FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 38, Nº 2, June 2025, pp. 209 - 220
https://doi.org/10.2298/FUEE2502209M

Original scientific paper

IOT BASED MEMORY FAULT DIAGNOSIS AND REPAIRING USING PARTICLE SWARM OPTIMIZATION (PSO)

Vinita Mathur[1], Sanjay Kumar Singh[1], Aditya Kumar Singh Pundir[2]

[1]Amity University Rajasthan, Jaipur, India
[2]Arya College of Engineering & IT, Jaipur, India

ORCID iDs:	Vinita Mathur	https://orcid.org/0000-0002-4132-9609
	Sanjay Kumar Singh	https://orcid.org/0000-0002-4426-3895
	Aditya Kumar Singh Pundir	https://orcid.org/0000-0002-6762-9263

Abstract. *With the advent of Internet of Things based devices, there has been an increased focus on memory, which is an integral part of IoT. As the requirement of memory increased, the probability of fault occurrence will also increase. Research problem addresses the requirement of efficient and autonomous system for detecting and repairing faults in IoT devices using BPSO. As IoT devices are widespread in many applications, ensuring the reliability becomes of foremost relevance. If any of these faults is undetected, then it leads to significant system failures. Traditional methods may not provide the necessary scalability, adaptability and efficiency for the IoT devices. This research aims to enhance the reliability of IoT devices by optimizing fault processes. The adoption of BPSO offers a novel approach to addressing the limitations of traditional fault detection methods, with the potential to significantly improve the performance and reliability of IoT networks. For Optimization, Binary particle Swarm Optimization algorithm is considered using certain parameters like storage temperature, Reference clock frequency, Power consumption, voltage level, ground loop, humidity, output current and EM Interference. On the basis of change in these parameters faults are generated. Once the faults are generated an optimization of fault is done with the help of BPSO in order to create an optimized fault dictionary. This fault dictionary provides the information which faulty memory location is to be repaired first. So that repair solution on the faulty memories can be applied. As in this article we have considered three faults individually stuck at Fault, Transition Fault and Coupling Fault for memory size 16x8x8 and 32x8x8 with spare memory 2*8*8 and 4*8*8. The result shows that the fault rate for optimization using fault detection method is 90.62% where as optimization using BPSO fault rate is 100%. This method provides better fault coverage for wide range of memory faults. However, it might require additional computation time.*

Key words: *Binary particle Swarm Optimization, Internet of Things, Memory testing, Modified memory built in self-repair, RAID 6.*

Received July 31, 2024; revised September 13, 2024 and October 11, 2024; accepted October 17, 2024
Corresponding author: Vinita Mathur
Amity University Jaipur, India
E-mail: Vinita.mathurdec@gmail.com

1. INTRODUCTION

With disruptive growth in IoT (Internet of Thing) in the current technology is gaining demand among the many sectors [1,2]. An IoT describes the system which consists of physical objects which is called "thing" [3]. These are connected to embedded system with sensors and with other technology which are used for connecting and exchanging the information with other systems over the internet. IoT consists of set of protocols (set of rules and regulations), standards, tools which provide designing and deploying the IoT devices and services. IoT devices use the machine learning algorithm to analyses the huge amount of data which is connected to sensor [4,5]. For secure data transmission and storage, memories are required. As the requirement of memories increases, the probability of fault occurrence in memories will also occur. Memory testing and repair is required with faster and optimized way. In order to provide an optimized repair solution a PSO (particle swarm optimization) Algorithm is used [6].

Addressing the gaps—such as real-time performance under network variability, energy efficiency, scalability, and comparative analysis with other algorithms could lead to more robust, efficient, and user-friendly systems. Additionally, exploring long-term reliability, security, and user interface design would further strengthen the practical application of the proposed framework.

The background of research on IoT-based memory fault diagnosis and repair using Particle Swarm Optimization (PSO) [7-9] is rooted in the growing complexity and critical importance of memory systems in modern computing. With the advent of the Internet of Things (IoT), vast networks of connected devices generate and process enormous amounts of data. Memory systems in these devices must be reliable and efficient, as faults can lead to data corruption, system failures, or even security vulnerabilities [10].

The research aims to enhance the reliability of IoT devices by optimizing fault processes. To develop an IoT-based system that leverages PSO for effective and efficient memory fault diagnosis and repair, the system should be capable of autonomously detecting the faults with high accuracy, minimal computational overhead for IoT configurations.

The hybrid GWO-PSO approach is a powerful tool for reactive power planning, offering improved performance over traditional PSO [11] variants. This research contributes to the field by providing an advanced optimization technique [12]. In this paper, for secure data transmission an MQTT protocol and raspberry pi field protocol is used and for frequent data storage (RAID 6 (redundant array of independent disk) is used) which stores the memory array of size 16*8*8 and 32*8*8 and two spare memories 2*8*8 and 4*8*8. For high fault driven and failure tolerance with high data retention time period like archiving memory fault diagnosis model requires [13,14]. In this paper, fault diagnosis is done on the basis of certain parameter like storage temperature (ambient), reference Clock frequency range, Voltage level, Vibration, Ground Loop, Humidity (increase in temp)-temp short circuit, output current and EM Interference (as frequency increase), as we have used Spartan 3E FPGA (Field programmable Gate array) for Static memory model. Once the faults are indentified with the help of fault diagnosis model then optimization is conducted, with the help of BPSO (Binary particle swarm optimization) Algorithm.

This approach leverages the strengths of both rule-based systems and BPSO. The rule-based system ensures that the initial selection is grounded in linguistic or domain knowledge, while BPSO optimizes the selection for the specific task at hand. The result is a robust, low-dimensional semantic representation that can be applied to various NLP tasks with improved efficiency and effectiveness [15].The basic parameters used in BPSO algorithm are dimension of the problem, population which is also called number of particle, acceleration coefficient, number of iteration, velocity of particle, inertia weight, random values of the contribution scale of the social and cognitive components and size of the neighborhood particles, mutation rate, stopping criteria and initial range.

Effective fault diagnosis model is proposed on the bases of parameters, including storage temperature, clock frequency, voltage levels, vibration, ground loop, humidity, output current, and electromagnetic interference. BPSO algorithm improved the accuracy and efficiency of fault identification and correction. The article concludes that the integration of BPSO with advanced fault diagnosis method which leads to improve fault coverage and reliability in memory systems, especially within the context of IoT devices and FPGA-based static memory models.

This paper contributes a novel approach to provide better fault coverage using optimization for wide range of memory faults. This paper is categorized into following sections. Section 2 describes the proposed architecture for fault detection and repairing using PSO Algorithm. Section 3 describes the standard PSO and BPSO for faults optimization. Section 4 presents the effective memory management unit and repair algorithm. Section 5 shows the comparative result of the proposed work and in Section 6 describes the conclusion and future scope of the proposed work.

2. Proposed Work

Fig. 1 shows a proposed architecture of IoT framework [15] for fault detection and repairing using optimizing technique. BPSO (Binary particle swarm optimization) algorithm is used to optimize the fault for repair algorithm. Faults free memory is display on the smart display device with the help of IoT. For Fault detection certain parameters (9 parameters) are taken under consideration. Smart devices access the data stored in RAID 6 array by connecting to it over a network using protocol like SMB (server message block) and NFS (Network file system). SMB is a network protocol used file sharing between the devices in network. It supports different types of authentication system. It provides encryption of data transmission over network for data protection. It allows only the active directory, whereas NFS is a distributed file system protocol that allows the system to access the data from the local storage. It enables the sharing of file and directories across the network. It provides transparency for the remote file and directories appears as if they are part of the local file system. It supports the authentication system to control the access for sharing resources. It is generally integrated with LDAP (Lightweight Directory access protocol). In such aforesaid network, the smart device acts as a client and interacts with RAID 6 array as network storage resources.

RAID 6 is a redundant array of independent disk of configuration that offers high fault tolerance and data protection. It provides high redundancy by utilizing a parity block. This parity information is provides across multiple hard drives with in the array. In this paper, RAID 6 consists of four arrays of SRAM (static memory) which split into two arrays. Total memory and spare memory array, total memory size is 16*8*8 and 32*8*8 and spare memory array size is 2*8*8 and 4*8*8.

Fault diagnosis generally requires a fault diagnosis model [16-17]. For this all faults are assembled into the faulty library, using certain parameters for fault selection. In this paper, we have assumed three faults: SAF (stuck at fault), TF (Transition fault), CF (Coupling fault). SAF in BIST (Build in self test) are commonly encountered faults. SAF occur when a signal is permanently stuck at either a logic high or at logic low. TF arise when there is error in the BIST [16] circuit during signal transistor, like changing from logic high to logic low vice versa. CF is unintended interaction or interface occurs between neighboring circuit elements. We have considered theses three fault to analysis individually using 9 fault characteristic parameters as shown in Table1. These parameter are considered for SPATAN 3E (FPGA).

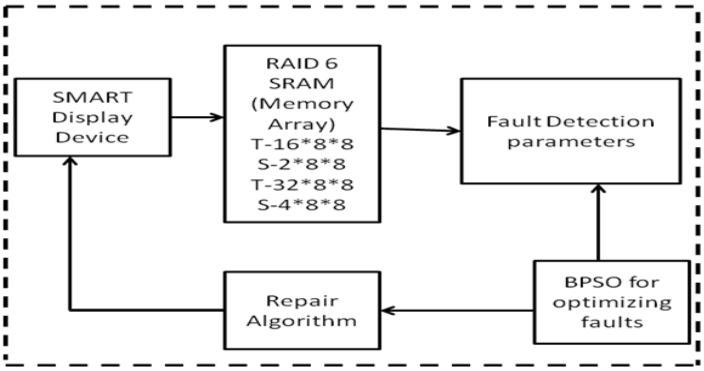

Fig. 1 Proposed architecture

SAF will arise if there is any change in these parameters like storage temperature (ambient temperature), power supply changes, vibration, humidity (due to increase in temperature). These faults occur when memory cell value is stuck at zero or one. TF faults arise when there is change in theses parameters humidity (temporary short circuit), ground looping (misinterpreted logic values). These faults occur when memory cell is not able to make 1 to 0 transitions or 0 to 1 transition. CF will arise if any cell is influenced by other cell due to noise, electrical current, EM interference occurs. These faults arise due to the aggressor cell which forces victim cell in memory to change its value.

There are various numbers of coupled cell combination possible. In this paper faults are generated with the help of fault characteristic parameters as shown in Table 2. Table 2 shows the two memory sizes which we have assumed in this paper - 16*8*8 and 32*8*8 individually for each fault. These memories are split into two parts - memory used and spare memory. Spare memory is for repair process. For repairing the fault an MMBISR algorithm is used.

Table 1 Fault characteristic parameters

S.No.	Fault characteristic parameters	S.No.	Fault characteristic parameters
1	Storage Temperature	6	Ground Loop
2	Reference clock Frequency	7	Humidity
3	Power Consumption	8	Output Current
4	Voltage level	9	EM Interference
5	Vibration		

Table 2 Cases of faults

S.No	Cases for Faults	Memory size	Used Memory	Spare Memory	Total used Memory	Faulty Memory (individual)
1	SAF	16*8*8	14*8*8	2*8*8	1024	43
		32*8*8	28*8*8	4*8*8	2048	101
2	TF	16*8*8	14*8*8	2*8*8	1024	48
		32*8*8	28*8*8	4*8*8	2048	81
3	CF	16*8*8	14*8*8	2*8*8	1024	25
		32*8*8	28*8*8	4*8*8	2048	40

Once the faults are identified the optimization of fault is performed using BPSO algorithm [18]. If fault diagnosis model identifies any faults in the memories, then these faults are categorized in certain cases. As in this paper, we have assumed three types of faults stuck at fault, transition fault and coupling faults for individual memories. Then these faults are categorized into banks and sub banks to create and find the least number of faults in the bank. A faulty dictionary is created which consists of the information that which fault memory location is to be repaired first. So, that repair solution on the faulty memories can be applied.

For repairing the fault in memories a MMBISR (Modified memory built in self-repair) is used which is a hybrid algorithm for SRAM. This algorithm provides least solution for optimized set of row and column combination which gives the suitable repairing operation as shown in Fig. 2.

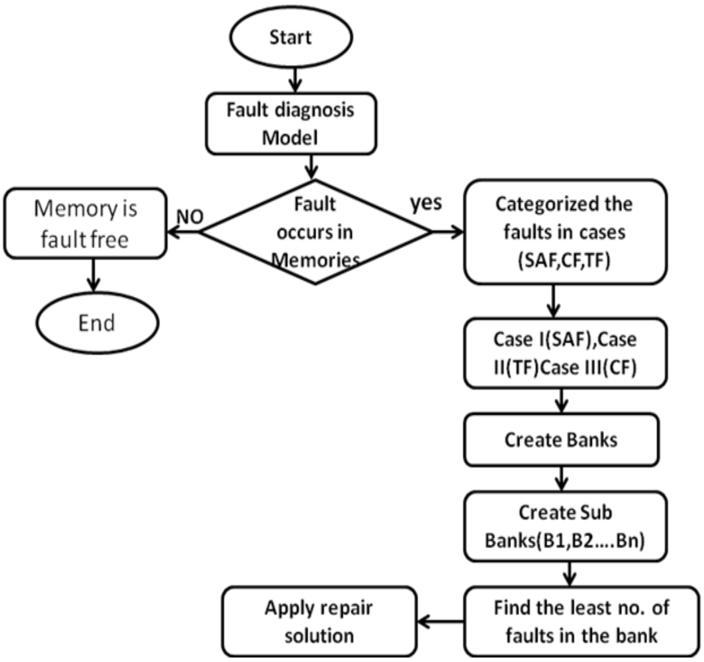

Fig. 2 Flow chart of Repair Algorithm

3. BPSO (BINARY PARTICLE SWARM OPTIMIZATION)

Standard PSO is population based search method in which number of individual called agent (particles) change their state with time and individual changes in velocity. In this each particle flies according to its own experience to find the best solution in the space, on the basis of their experience neighboring particle defines their best solution for itself and its neighbor [19]. Every particle in PSO modifies its position on the basis of its distance between its current position and pbest (personal best), current position, current velocity, and the distance between its current position and gbest (global best) [20]. Fig. 3 shows the basic PSO, here Z is the solution of the space function and Y is the optima that is to be finding theatricality, v_i^{t+1} is the velocity after being affected by the neighbor v_i^t is the current velocity and χ_i^t is the initial particle. The updated position and velocity is calculated as shown in equation (1) and (2).

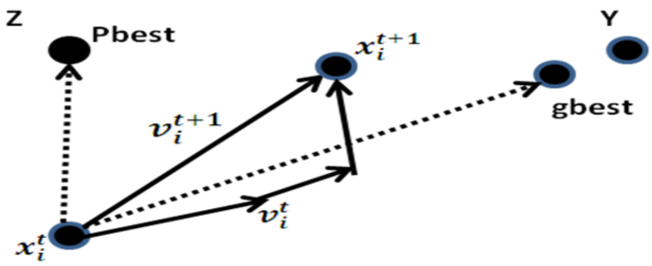

Fig. 3 Basic PSO

$$\nu_i^{t+1} = \nu_i^t + c_1 r_1 (p_{best}^t - \chi_i^t) + c_2 r_2 (p_{best}^t - \chi_i^t) \tag{1}$$

$$\chi_i^{t+1} = \chi_i^t + \nu_i^{t+1} \tag{2}$$

Here, P_{best}^t = Personal best solution

P_{gbest}^t =Global best solution

C_1 and C_2= cognitive learning rate

r_1 and r_2 = Random number range [0,1]

The equation (1) and (2) contain the change in velocity which includes three components momentum, social component and cognitive components which determine how to balance the performance of PSO [21-23].

As BPSO (Binary Particle Swarm Optimization) algorithm contributes to the IoT-based memory fault diagnosis by optimizing the detection processes. It improves fault coverage, reduces computational overhead, and ensures efficient and accurate fault management, all within the constraints of a binary decision-making framework. Flow chart of BPSO Algorithm for fault diagnosis is shown in Fig. 4.

Step 1: Define the BPSO parameters number of particles, number of iterations, number of binary variables (one for each cell), inertia weight=0.5, cognitive coefficients c1 and c2=1.5.

Step 2: Initializes the position of particles in the search space. Function “rand (num Particles, num Variables)” is used to generate a matrix of random value between 0 and 1 where num particle is number of particle swam and num variable is the number of binary variable. To range the random velocity a scale 0.1 is multiple to the function. It is used to update the particles position and to initialize the small random values to avoid large changes in position during the early iterations of the algorithm. Initialize the pBest, it is a matrix that stores the best position found by each particle. Initially, it is set to the same value as particles, meaning each particle's initial position is considered its personal best.

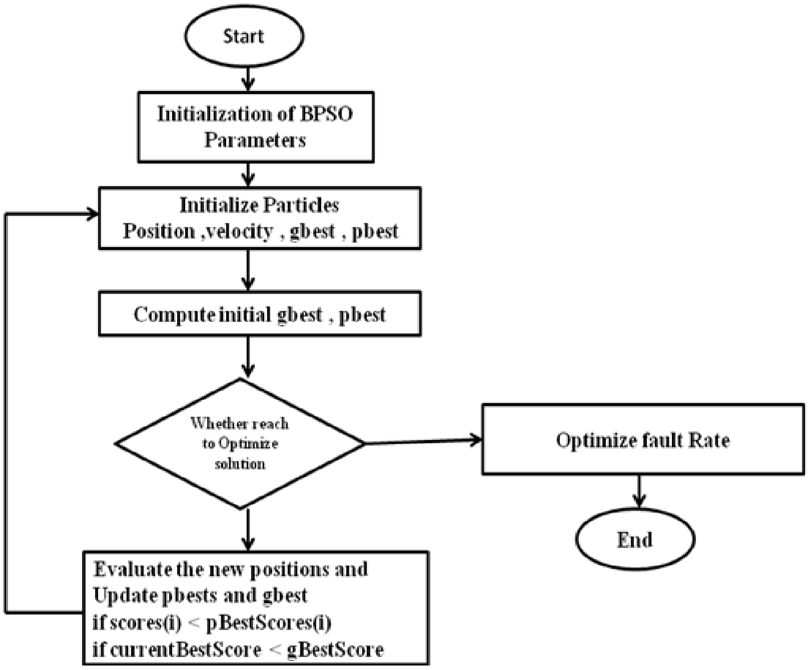

Fig. 4 Flow Chart of BPSO Algorithm

Step 3: Initialize gBest Score with a large value and Compute initial pBest Scores and gBest.

Step 4: Iteration through each generation, updating the particles' velocities and positions based on their personal best positions (pBest) and the global best position (gBest). Iteration process continues until the specified number of iterations is reached, improving the particle swarm performance in finding optimal solutions. The sigmoid function in equation (3) and (4) ensures that the particle positions remain binary by mapping the continuous velocity values to binary positions.

$$Sigmoid(velocities(i,:)) > 0.5 \tag{3}$$

$$Sigmoid(\chi) = 1/(1+\exp(-\chi)) \tag{4}$$

Step 5: Evaluate the new positions and Update personal bests and global best.

4. EFFECTIVE MEMORY MANAGEMENT UNITS AND REPAIR ALGORITHM

Effective Memory Management Unit is shown in Fig. 5 the fault collection process using fault diagnosis model. In this paper, we have considered three cases of faults individually SAF, TF and CF. In case 1:43 and 101 SAF arises for memory size 16*8*8 and 32*8*8. In case 2: 48 and 81 TF arises for memory size 16*8*8 and 32*8*8. In cases 3: 25 and 40 CF arises for memory size 16*8*8 and 32*8*8. Collected faults are divided into sub banks using PSO algorithm. As shown in case1; Bn sub banks are generated on the bases of faults arise for memory array 16*8*8 and 32*8*8.on the bases of number of faults in the banks precedence list is prepare for repairing process. For repairing these fault two spare memory are allocated 2*8*8 for 16*8*8 and 4*8*8 for 32*8*8.

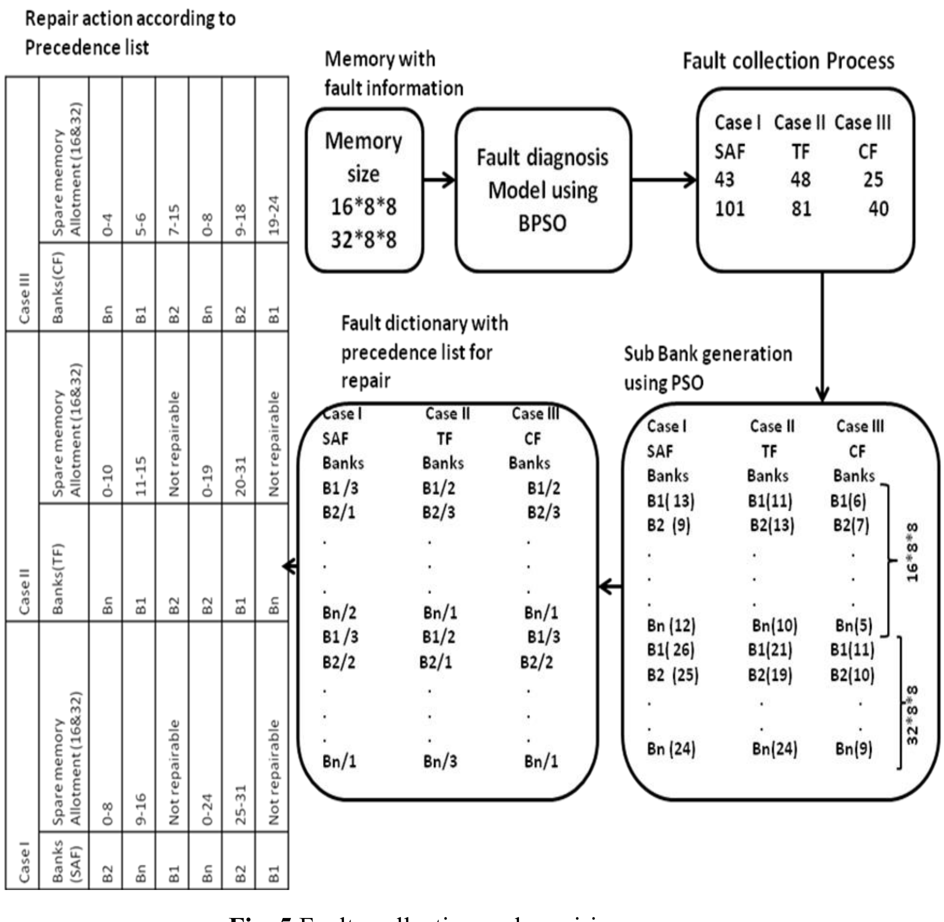

Fig. 5 Faults collection and repairing process

On bases of precedence list memory cell are allocated to faulty memory. First preference for memory repair will be given to the bank which carries less numbers of faults. As from figure in case 1 B2 bank contains 9 SAF which will be repaired by 0-9 spare memory cells then Bn bank contains 12 SAF which will be repaired by 9-16 spare memory cells. Once the spare memory cells are fully occupied then other faulty banks will be non repairable. The same process is used for transition and coupling faults.

For repairing MMBISR algorithm is used. The banks with low precedence will repair first. Precedence list is mapped information of faulty rows/columns with accessible spare rows/columns address. Rows/columns count register contains the information of used spare rows/columns. First preference in the repair process is given to the least number of faulty banks and so on. Once the spare replacement process done, then spare count is decreased by one and attentive row/column address is deleted from the list. If the spare memory is completely occupied then memory bank is declared as faulty and unrepaired. When precedence list is made then only the request of the service is placed in the repair queue as per the precedence list. Service for the repair cell is provided on the basis of FIFO principle. RTL description of proposed work is shown which the total number of spare memory demand goes to the spare allocator and redundancy analyzer module. This module is used for mapping unit. Mapped locations are replaced by the actual row/column address such that if called then they can be used in place of faulty memory.

5. Comparative Results

A comparison between optimized March C- and proposed BPSO algorithm for fault detection with respect time taken is shown in Fig. 6. The lower the execution time, the better the performance. Here, BPSO shows better performance due to lower execution time.

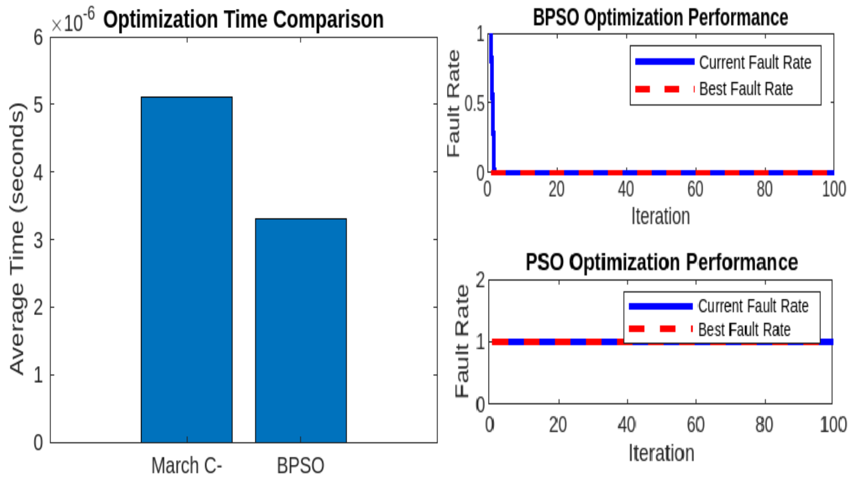

Fig. 6 Optimization time comparisons between March C-and BPSO and comparisons between PSO and BPSO performance

Another comparison between standard PSO and BPSO is also performed on the bases of fault rate with respect to number of iteration. Fault rate is the ratio of number of faulty cells to the total number of cells.

$$\text{Fault Rate} = \frac{No.\ of\ faulty\ cells}{Total\ No.\ of\ cell} \tag{5}$$

As shown in Fig. 6, the fault rate of BPSO is lower which indicates an improved reliability and enhanced performance. Further, another comparison parameters is fault rate convergence and optimal value between PSO and BPSO. As PSO operates on continuous values where as BPSO is designed for discrete/binary values problem as in this paper. Both parameters evaluate the algorithm performance involving to measure how quickly it reaches the optimal solution and how stable the fault detection performance over the time as shown in Fig. 7.

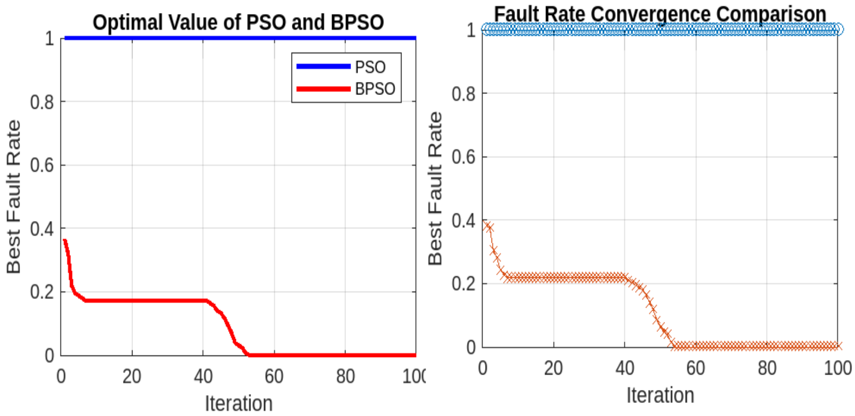

Fig. 7 Optimal and convergence comparisons between PSO and BPSO

ROC (Receiver Operating Characteristic) curve is additional parameter to evaluate the performance of the binary classified model. It provides the tradeoff between the sensitivity (TPR) and the (FPR) specificity across different threshold as shown in equation (5) and (6). Figure 8 show the ROC of the PSO and BPSO. A curve closer to the top-left corner shows that BPSO is superior at detecting faults with a low false positive rate. Table 3 shows the repairing rate and repairing time of proposed work using repairing Algorithm MMBISR which is 94.43 for 16*8*8 and 32*8*8 as repairing time is 2100 ms and 2175ms as shown in Fig. 8.Table 4 shows the metrics of algorithms used.

$$\mathrm{TPR} = \frac{TruePositive}{TruePositive + FalseNegative} \tag{5}$$

$$\mathrm{FPR} = \frac{FalsePositive}{FalsePositive + TrueNegative} \tag{6}$$

Table 3 Repair rate of SRAM

S.No.	SRAM	Parameters	MMBISR Using PSO (SPATAN 6)
1	16*8*8	Repair Rate,	94.43,2100
2	32*8*8	repairing time (ms)	94.43,2175

Table 4 Metrics of PSO and BPSO Value

Metrics	PSO Value	BPSO Value
Fault Detection Rate (FDR)	95%	93%
False Positive Rate (FPR)	5%	6%
Fault Repair Rate (FRR)	90%	88%
Repair Time	3.5sec	3.0 sec

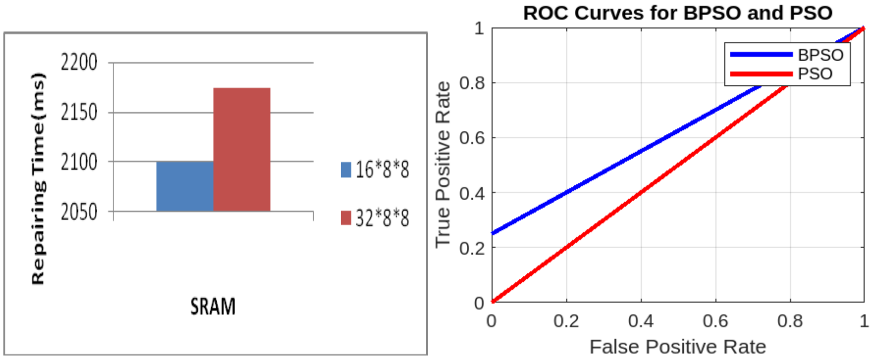

Fig. 8 ROC curve for PSO and BPSO and Repair timing

6. Conclusion

This article proposes the state of art IoT based fault diagnosis model using binary particle swarm optimization. Optimization of fault in memory is performed using certain parameters of PSO algorithm. Result shows that BPSO Algorithm fault detection time is 3.2 sec which is less than particle swarm optimization as it provide optimal solution faster for fault detection and optimization. As in this article we have considered three faults individually stuck at Fault, Transition Fault and Coupling Fault for memory size 16x8x8 and 32x8x8. ROC curve show that BPSO provide better performance for binary classified Model. For repairing purpose we have used MMBISR (Modified Memory Built in Self Repair) algorithm on SPARTAN 3E, which shows better repair rate and 94.43% memory utilization. The integration of BPSO for fault diagnosis and optimization is effective but introduces complexity in terms of implementation. BPSO used in resource-constrained environments like IoT devices, may lead to increased computational overhead. Incorporate machine learning techniques such as clustering, classification or anomaly detection to enhance fault diagnosis capabilities which will improve fault classification accuracy.

References

[1] M. Bagaa, T. Taleb, J. B. Bernabe and A. Skarmeta, "A Machine Learning Security Framework for Iot Systems", *IEEE Access*, vol. 8, pp. 114066-114077, 2020.
[2] N. M. Karie, N. M. Sahri, W. Yang, C. Valli and V. R. Kebande, "A Review of Security Standards and Frameworks for IoT-Based Smart Environments", *IEEE Access*, vol. 9, pp. 121975-121995, 2021.
[3] J. L. Reed and A. Ş. Tosun, "BULWARK: A Framework to Store IoT Data in User Accounts", *IEEE Access*, vol. 10, pp. 15619-15634, 2022.

[4] H. Kim et al., "IoT-TaaS: Towards a Prospective IoT Testing Framework", *IEEE Access*, vol. 6, pp. 15480-15493, 2018.

[5] W. Sun, M. Tang, L. Zhang, Z. Huo and L. Shu, "A Survey of Using Swarm Intelligence Algorithms in IoT", *Sensors*, vol. 20, no. 5, p. 1420, 2020.

[6] R. Singh and B. Bhushan, "Data-Driven Technique-Based Fault-Tolerant Control for Pitch and Yaw Motion in Unmanned Helicopters", *IEEE Trans. Instrum. Meas.*, vol. 70, p. 3502711, 2021.

[7] A. Amouri, V. T. Alaparthy and S. D. Morgera, "A Machine Learning Based Intrusion Detection System for Mobile Internet of Things", *Sensors*, vol. 20, no. 2, p. 461, 2020.

[8] C.-Y. Lee and W.-C. Lin, "Induction Motor Fault Classification Based on ROC Curve and t-SNE", *IEEE Access*, vol. 9, pp. 56330-56343, 2021.

[9] D. Zhang, W. Xiang, Q. Cao, "Application of Incremental Support Vector Regression Based on Optimal Training Subset and Improved Particle Swarm Optimization Algorithm in Real-Time Sensor Fault Diagnosis", *Appl. Intell.*, vol. 51, pp. 3323-3338, 2021.

[10] E. Xu, Y. Li, L. Peng, M. Yang and Y. Liu, "An Unknown Fault Identification Method Based on PSO-SVDD in the IoT Environment", *Alex. Eng. J.*, vol. 60, no. 4, pp. 4047-4056, 2021.

[11] M. Suhail Shaikh, S. Raj, R. Babu, S. Kumar and K. Sagrolikar, "A Hybrid Moth–Flame Algorithm with Particle Swarm Optimization with Application in Power Transmission and Distribution", *Decis. Anal. J.*, vol. 6, p. 100182, 2023.

[12] A. Pundir, "Novel Modified Memory Built in Self-Repair (MMBISR) for SRAM Using Hybrid Redundancy-Analysis Technique", *IET Circuits Devices Syst.*, vol. 13, no. 6, pp. 836-842, 2019.

[13] Z. Yu, L. Zhang and J. Kim, "The Performance Analysis of PSO-ResNet for the Fault Diagnosis of Vibration Signals Based on the Pipeline Robot". *Sensors*, vol. 23, no. 9, p. 4289, Apr 2023.

[14] Y. Cheng, Z. Wang, W. Zhang and G. Huang, "Particle Swarm Optimization Algorithm to Solve the Deconvolution Problem for Rolling Element Bearing Fault Diagnosis", *ISA Trans.*, vol. 90, pp. 244-267, July 2019.

[15] V. Mathur, A. K. Pundir, R. K. Gupta and S. K. Singh, "Recrudesce: IoT-Based Embedded Memories Algorithms and Self-Healing Mechanism", In Proceedings of Congress on Control, Robotics, and Mechatronics, Smart Innovation, Systems and Technologies, 2024, vol. 364, pp. 113-120.

[16] V. Mathur, A. K. Pundir, S. Singh and S. K. Singh, "An Insight into Algorithms and Self Repair Mechanism for Embedded Memories Testing", In Proceedings of Flexible Electronics for Electric Vehicles (FLEXEV 2022), 2024, vol 1065, pp 505-517.

[17] W. Li, W. Chen, C. Guo, Y. Jin and X. Gong, "Optimal Placement of Fault Indicators in Distribution System Using PSO Algorithm", In Proceedings of the 43rd Annual Conference of the IEEE Industrial Electronics Society (IECON 2017), Beijing, China, 2017, pp. 375-380.

[18] S. Kumar, T. Kaur, S. Upadhyay, V. Sharma and D. Vatsal, "Optimal Sizing of Stand Alone Hybrid Renewable Energy System with Load Shifting", *Energy Sources A: Recovery, Util. Environ. Eff.*, vol. 47, no. 1, pp. 1490-1509, 2020.

[19] S. Mahapatra, M. Badi and S. Raj, "Implementation of PSO, It's Variants and Hybrid GWO-PSO for Improving Reactive Power Planning", In Proceedings of 2019 Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2019, pp. 1-6.

[20] A. Pakzad, and M. Analoui, "A Rule-Based/BPSO Approach to Produce Low-Dimensional Semantic Basis Vectors Set", *Turk. J. Electr. Eng. Comp. Sci.*, vol. 30, no. 7, p. 8, 2022.

[21] M. Suhail Shaikh, C. Hua, S. Raj, S. Kumar, M. Hassan, M. Mohsin Ansari and M. Ali Jatoi, "Optimal Parameter Estimation of 1-Phase and 3-Phase Transmission Line for Various Bundle Conductor's Using Modified Whale Optimization Algorithm", *Int. J. Electr. Power Energy Syst.*, vol. 138, p. 107893, 2022.

[22] M. S. Shaikh, S. Raj, M. Ikram, et al. "Parameters Estimation of AC Transmission Line by an Improved Moth Flame Optimization Method", *J. Electr. Syst. Inf. Technol.*, vol 9, p. 25, 2022.

[23] V. Mathur, S. Singh and A. Pundir, "An Investigation of Augment March C - Algorithm Using IoT: Heuristic Approach", *Comput. Technol.*, vol. 29, no. 4, pp. 110-121, 2024.