been introduced [8] to solve this problem by furtherly transforming it into a quadratic unconstrained binary optimization (QUBO) form [9]. However, this QUBO transformation involves the incorporation of slack terms, which alters the original objective function, leading to potentially wrong NE solutions. Moreover, the quantum solvers in [9] can identify only a subset of pure strategy NE solutions (i.e., binary '0' or '1' values) and do not address mixed strategy NE solutions (i.e., decimal values), thereby remaining a significant portion of NE solutions undiscovered. These challenges inspire us to develop a more versatile and efficient NE solver.

In this paper, we present C-Nash, the first ferroelectric computing-in-memory (CiM) architecture designed to handle both pure and mixed NE strategies with high problem-solving efficiency. The main contributions of this work can be summarized as follows:

- We present a novel transformation method that converts the quadratic optimization problem [7] to a lossless MAX-QUBO form without introducing extra slack variables. This transformation significantly reduces the complexity of the object function (Sec. 3.1).
- Leveraging the non-volatile memory property and three-terminal structure of ferroelectric FET (FeFET), we propose a FeFET-based bi-crossbar structure designed to accelerate vector-matrix-vector (VMV) multiplication, the core operation of QUBO forms (Sec. 3.2).

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06…$15.00
https://doi.org/10.1145/3649329.3655988

- We build a winner-takes-all (WTA) tree for implementing the MAX function within the MAX-QUBO form, along with a two-phase simulated annealing (SA) logic to approach NE solutions (Sec. 3.3 and 3.4).
- The proposed C-Nash achieves significant improvements in both the success rate of finding NE solutions and the number of found solutions, along with a remarkable reduction in time-to-solution compared to other Nash solvers (Sec .4).

Evaluations show that C-Nash not only showcases a remarkable success rate, achieving up to a 68.6% higher rate in discovering NE solutions compared to D-Wave Advantage 4.1 , but also successfully identifies all NE solutions while the others only find some of pure NE solutions. Moreover, C-Nash consumes notably minimal time costs, offering a 105.3-157.9X/18.4-79.0X less time-to-solutions compared to D-Wave 2000 Q6 and D-Wave Advantage 4.1, respectively.

2 BACKGROUND

In this section, we review some basics of NE, NE solvers, CiM and FeFET device.

2.1 Nash Equilibrium Basics

NE is a pivotal solution concept in non-cooperative games. It represents a strategy profile in which no player has an incentive to unilaterally deviate from their respective strategic choices. In a two-player game, an NE solution contains a pair of strategies of player 1 and player 2 (p^*, q^*) such that the expected payoff functions f_1 and f_2 for player 1 and 2, respectively, satisfy:

$$f_1(p^*,q^*) \geq f_1(p,q^*),\ f_2(p^*,q^*) \geq f_2(p^*,q),\ \forall p,\ \forall q \tag{1}$$

The payoff functions are defined as follows:

$$f_1(p,q) = p^T Mq,\ f_2(p,q) = p^T Nq \tag{2}$$

where M and N are $n \times m$ matrices representing the payoffs to player 1 and 2, respectively. p and q are the strategies of player 1 and 2, respectively, i.e., $p = (p_1, p_2, ..., p_n), q = (q_1, q_2, ..., q_m)$, and $\sum_i^n p_i = 1, \sum_j^m q_j = 1$, where p_i and q_j indicate the probability for player 1 and 2 to choose action i and j, respectively. A pure strategy is defined when players make their choices deterministically, i.e., both p^* and q^* have only one nonzero element. For instance, $p_1^* = 1$ implies that player 1 chooses action 1 as his strategy. In contrast, a mixed strategy allows players to choose their actions with probabilities, i.e., p^* and q^* contain multiple decimal elements. For example, the strategy $p_1^* = 0.5, p_2^* = 0.5$ indicates a 50% probability for player 1 to take action 1 and a 50% probability for action 2, respectively.

2.2 Existing NE solvers

A method is proposed in [6] to transform the state of NE (Eq. (1) and 2) into a quadratic optimization problem:

$$\max_{p,q,\alpha,\beta} f = p^T(M+N)q - \alpha - \beta \tag{3}$$

subject to

$$\begin{aligned} Mq - \alpha e &\leq 0 \\ N^T p - \beta l &\leq 0 \\ e^T - 1 &= 0 \\ l^T - 1 &= 0 \\ p, q &\geq 0 \end{aligned} \tag{4}$$

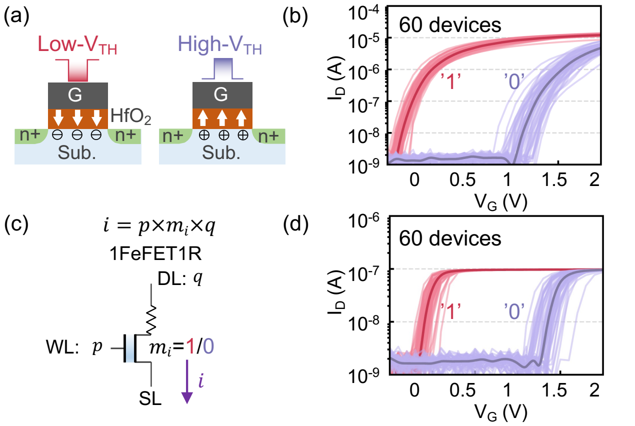

Figure 2: (a) FeFET can be programmed to store low/high V_{TH} by applying different write pulses; (b) I$_D$-V$_G$ characteristics of FeFET storing binary state m_i; (c) The 1FeFET1R structure storing binary m_i naturally performs $i = p \times m_i \times q$ through its drain-source current i by applying two inputs p and q at the gate WL and drain DL, respectively, and leveraging (d) the I$_D$-V$_G$ characteristics with suppressed ON current variability.

where α and β are scalars, e and l are $n \times 1$ and $m \times 1$ vectors of ones, respectively.

To find the NE solutions of Eq. (3) with the constraints in Eq. (4), several quantum-based NE solvers have been proposed [8, 9]. These solvers initially transform the problem into a QUBO formula and subsequently map it onto quantum annealers to search for NE solutions. The general QUBO form [10] is expressed as:

$$\min y = \vec{x}^{\,T} Q \vec{x} \tag{5}$$

where $\vec{x} \in \{0,1\}^n$, and Q is an $n \times n$ matrix. By introducing slack variables, the quadratic optimization problem above can be transformed into QUBO formula:

$$\begin{aligned} \min f = &-p^T(M+N)q + \alpha + \beta \\ &+ A(\sum_i p_i - 1)^2 + B(\sum_j q_j - 1)^2 \\ &+ C(\sum_{i,j} m_{i,j}q_j - \alpha + \zeta)^2 + D(\sum_{j,i} n_{i,j}p_i - \beta + \eta)^2 \end{aligned} \tag{6}$$

where A, B, C, D are constants. We refer to this transformation as Slack-QUBO (S-QUBO). As illustrated in Fig. 1(a), S-QUBO introduces slack variables ζ and η to equate inequalities. This lossy transformation changes the original constraints and results in potential deviations in optimal solutions. To address this problem, we propose a novel transformation method to losslessly convert the original constraints into an equality.

The D-Wave quantum annealers are proposed for implementing S-QUBO [8]. However, as shown in Fig. 1(b), the annealer is limited to address pure strategy NE, i.e., only two states for each variable can be represented, i.e., $x_i = 1$ or 0, but unable to accommodate mixed strategy NE, which assigns decimal probability for the variables, i.e., $x_i = 0.3$. Meanwhile, practical D-Wave annealers require expensive cryogenic cooling and exhibit limited connectivity between variables, affecting capability in addressing complex and large-scale S-QUBO [11–13]. To tackle these challenges, we propose to leverage FeFET based CiM architecture to address both pure and mixed strategy NE with improved solving efficiency.

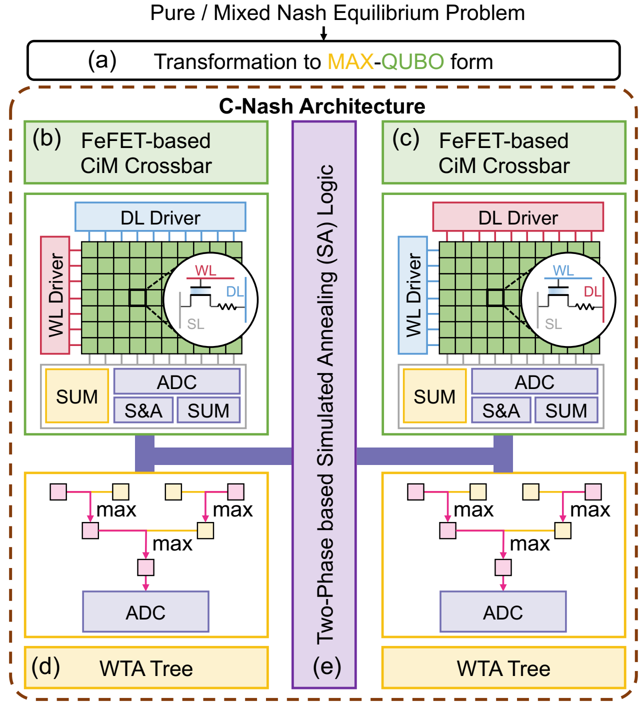
Pure / Mixed Nash Equilibrium Problem
(a) Transformation to MAX-QUBO form
C-Nash Architecture
(b) FeFET-based CiM Crossbar
(c) FeFET-based CiM Crossbar
DL Driver
WL Driver
WL
DL
SL
SUM
ADC
S&A
SUM
max
max
max
ADC
Two-Phase based Simulated Annealing (SA) Logic
(d) WTA Tree
(e)
WTA Tree

Figure 3: Overview of C-Nash. (a) MAX-QUBO transformation; (b), (c) FeFET-based bi-crossbar structure; (d) Winner-takes-all (WTA) tree; (e) Two-phase based simulated annealing (SA) logic.

2.3 CiM Preliminaries and FeFET Basics

Crossbar structure is an essential implementation approach of CiM, which employs non-volatile memory (NVM), e.g., resistive RAM (ReRAM) [14] and magnetic tunneling junction (MTJ) [15], to store matrix weight elements and perform parallel vector-matrix multiplication operations for neural network accelerations. However, these NVMs still possess untapped potential for enhancing energy and area efficiency. This has prompted us to explore the use of another NVM technology , namely FeFET, for designing the crossbar. Compared with other NVMs, e.g., ReRAM [16] and MTJ [17], FeFET, as shown in Fig. 2(a), offers several advantages, including CMOS-compatibility, energy efficient read and write due to voltage-driven mechanism, and compact three-terminal structure [18–24].

Fig. 2(a) illustrates that FeFET can store low and high V_{TH} states by applying negative and positive write pulses to its gate. Fig. 2(b) presents the experimentally measured I_D-V_G characteristics of FeFET. Capitalizing on the characteristics, [25] introduced a 1FeFET1R structure shown in Fig. 2(c) to suppress the ON current variability. The suppressed ON current shown in Fig. 2(d) enable the structure storing m_i to implement consecutive multiplication, i.e., $i = p \times m_i \times q$ by applying input p / q at the gate WL / drain DL. This operation can be extended to support VMV multiplications when 1FeFET1R cells are organized in an array, which is exceptionally well-suited for the QUBO computations [26].

3 C-NASH ARCHITECTURE

In this section, we introduce C-Nash, the first ferroelectric CiM architecture to efficiently solve mixed strategy NE. Fig. 3 depicts an overview of the proposed C-Nash architecture. To address mixed strategy NE problems, instead of employing a lossy conversion like S-QUBO (see Sec. 2.2), C-Nash performs a lossless transformation of the quadratic optimization problem into a MAX-QUBO formulation (see Sec. 3.1), which is then mapped onto a FeFET-based CiM bi-crossbar structure (see Sec. 3.2). A WTA tree (see Sec. 3.3) is built to facilitate the iterative computations of the MAX-QUBO forms, following a two-phase based SA process (see Sec. 3.4).

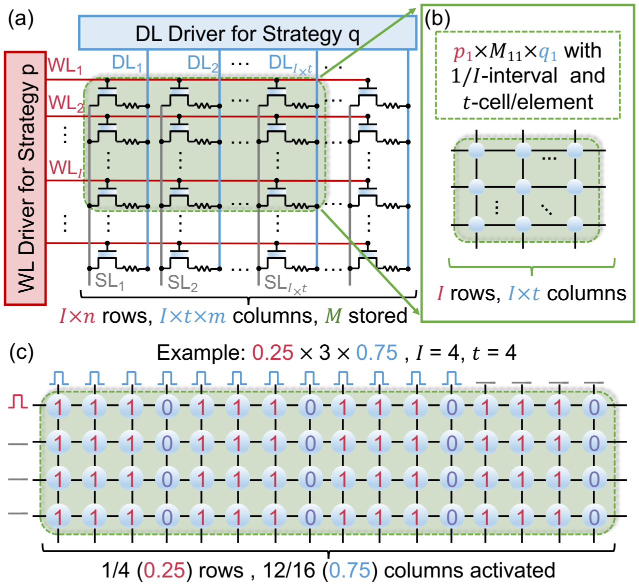

Figure 4: Mapping strategy of bi-crossbar. (a) A FeFET based crossbar implementing $p \times M \times q$ with strategy inputs quantified to I intervals and each element of M represented by t cells; (b) Subarray implementing scalar multiplications $p_1 \times M_{11} \times q_1$; (c) A mapping example of $0.25 \times 3 \times 0.75$ with $I = 4$ and $t = 4$.

3.1 Transformation to MAX-QUBO

Unlike S-QUBO approach, our proposed transformation converts the two inequalities in Eq. (4) into two equations:

$$\alpha = \max(Mq) \tag{7}$$

$$\beta = \max(N^T p) \tag{8}$$

where Mq and $N^T p$ are product vectors with n and m elements, respectively. We then obtain the MAX-QUBO form as follows:

$$\min_{p,q} f = \max(Mq) + \max(N^T p) - p^T (M + N) q \tag{9}$$

where $\sum_i p_i = 1$ and $\sum_i q_i = 1$ are satisfied by circuits.

Therefore, given a decision-making problem (payoff matrices M and N) and its associated strategies from two players (p and q), the objective function can be divided into three components: $p^T (M + N)q$, $\max(Mq)$, and $\max(N^T p)$. Note that all three components involve matrix-vector (MV) or vector-matrix-vector (VMV) multiplications, which can be addressed perfectly by our proposed FeFET-based CiM design.

3.2 FeFET-based Bi-Crossbar

We employ the 1FeFET1R structure shown in Fig. 2 to build the CiM crossbar, as shown in Fig. 3(a)/(b). Both strategies p and q are applied to the inputs of crossbars, which store the payoff matrices.

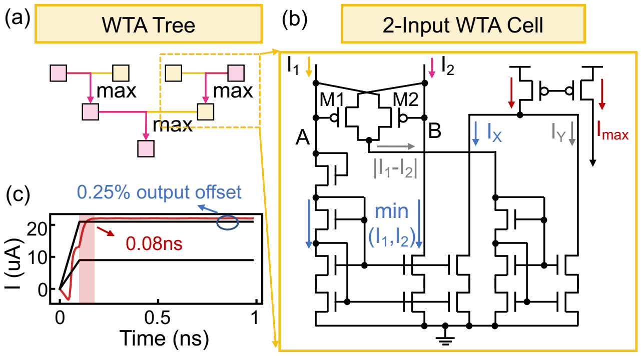

Figure 5: (a) Diagram of a WTA tree; (b) Schematic of a 2-input WTA cell; (c) Transient waveforms of a WTA cell.

Fig. 4(a) depicts a comprehensive mapping of $p^T Mq$ to the crossbar. The payoff matrix M is stored within 1FeFET1R cells, and the input strategies p and q are encoded as driving voltages applied at the WLs and DLs, respectively. The summed output currents on the SLs effectively represent the product of $p^T Mq$. The mapping strategy assumes that (i) each 1FeFET1R cell stores 1 bit, t cells are used to represent a payoff matrix element, and t is determined by the max value of matrix element; (ii) The probability of taking an action within the mixed strategies p and q is quantified into I intervals, indicating that I rows/ $I \times t$ columns are activated to represent a decimal value in the range $\{0, 1/I, 2/I, ..., 1\}$. Therefore, the size of a crossbar implementing $p \times M \times q$ in (a) and $p_1 \times M_{11} \times q_1$ in (b) are $(I \times n) \times (I \times t \times m)$ and $I \times (I \times t)$, respectively.

Fig. 4(c) shows an example of (b), assuming $p_1 = 0.25$, $M_{11} = 3$, $q_1 = 0.75$, i.e., a multiplication $0.25 \times 3 \times 0.75$ with $I = 4$ and $t = 4$. The required crossbar size is 4×16. Horizontally, four adjacent cells represent the element 3. 1 out of 4 total rows (i.e., 0.25) and 8 out of 12 total columns (i.e., 0.75) are activated, respectively. The accumulated current of the crossbar representing the product is then measured and quantified as the MAX-QUBO form result.

3.3 Winner-Takes-All Tree

In C-Nash, we build a WTA tree to implement MAX function, i.e., $\max(Mq)$, and $\max(N^T p)$, within the MAX-QUBO form as in Sec. 3.1 for the computation of its objective function. Fig. 5(a) shows a WTA tree consisting of three 2-input WTA cells to obtain the maximum among four inputs. Generally, for D inputs, i.e., $D = n$ for $\max(Mq)$, and $D = m$ for $\max(N^T p)$, the number of WTA cells needed N is $N = 2^{K-1}+2^{K-2}+...+2^0 = 2^K - 1$ where $K = \lceil \log_2 D \rceil$.

Fig. 5(b) illustrates the circuit schematic of a 2-input WTA cell, which utilizes a high-swing self-biased cascode current mirror to generate equal currents flowing through nodes A and B with equal voltages. When $I_1 > I_2$, the gate voltage of M1 within the cross-coupled PMOS pair becomes higher than that of M2, turning on M2 to conduct "extra" current $I_1 - I_2$. If $I_2 > I_1$, the gate voltage of M2 increases, turning on M1 to conduct "extra" current $I_2 - I_1$. The smaller input current and the "extra", i.e., $\min(I_1, I_2)$ and $|I_1 - I_2|$, are then copied to I_X and I_Y through the cascode current mirror. The output current, $I_{\max}$, is therefore given by:

$$I_{\max} = I_X + I_Y = \min(I_1, I_2) + |I_1 - I_2| = \max(I_1, I_2) \qquad (10)$$

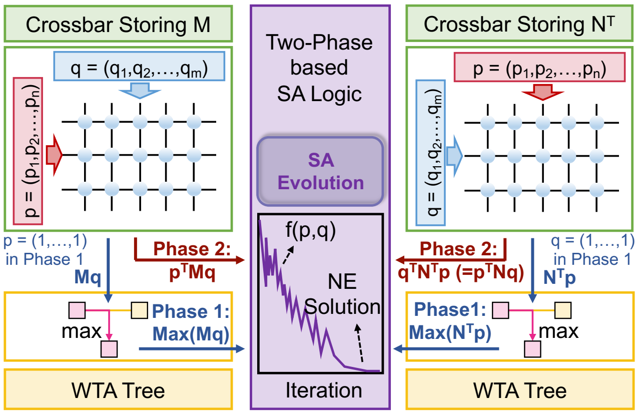

Figure 6: Operation flow of C-Nash within an SA iteration. Phase 1: Computations of $\max(Mq)$ and $\max(N^T p)$; Phase 2: Computation of $p^T(M + N)q$.

The transient waveforms in Fig. 5(c) validate the function of 2-input WTA cell, with 0.08ns latency and 0.25% output offset.

3.4 Operation Flow

The bi-crossbar and the WTA tree described above are used to implement one iteration of MAX-QUBO formula within a two-phase SA process. Fig. 6 depicts the operational flow.

In Phase 1, both the input vector p in the crossbar storing M, and q in the crossbar storing N^T, are configured as unit vectors, with all elements set to 1. Consequently, the outputs of the two crossbars are the matrix-vector multiplication vectors, i.e., Mq and $N^T p$, respectively. These two vectors are then fed into the current domain WTA trees to identify their maximum elements $\max(Mq)$ and $\max(N^T p)$, respectively. These two maximum elements, serving as two terms in the objective function Eq. (9), are recorded in the two-phase based SA logic for further computation.

In Phase 2, the bi-crossbar perform VMV multiplications, i.e., $p^T Mq$ and $p^T Nq$, respectively, while the WTA trees are deactivated. The VMV multiplication outputs are directly transmitted to the two-phase based SA logic. Through addition and subtraction operations, the value of the object function Eq. (9) given a pair of strategy (p, q) is computed as an iteration result of annealing process.

Alg. 1 illustrates the SA process in C-Nash. During each SA iteration, the strategy pair in the last iteration randomly increment or decrement the action probabilities by the value of interval to generate new strategy pair (p_n, q_n) for MAX-QUBO computation. The output value of the objective function given the strategy pair, denoted as f_n, is computed. The SA logic then compares f_n with the objective function recorded f_c and updates the recorded strategy pair and corresponding objective function f_c per the comparison and probability related to the annealing temperature.

4 EVALUATION

In this section, we validate the robustness of the crossbar and WTA component of C-Nash, and evaluate the problem-solving efficiency of C-Nash addressing three games in [8]. All simulations were performed using Cadence SPECTRE, and the Preisach FeFET model [27] was adopted. For MOSFETs, the TSMC 28nm model with TT

Algorithm 1 Simulated Annealing of C-Nash for Mixed Strategy Nash Equilibrium

```
Input:
  $p = p_0,\ q = q_0$                         ▷ Generation of initial strategy pair
  $T = T_{max}$                               ▷ Starting temperature
Output: Mixed strategy Nash equilibrium solution.
  $p_c = p_0,\ q_c = q_0$
  $f_c = f(p_c, q_c)$
  while $T \geq T_{min}$ do
  Generate a new strategy pair $(p_n, q_n)$
  $f_n = f(p_n, q_n)$
  $\Delta E = f_n - f_c$
      if $\Delta E \leq 0$ then
          $p_c = p_n,\ q_c = q_n$             ▷ Accept the new strategy pair
      else
          Accept $(p_n, q_n)$ with a probability $e^{\frac{-\Delta E}{T}}$
      end if
      $T = D(T)$                              ▷ Temperature decay
  end while
```

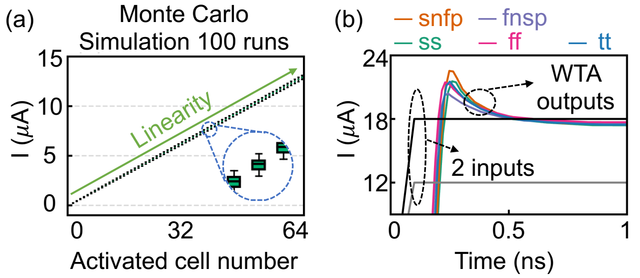

Figure 7: Robustness validation on C-Nash. (a) Monte Carlo simulation exhibits a good linearity of a 64×64 crossbar w.r.t the number of activated cells within a column; (b) Waveforms of WTA with different process corners.

process corner was employed at 27°C. The wiring parasitics for the 28nm technology node were extracted from DESTINY [28].

4.1 Robustness of C-Nash

We evaluate the robustness of C-Nash considering the device variability, where each 1FeFET1R cell of a 64× 64 crossbar assumes a device-to-device variability, i.e., $\sigma = 40mV$ for FeFET V_{TH} from [29], and an 8% resistor variability derived from [30]. Fig. 7(a) illustrates the output currents of the crossbar across 100 Monte Carlo simulations, validating a robust linearity with respect to the number of activated cells. The output waveforms of WTA component across various process corners, including ss, snfp, fnsp, ff, and tt, shown in Fig. 7(b), confirm the robustness of WTA tree in C-Nash.

4.2 NE Solving Efficiency

Table 1: Success Rates of Finding an NE Solution

Nash Solver	Battle of the Sexes (2 actions)	Bird Game (3 actions)	Modified Prisoner's Dilemma (8 actions)
D-Wave 2000 Q6†	99.62	88.16	-
D-Wave Advantage 4.1†	98.04	72.36	13.30
C-Nash (this work)	100	88.94	81.90

†: Extracted from literature.

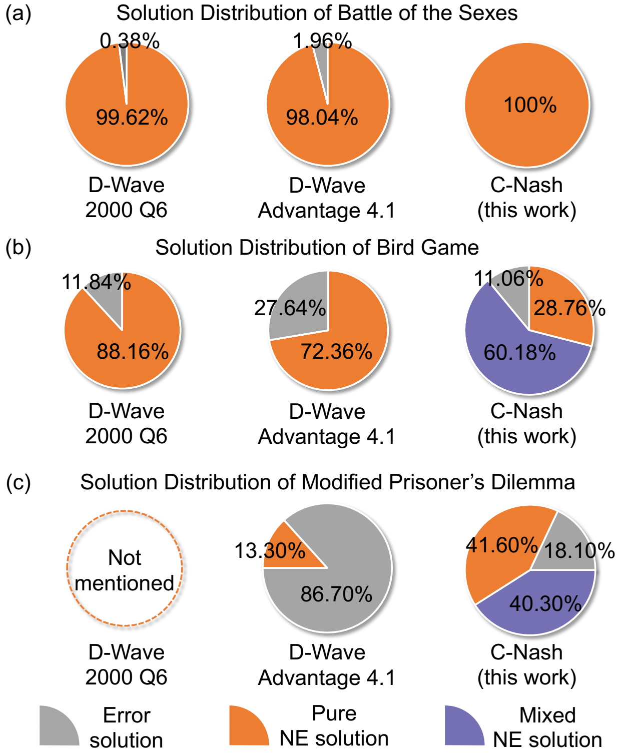

Figure 8: Solution distributions of three Nash solvers in solving three games: (a) Battle of the Sexes; (b) Bird Game; (c) Modified Prisoner's Dilemma.

The problem-solving efficiencies between C-Nash and two D-Wave quantum NE solvers [8] are evaluated, in terms of success rate, number of found solutions (pure or mixed) and average time to find solutions. The benchmark dataset comprises three instances [8], i.e., "Battle of the Sexes", "Bird Game", and "Modified Prisoner's Dilemma", each involving two, three, and eight actions, respectively. The target NE solutions of these games are obtained using Nashpy [31] as the ground truth. The instances are executed 5000 SA runs using C-Nash, with each run comprising 10000, 15000 and 50000 iterations, respectively. Time required for finding solutions is derived based on the operational frequency of FeFET crossbar arrays detailed in [29], scaling to a precision of 1-bit/1-bit.

Table 1 summarizes the success rate of finding an NE solution using three Nash solvers. The two D-Wave Nash solvers exhibit success rates exceeding 90% in addressing the relatively simple "Battle of the Sexes" game with two actions. Nevertheless, as the problem complexity increases, such as in the "Bird Game" with three actions, their success rates drop to 88.16% and 72.36%, respectively. The D-Wave Advantage 4.1 achieves a low success rate of 13.3% in more complex "Modified Prisoner's Dilemma", featuring eight actions. These results show that with increased problem complexity, conventional transformation method S-QUBO introduces more slack variables, thus leading to larger deviations in the object function,

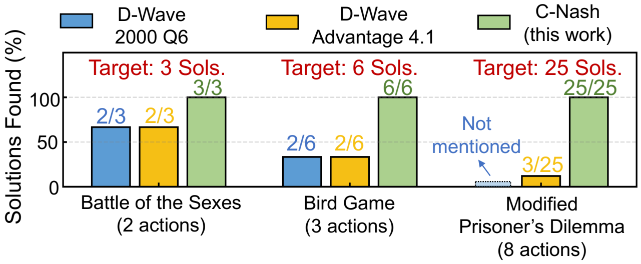

Figure 9: Proportions of distinct NE solutions found by different NE solvers.

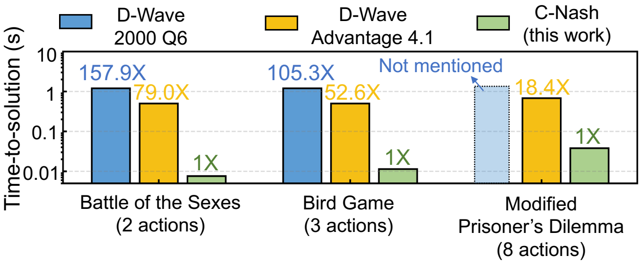

Figure 10: Time to solutions of three Nash solvers.

and degraded performance of Nash solvers. On the contrary, C-Nash achieves a remarkable 100% success rate in addressing the "Battle of the Sexes", and maintains higher success rates, i.e., 88.94% and 81.90% in the other two games. This is because that the proposed lossless MAX-QUBO conversion of C-Nash preserves the integrity of object function, eliminating the potential solution deviations.

We examine the quality of found solutions, specifically (i) the distributions of NE solutions identified by each NE solver across all SA runs, and (ii) the number of distinct solutions found by solvers. Fig. 8(a)-(c) compare the distribution of solutions found by the NE solvers in "Battle of the Sexes", "Bird Game" and "Modified Prisoner's Dilemma", respectively. It can be seen that C-Nash not only identifies pure NE solutions in all games, but also finds mixed NE solutions the other two solvers cannot find. Note that solvers may find the same solutions in different SA runs. Fig. 9 reveals that as the problem complexity increases, the proportions of distinct solutions found by the two D-Wave solvers over target (ground truth) solutions gradually diminish, while C-Nash consistently discovers all possible solutions. This is because that C-Nash leverages lossless MAX-QUBO transformation to maintain the objective function's integrity, and can accommodate MAX-QUBO form with decimal values by exploiting the FeFET to support VMV multiplications.

Fig. 10 illustrates the average time for finding the solutions of the three Nash solvers. Compared to the other two D-Wave-based solvers, C-Nash consumes much less time for finding the solutions. This is due to two main factors: (i) The S-QUBO transformation of the two D-Wave based solvers deviates the objective function, resulting in potential "fake" optimal solution that minimizes the deviated objective function but is not the NE solution for original objective function. The integrity of the objective function in C-Nash prevents the "fake" optimal solution, thus requiring fewer SA runs to find all NE solutions. (ii) The core operations within an SA iteration, i.e., MV multiplication, are accelerated by the FeFET-based CiM bi-crossbar structure, resulting in significant time savings. Therefore, C-Nash achieves a better time-to-solution.

5 CONCLUSION

In this paper, we present C-Nash, the first ferroelectric CiM architecture designed for solving mixed strategy NE solutions. Our work proposes a novel transformation method that converts quadratic optimization problem into a MAX-QUBO form without changes in original objective function. Leveraging the unique characteristics of FeFET, we introduce a FeFET-based bi-crossbar that accelerates the VMV multiplication for MAX-QUBO form, and a WTA tree that implements the MAX function. The iterative MAX-QUBO form computations are controlled by a SA logic. Evaluations show significant improvements of C-Nash than other NE solvers, including higher success rates in finding NE solutions, increased number of solutions that can be found, and less time costs for finding solutions.

ACKNOWLEDGMENT

This work was partially supported by National Key R&D Program of China (2022YFB4400300), NSFC (62104213, 92164203) and SGC Cooperation Project (Grant No. M-0612).

REFERENCES

[1] D. Carfì *et al.*, "Fair redistribution in financial markets: a game theory complete analysis," *Journal of Advanced Studies in Finance*, vol. 2, no. 2, p. 4, 2011.
[2] S. Roy *et al.*, "A survey of game theory as applied to network security," in *2010 43rd Hawaii International Conference on System Sciences.* IEEE, 2010, pp. 1–10.
[3] A. Vetta, "Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions," in *The 43rd Annual FOCS.* IEEE, 2002, pp. 416–425.
[4] C. Zhao *et al.*, "Particle swarm optimization algorithm with self-organizing mapping for nash equilibrium strategy in application of multiobjective optimization," *IEEE TNNLS*, vol. 32, pp. 5179–5193, 2020.
[5] R. Axelrod, "Effective choice in the prisoner's dilemma," *Journal of conflict resolution*, vol. 24, pp. 3–25, 1980.
[6] G. Gottlob *et al.*, "Pure nash equilibria: Hard and easy games," in *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, 2003, pp. 215–230.
[7] O. L. Mangasarian *et al.*, "Two-person nonzero-sum games and quadratic programming," *Journal of Mathematical Analysis and applications*, vol. 9, pp. 348–355, 1964.
[8] F. S. Khan *et al.*, "Calculating nash equilibrium on quantum annealers," 2023.
[9] O. Okrut *et al.*, "Calculating nash equilibrium and nash bargaining solution on quantum annealers," *arXiv e-prints*, pp. arXiv–2112, 2021.
[10] P. Date *et al.*, "Qubo formulations for training machine learning models," *Scientific reports*, vol. 11, p. 10029, 2021.
[11] T. Albash *et al.*, "Demonstration of a scaling advantage for a quantum annealer over simulated annealing," *Physical Review X*, vol. 8, p. 031016, 2018.
[12] V. S. Denchev *et al.*, "What is the computational value of finite-range tunneling?" *Physical Review X*, vol. 6, p. 031015, 2016.
[13] S. Boixo *et al.*, "Computational multiqubit tunnelling in programmable quantum annealers," *Nature communications*, vol. 7, p. 10327, 2016.
[14] L. Zhao *et al.*, "Convfifo: A crossbar memory pim architecture for convnets featuring first-in-first-out dataflow," in *29th ASP-DAC.* IEEE, 2024, pp. 824–829.
[15] A. Mondal *et al.*, "In-situ stochastic training of mtj crossbar based neural networks," in *Proceedings of the International Symposium on Low Power Electronics and Design*, 2018, pp. 1–6.
[16] S. Salahuddin *et al.*, "The era of hyper-scaling in electronics," *Nat. Electron.*, vol. 1, pp. 442–450, 2018.
[17] C. Zhuo *et al.*, "Design of ultra-compact content addressable memory exploiting 1t-1mtj cell," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.
[18] X. S. Hu *et al.*, "In-memory computing with associative memories: a cross-layer perspective," in *2021 IEEE IEDM.* IEEE, 2021, pp. 25–2.
[19] Q. Huang *et al.*, "Computing-in-memory using ferroelectrics: From single-to multi-input logic," *IEEE D&T*, vol. 39, no. 2, pp. 56–64, 2021.
[20] J. Cai *et al.*, "Energy efficient data search design and optimization based on a compact ferroelectric fet content addressable memory," in *ACM/IEEE DAC*, 2022, pp. 751–756.
[21] X. Yin *et al.*, "Ferroelectric ternary content addressable memories for energy-efficient associative search," *IEEE TCAD*, vol. 42, no. 4, pp. 1099–1112, 2022.
[22] Q. Huang *et al.*, "Fefet based in-memory hyperdimensional encoding design," *IEEE TCAD*, 2023.
[23] C.-K. Liu *et al.*, "Cosime: Fefet based associative memory for in-memory cosine similarity search," in *IEEE/ACM ICCAD*, 2022, pp. 1–9.
[24] H. Xu *et al.*, "On the challenges and design mitigations of single transistor ferroelectric content addressable memory," *IEEE EDL*, 2023.
[25] X. Yin *et al.*, "An ultracompact single-ferroelectric field-effect transistor binary and multibit associative search engine," *Advanced Intelligent Systems*, p. 2200428, 2023.
[26] X. Yin *et al.*, "Ferroelectric compute-in-memory annealer for combinatorial optimization problems," *Nature Communications*, vol. 15, no. 1, p. 2419, 2024.
[27] K. Ni *et al.*, "A circuit compatible accurate compact model for ferroelectric-fets," in *IEEE VLSI*, 2018, pp. 131–132.
[28] M. Poremba *et al.*, "Destiny: A tool for modeling emerging 3d nvm and edram caches," in *DATE.* EDA Consortium, 2015, pp. 1543–1546.
[29] T. Soliman *et al.*, "First demonstration of in-memory computing crossbar using multi-level cell fefet," 2023.
[30] D. Saito *et al.*, "Analog in-memory computing in fefet-based 1t1r array for edge ai applications," in *2021 Symposium on VLSI Technology.* IEEE, 2021, pp. 1–2.
[31] The Nashpy project developers , "Nashpy: <release title>." [Online]. Available: http://dx.doi.org/10.5281/zenodo.<DOINUMBER>