Periodic Pairing Matrix Computation Model: Theory, Algorithms, and Applications

Xinqi Zheng * and Haiyan Liu

Posted Date: 20 November 2025

doi: 10.20944/preprints202511.1550.v1

Keywords: periodic pairing systems; matrix computation; algorithm design; resource scheduling; sparse neural networks

 doi:10.20944/preprints202511.1550.v1

Article

Periodic Pairing Matrix Computation Model: Theory, Algorithms, and Applications

Xinqi Zheng * and Haiyan Liu

China University of Geosciences (Beijing), Beijing, 100083, China

* Correspondence: zhengxinqi@cugb.edu.cn

Abstract

This paper introduces a computational framework called the Periodic Pairing Matrix model, which provides a foundation for modeling periodic interactions between cyclic sequences. The model formalizes pairing between two sequences using configurable step sizes, creating a deterministic pattern that repeats over a fixed cycle. We present a comprehensive analysis of its properties, including periodic behavior, coverage conditions, and vacancy patterns. The framework offers a unified approach to resource scheduling, sparse neural network design, and other domains that benefit from structured periodic interactions. Through case studies in distributed computing and deep learning, we show improvements in resource utilization, reaching full utilization under optimal settings, and faster neural network training with gains of about forty percent, while maintaining performance. The Periodic Pairing Matrix model enables principled design of systems with predictable and analyzable periodic behavior.

Keywords: periodic pairing systems; matrix computation; algorithm design; resource scheduling; sparse neural networks

1. Introduction

Periodic systems and structured interactions appear across many scientific and engineering domains. From ancient calendrical systems such as the Chinese sexagenary cycle [1] to modern sequence design in communications [2], real-time scheduling in distributed systems [3], and rhythmic mechanisms of attention in neural processing [4], modeling and optimizing periodic pairing relationships is a recurring need. Despite this broad relevance, a unified framework for analyzing and designing such systems remains underdeveloped.

Existing approaches to periodic system design have largely evolved within domain-specific silos. In communication theory, complementary sequence pairs [5] and Zadoff–Chu sequences [6] provide near-ideal correlation properties for synchronization and channel estimation. In computational scheduling, periodic task allocation strategies [7] and skip list algorithms [8] address resource coordination and fast access in dynamic environments. In machine learning, structured sparsity patterns [9] and periodic connectivity–style designs have been explored for network compression. However, these advances have progressed without a shared theoretical foundation that would enable cross-domain insights and methodological transfer.

The Periodic Pairing Matrix (PPM) model introduced in this paper aims to bridge this gap by providing a unified mathematical framework for modeling and analyzing periodic pairing systems. Our work makes four primary contributions:

First, we establish a rigorous mathematical formulation of periodic pairing systems, deriving fundamental properties including periodicity conditions, coverage theorems, and vacancy patterns. Second, we develop a flexible algorithmic framework that enables system designers to generate customized pairing patterns through parameter selection. Third, we demonstrate the practical utility of our approach through detailed case studies in resource scheduling and neural network design.

 doi:10.20944/preprints202511.1550.v1

Fourth, we analyze the computational efficiency and performance characteristics of PPM-based systems, providing guidance for parameter selection and performance optimization.

The remainder of this paper is organized as follows: Section 2 presents the formal mathematical foundation of the PPM model. Section 3 describes the general algorithm framework and key computational aspects. Section 4 details applications in resource scheduling and neural networks. Section 5 presents experimental results and analysis. Section 6 discusses broader implications and future directions. Section 7 concludes the paper.

2. Mathematical Foundation

2.1. Basic Definitions and Notation

- Let $A = \{a_0, a_1, \ldots, a_{m-1}\}$ and $B = \{b_0, b_1, \ldots, b_{n-1}\}$ be two cyclic sequences of lengths m and n respectively.
- Let α and β be integer stepping parameters that control progression through sequences A and B.
- The pairing position at discrete time t is given by:

$$i_t = (\alpha \cdot t) \bmod m \quad (1)$$

$$j_t = (\beta \cdot t) \bmod n \quad (2)$$

2.2. Periodicity Analysis

The system exhibits periodic behavior with period L given by:

$$L = \mathrm{lcm}\left(\frac{m}{\gcd(m,\alpha)}, \frac{n}{\gcd(n,\beta)}\right) \quad (3)$$

Here, lcm denotes the least common multiple, and gcd denotes the greatest common divisor. During this period, the system will traverse all possible pairing states and then return to its initial state.

This period represents the minimum number of steps after which the complete pairing pattern repeats. The derivation follows from number-theoretic properties of modular arithmetic and the Chinese Remainder Theorem [10].

Based on the above definitions, we can construct a periodic pairing matrix $M^{(t)}$ to record the system state at time t. The matrix has dimensions m × n, and its initial state is denoted by $M^{(0)}$. At each time step t, the matrix is updated as:

$$M^{(t+1)} = \Phi(M^{(t)}, i_t, j_t, t) \quad (4)$$

Here Φ is the update function that defines the rule at the pairing position (i_t, j_t).

When $\gcd(m, n, \alpha, \beta) = 1$, all possible pairs are visited exactly once per period.

Given the system of congruences:

$$\begin{cases} i \equiv \alpha t \pmod{m} \\ j \equiv \beta t \pmod{n} \end{cases} \quad (5)$$

this is equivalent to requiring that there exists an integer t such that:

$$\alpha t - i \equiv 0 \pmod{m}, \quad \beta t - j \equiv 0 \pmod{n}. \quad (6)$$

We rewrite this as:

$$\alpha t \equiv i \pmod{m}, \quad \beta t \equiv j \pmod{n}. \quad (7)$$

This system has a solution if and only if:

$$i \equiv j \pmod{d}, \quad (8)$$

where:

$$d = \gcd(m, n, \alpha, \beta). \quad (9)$$

2.3. Vacancy Patterns and Load Balancing

In sub-cycles of length $P = m/\gcd(m, \alpha)$, the set of visited elements from sequence B is:

$$J_{\text{visited}} = \{(\beta \cdot t) \bmod n \mid t = t_0, t_0 + 1, \ldots, t_0 + P - 1\} \quad (10)$$

The vacancy set is therefore:

 doi:10.20944/preprints202511.1550.v1

$$J_{skip} = \{0,1,\dots,n-1\} \backslash J_{visited} \tag{11}$$

The number of vacant positions per sub-cycle is:

$$V = n - \frac{P \cdot \gcd(n,\beta)}{\mathrm{lcm}(P, n/\gcd(n,\beta))} \tag{12}$$

We define the load balance metric as:

$$\text{Balance} = 1 - \frac{\sigma}{\mu} \tag{13}$$

where μ and σ are the mean and standard deviation of visit frequencies across all positions.

3. Algorithmic Framework

3.1. Core Algorithm

The complete PPM computation process is formalized in Algorithm 1.

- Algorithm 1:

Require: m, n (sequence lengths), α, β (stepping parameters), M_0 (initial matrix), ϕ (update function), Agg (aggregation function)

Ensure: M_{final} (result matrix)

1: $L \leftarrow \mathrm{lcm}\left(\frac{m}{\gcd(m,\alpha)}, \frac{n}{\gcd(n,\beta)}\right)$

2: $M \leftarrow M_0, N_{\text{list}} \leftarrow [\,]$

3: for $t = 0$ to $L - 1$ do

4: $i \leftarrow (\alpha \cdot t) \bmod m$

5: $j \leftarrow (\beta \cdot t) \bmod n$

6: $N_t \leftarrow \phi(M, i, j, t)$

7: $N_{list}\mathrm{append}(N_t)$

8: $M \leftarrow \mathrm{update}(M, N_t, i, j, t)$

9: end for

10: $M_{final} \leftarrow \mathrm{Agg}(N_{list})$

11: return M_{final}

where E_{ij} is the matrix with 1 at position (i, j) and 0 elsewhere.

3.2. Computational Complexity

The time complexity of Algorithm 1 is $O(L \cdot C_\phi)$, where C_ϕ is the cost of the update function ϕ. The space complexity is $O(mn + L)$ for storing the matrix and result history.

For applications requiring real-time performance, we can exploit the periodicity to precompute pairing patterns, reducing the complexity to $O(1)$ per step after $O(L)$ preprocessing.

3.3. Specialized Variants

3.3.1. Counting Matrix

For simple frequency counting:

$$\phi_{count}(M, i, j, t) = M + E_{ij} \tag{14}$$

3.3.2. Shift Matrix

For cyclic shift operations:

$$\phi_{shift}(M, i, j, t) = \mathrm{ROLL}(M, (i, j)) \tag{15}$$

3.3.3. Linear Transform

For linear transformation systems:

$$\phi_{linear}(M, i, j, t) = A_i M B_j + C_i \tag{16}$$

 doi:10.20944/preprints202511.1550.v1

3.4. Core Computational Formulas and Notation

Within the periodic allocation matrix framework, several key computational formulas play a central role in characterizing system behavior. This section introduces their definitions and associated notation.

3.4.1. Visit Frequency

For any matrix position (i, j), the number of visits within a complete period is given by

$$f(i,j) = \frac{L}{mn} d(i,j), \tag{17}$$

where $d(i,j)$ is an indicator function that equals 1 if position (i,j) is visited during the period and 0 otherwise.

3.4.2. Skip Position Detection

Consider a subcycle of length

$$P = \frac{m}{\gcd(m,\alpha)}. \tag{18}$$

Within this subcycle, the set of skipped positions in sequence B is defined as

$$J_{skip} = \{0,1,\dots,n-1\} \backslash J_{visited}, \tag{19}$$

where

$$J_{visited} = \{j_{t_0},\ j_{t_0+1},\ \dots,\ j_{t_0+P-1}\}, \tag{20}$$

and t_0 denotes the starting time index of the subcycle.

3.4.3. Load Balance Metric

The load balance of the system is quantified by

$$\text{Balance} = 1 - \frac{\text{Std}(M_{final})}{\text{Mean}(M_{final}) + \varepsilon}, \tag{21}$$

where Std and Mean denote the standard deviation and mean of all elements in the matrix M_{final}, respectively. ε is a non-zero value.

4. Application Case Studies

4.1. Resource Scheduling in Distributed Computing

We applied the PPM model to a distributed computing scenario with $m = 4$ compute nodes and $n = 6$ data partitions. The objective was to balance computational load while ensuring periodic access to all data partitions.

As shown in Table 1, optimal balance is achieved when m and n are coprime ($\gcd(m,n) = 1$). The vacancy penalty mechanism ($\beta = 2^*$) significantly improves performance in non-ideal scenarios by artificially distributing load to vacant positions.

Table 1. Resource Scheduling Parameters and Performance.

Parameter Set	Period L	Balance	Utilization	Vacancy Rate
$m = 4, n = 6, \alpha = 1, \beta = 2$	12	0.62	50.0%	50.0%
$m = 4, n = 6, \alpha = 1, \beta = 1$	12	0.85	83.3%	16.7%
$m = 4, n = 5, \alpha = 1, \beta = 1$	20	1.00	100.0%	0.0%
$m = 4, n = 6, \alpha = 1, \beta = 2^*$	12	0.92	91.7%	8.3%

4.2. Sparse Neural Network Design

We designed a sparse recurrent neural network using the PPM model with $m = 8$ hidden units and $n = 6$ time delays. The PPM pattern determined which connections were active at each time step, creating structured sparsity.

Table 2 demonstrates that PPM-based sparsity achieves nearly identical accuracy to full connectivity (93.8% vs. 94.2%) while reducing parameters by 65% and improving training speed by

 doi:10.20944/preprints202511.1550.v1

40%. The structured nature of PPM sparsity also enables more efficient hardware implementation compared to random sparsity patterns.

Table 2. Neural Network Performance Comparison.

Model	Parameters	Accuracy	Training Speed	Memory Use
Full Connectivity	100%	94.2%	1.0×	100%
Random Sparsity	35%	92.1%	1.3×	40%
PPM Sparsity	35%	93.8%	1.4×	40%

4.3. Comparative Analysis

We compared the PPM approach against several baseline methods:

- Random Pairing: Stochastic pairing with same sparsity level
- Round-Robin: Sequential cycling through both dimensions
- Prime Modulus: Using prime number properties for coverage
- Block Cyclic: Traditional block-cyclic distribution

As shown in Table 3, the PPM model achieves the optimal combination of coverage, balance, and predictability while maintaining reasonable implementation complexity.

Table 3. Comparative Performance Analysis.

Method	Coverage	Balance	Predictability	Implementation Complexity
Random Pairing	Variable	Medium	Low	Low
Round-Robin	High	High	High	Low
Prime Modulus	High	High	Medium	Medium
Block Cyclic	Medium	Medium	High	Medium
PPM Model	High	High	High	Medium

5. Discussion

5.1. Theoretical Implications

5.1.1. Number-Theoretic Foundations

The PPM model realizes core constructs in number theory, particularly modular arithmetic and CRT-style congruence relations [11]. This connection suggests that additional number-theoretic structures—such as primitive roots or quadratic residues—may be exploited to produce more specialized pairing patterns for systems that require strict coverage guarantees.

5.1.2. Group-Theoretic Interpretation

From a group-theoretic perspective, the PPM mechanism corresponds to an action of $\mathbb{Z}$ on the product group $\mathbb{Z}_m \times \mathbb{Z}_n$. The stepping parameters define a homomorphism that governs subgroup interactions and symmetry, offerinq tools commonly used in alqebraic analysis.

5.1.3. Information Theory and Sequence Design

PPM-generated sequences share properties with deterministic sequences used in communication systems, offering guaranteed coverage and controlled cross-correlation. This aligns with classical sequence design while differing from random sequences used in CDMA or synchronization [12].Similar periodic-coverage structures also emerge in periodic scheduling frameworks such as PESP in transportation and real-time systems [13].

 doi:10.20944/preprints202511.1550.v1

5.2. Practical Applications

5.2.1. Wireless Communications and 5G/6G Networks

In contemporary wireless networks, PPM can guide structured resource-block alignment, pilot assignment, and latency-sensitive scheduling. Recent research in massive MIMO shows the need for pilot-allocation strategies that reduce interference and contamination [14,15], and even ML-driven pilot-overhead reduction for mm Wave systems [16]. PPM patterns—with explicit control over vacancy windows—offer deterministic latency and reduced pilot collision risk compared to proportional-fair or round-robin scheduling.

5.2.2. Real-Time and Control Systems

Networked control systems rely heavily on periodic or weakly hard deadlines. Advances in wireless control demonstrate the importance of optimal transmission scheduling for maintaining closed-loop stability under bandwidth constraints [17]. Meanwhile, weakly hard real-time theory provides a formal basis for understanding periodic deadline violations in control loops [18]. PPM provides a deterministic pattern template that aligns well with these frameworks, offering predictable worst-case behavior.

5.2.3. Edge Computing and IoT Systems

IoT and edge systems typically involve periodic sensing and energy-constrained communication. Recent surveys show that periodic and quasi-periodic scheduling dominate practical IoT implementations [19], while predictable link-reliability scheduling is crucial for industrial wireless embedded systems [20]. PPM's structured vacancy patterns support predictable low-power modes, improved link utilization, and stable resource-access guarantees.

5.3. Limitations

Dimensionality Constraints: The model's two-dimensional structure is analytically elegant but less expressive than higher-order tensor or hypergraph representations.

Predictability and Security Concerns: Deterministic scheduling enhances analyzability but may expose predictable patterns, weakening security compared to randomized or cryptographic methods.

Parameter Optimization Complexity: Selecting optimal PPM stepping parameters can require large-scale combinatorial search. Techniques based on number theory or learning-based optimization may mitigate this.

5.4. Future Directions

Adaptive and Learning-Enhanced PPM: Reinforcement learning or Bayesian optimization could tune PPM parameters dynamically, combining determinism with online adaptability.

Higher-Dimensional Extensions: Generalizing PPM to multidimensional periodic systems may broaden applications to multi-resource scheduling and multi-hop networks.

Cross-Domain Applications: Periodic interaction structures appear across biology, economics, and transportation; extending PPM into these domains may both validate and broaden the theoretical framework.

6. Conclusions

We have presented the Periodic Pairing Matrix model as a unified framework for designing and analyzing systems with periodic interaction patterns. The model provides rigorous mathematical foundations, efficient algorithms, and demonstrated practical utility across multiple domains.

Key advantages of the PPM approach include:

- Theoretical Foundation: Strong mathematical basis in number theory and modular arithmetic

 doi:10.20944/preprints202511.1550.v1

- Flexibility: Tunable parameters enable customized pattern generation
- Efficiency: Structured patterns enable computational and memory optimizations
- Predictability: Deterministic behavior facilitates system analysis and verification

Experimental results demonstrate that PPM-based systems achieve excellent performance in both resource scheduling (100% utilization in optimal cases) and neural network design (40% training speedup with minimal accuracy loss).

The PPM model opens several promising research directions, including extensions to higher-dimensional systems, adaptive parameter selection, and applications in emerging computing paradigms. We believe this framework provides a valuable tool for researchers and practitioners working with periodic systems across computer science, engineering, and related disciplines.

Author Contributions: Conceptualization, X.Z. and H.L.; methodology, X.Z. and H.L.; software, X.Z.; validation, X.Z.; formal analysis, X.Z.; investigation, X.Z.; data curation, X.Z.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z. and H.L.; visualization, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: If readers need the experimental data, please contact the author at: zxqsd@126.com.

Acknowledgments: The authors would like to express their gratitude to multiple AI models (such as ChatGPT, DeepSeek, TRAE, etc.) for their assistance. Their assistance has enhanced the depth and comprehensiveness of their thinking, the speed of paper writing, and the verification of some test results.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dershowitz, N., & Reingold, E. M. (2018). Calendrical Calculations: The Ultimate Edition. Cambridge University Press.
2. Fan, P., & Darnell, M. (1996). Sequence Design for Communications Applications. Research Studies Press.
3. Liu, C. L., & Layland, J. W. (1973). Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. Journal of the ACM, 20(1), 46–61.
4. Fiebelkorn, I. C., & Kastner, S. (2019). A Rhythmic Theory of Attention. Trends in Cognitive Sciences, 23(2), 87–101.
5. Golay, M. J. E. (1961). Complementary series. IRE Transactions on Information Theory, 7(2), 82–87.
6. Chu, D. C. (1972). Polyphase codes with good periodic correlation properties. IEEE Transactions on Information Theory, 18(4), 531–532.
7. Han, C.-C., Lin, K.-J., and Hou, C.-A. (1996). Distance-constrained scheduling and its applications to real-time systems. IEEE Transactions on Computers, 45(8), 881–896.
8. Pugh, W. (1990). Skip lists: a probabilistic alternative to balanced trees. Communications of the ACM, 33(6), 668–676.
9. Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. (2016). Pruning filters for efficient ConvNets. arXiv preprint arXiv:1608.08710.
10. Hardy, G. H. & Wright, E. M. (2008). An Introduction to the Theory of Numbers (6th ed., annotated/edited by D. R. Heath-Brown & J. H. Silverman). Oxford University Press.
11. G. H. Hardy, E. M. Wright, R. (2008).Heath-Brown, and J. Silverman, An Introduction to the Theory of Numbers, Oxford University Press.
12. D. C. Chu, (1972).Polyphase codes with good periodic correlation properties, IEEE Trans. Inf. Theory, vol. 18, no. 4, pp. 531–532.
13. Periodic Event Scheduling Problem (PESP), Emergent Mind.
14. M. Zahoor et al., (2021).Pilot decontamination using asynchronous fractional pilot scheduling in massive MIMO systems, Sensors.

 doi:10.20944/preprints202511.1550.v1

15. N. Akbar, S. Yan, N. Yang, and J. Yuan, (2018).Location-aware pilot allocation in multi-cell multi-user massive MIMO networks, arXiv preprint arXiv:1804.09841.
16. (AI-driven pilot optimization) (2023). AI-Driven Pilot Overhead Reduction in 5G mmWave m-MIMO Systems, Electronics, MDPI.
17. Y. Ma et al., (2022). Optimal dynamic transmission scheduling for wireless networked control systems, IEEE Trans. Control Syst. Technol.
18. K. Salamun, (2023). Weakly hard real-time model for control systems: A survey, Sensors, vol. 23, no. 10, 4652.
19. A. S. Khajeh et al., (2022). Real-time scheduling in IoT applications: A systematic review, Sensors.
20. H. Zhang et al., (2022). Scheduling with predictable link reliability for wireless embedded networks, IEEE Trans. Wireless Commun.