test pattern set [7]. For the assumed fault model they provide very good resolution.

The second type of approach is based on the effect-cause principle. The algorithms that utilize the effect-cause based approach are observing the actual responses (effects) and determine which fault (cause) might have caused the failure effect which is observed. As the name suggests the effect-cause algorithm directly examines the response of the failing chip and then derives the fault candidates using path-tracing algorithms. Each primary output (PO) is being traced backward so that the error propagation paths can be defined for all possible fault candidates. The effect-cause techniques are more likely to be memory efficient and can be easily integrated in larger designs. Effect-cause analysis can perform both model dependent and model independent diagnosis.

The suspected faults grow exponentially with the number of defects:

$$\textit{Suspected faults} = (\textit{No. of lines})^{(\textit{No. of defects})} \qquad (1)$$

In order to deal with this exponential search space and different failures special diagnostic algorithms for efficient diagnosis are developed. The works [8], [9] have proposed an incremental multiple-fault simulation strategy. Candidate faults are injected sequentially, fault ranking is performed according to the number of pass and fail patterns explained by them. However, the wrong fault chosen at any stage may lead to a faulty solution. The algorithm is inherently based on single fault simulation.

The effect-cause analysis can also be performed by deducing internal signal values in the CUD [10]. Here, the faults were located without knowing the expected output values. A Boolean satiability-based method for multiple-fault diagnosis has been proposed in [11] which handle both combinational and sequential circuits in the same way. The diagnostic resolution achieved by this approach is higher. The framework proposed in [12] deal with several fault models at the same time.

This paper presents a novel effect-cause multiple fault diagnosis approach based on multiple fault simulation and multiple fault injection. In order to explore the exponential search space of multiple fault diagnosis problems, population based searches like Particle Swarm Optimization (PSO) [13] can be used. Initially, a list of possible fault candidates is found out by critical path tracing from each failing primary output and taking a union of them. If there exists a single perfect fault candidate, this method stop and report the result. Otherwise, the faults are arranged in descending order according to the number of test patterns they can explain.

The initial particles of PSO are chosen at random from the possible faulty sites with more priority given to the faults having higher ranks. Since the number of faults in each particle is a variable, each particle is a set of faults with varying cardinality. The PSO output is given as sets of faults, which could successfully explain the entire passing and failing pattern set. The main advantage is that multiple faults can be analyzed simultaneously.

Fault simulation [14] is a more challenging task than logic simulation due to the added dimension of complexity; that is, the behavior of the circuit containing all the modeled faults must be simulated. During single-fault simulation, we transform the model of the fault-free circuit C so that it models the circuit C_F created by a single stuck-at fault f_i and C_F is simulated. Similarly, during multiple-fault simulation, we transform the model of the fault free circuit C so that it models the circuit C_F created by injecting all suspected faults.

The use of manometer technologies increases cross-coupling capacitance and inductance between interconnects, leading to severe crosstalk effects that may result in improper functioning of a chip as illustrated in [15]. Crosstalk effects can be separated to two categories: crosstalk glitches and crosstalk delays. A crosstalk glitch is a pulse that is provoked by coupling effects among interconnects lines. The magnitude of the glitch depends on the ratio of the coupling capacitance to the line-to-ground capacitance. Crosstalk delay is a signal delay that is provoked by the same coupling effects among interconnects lines, but it may be produced even if line drivers are balanced but have large loads.

Capacitive couplings can be treated as potential logic faults. Classical fault models do not cover this class of faults. Conventional fault analysis may be invalid if these effects are not taken into consideration based on the physical layout. PSO is proposed as a multi-objective diagnostic tool by considering crosstalk defects along with other faults.

II. METHODS AND MATERIAL

Multi-Objective PSO Based Multiple Fault Diagnosis

The proposed diagnostic algorithm works on circuits with the primitive gate types AND, OR, NOT, NAND, NOR, XOR, and XNOR, and with fault-free memory elements (D flip−flops). Fig.1. Shows block diagram representation of proposed multi-objective PSO [16] based multiple fault diagnosis. In fault diagnosis process all the tests are executed and every response is stored in external memory and will be used for further analysis. The algorithm starts after testing has failed. The inputs are the logic netlist (specification), and the faulty behavior is given as a set of failing test-vector responses. The objective of diagnosis is to identify logic faults in the netlist that could explain the observed test-vector responses.

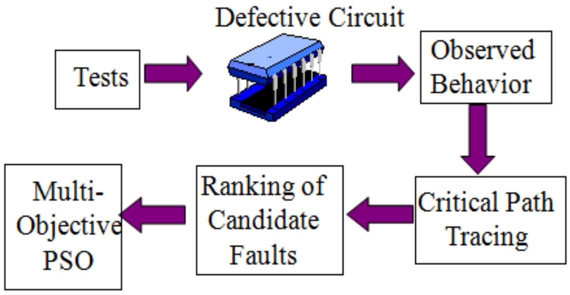

Figure.1: Block Diagram Representation of Proposed Multi-Objective PSO Based Multiple Fault Diagnosis

Critical path tracing (CPT) [17], [12] is a backtracing algorithm which determines the faults detected by a set of tests. It starts at each failing Primary Output (PO) to reach the Primary Inputs (PIs) by tracing each critical line passing through sensitive gate inputs. A gate input i is sensitive if complementing the value of i changes the value of the gate output. In presence of a gate with only nonsensitive inputs, the algorithm stops. Among the possible fault candidates found out by CPT, we try to find a single fault candidate using fault simulation process. If no such single perfect fault candidate exists, candidate faults are arranged according to the number of passed and failed patterns they can explain.

An optimization function with more than one objective is termed as multi-objective optimization function. PSO is a method for the optimization of nonlinear functions. It is developed under the inspiration of behavior of bird flocks, and fish schools. The simple population based technique can be effectively extended for multi-objective optimization. Discrete PSO (DPSO) [18] is used to solve the multiple fault simulation problems.

A. Structure of a Particle

A particle is an m-bit binary array, m being equal to the total number of candidate faults found out by using CPT. A “1” at ith position indicates that the ith fault in the candidate list is present in the circuit, a “0” indicates its absence.

B. Initial Positions of particles

The population size is an important parameter for the performance of the PSO. There is no rule to fix the population size; we have taken the population size to be double the size of the suspected list of faults. Population size is fixed to 1000.

C. Fitness of a Particle

The fitness function or the objective function is the basic performance parameter of a population. The fitness value is bigger, and the performance is better. Fitness function is the inverse of optimization function. In our work, there are two objectives.

1) Objective Function 1: The following terminologies have been used to formulate the objective function.

F: A Boolean array of size equal to the number of collapsed faults present in the circuit. The *i*th element of *F* is given by f_i. f_i is “1” if the ith fault is present, “0” otherwise.
T: The set of test patterns.
C_A: Actual circuit under test.
C_F: The faulty circuit with faults indicated in F.
Simulate (C, t_i): A function which takes a circuit C and a test pattern $t_i \in T$, and returns the output response obtained when t_i is applied to C.
Equal (O_1, O_2): A function that takes two output responses O_1 and O_2 and returns “1” if they are same. Otherwise, it returns “0.”
The optimization function is given by

$$Minimize \sum_{i=1}^{F} f_i \quad (2)$$

Subject to the constraint

$$\sum_{j=1}^{T} Equal\big(Simulate(C_A, t_j), Simulate(C_F, t_j\big) = T \quad (3)$$

Fitness of a particle is calculated in terms of the number of test patterns, the particle can explain. The faults depicted by the particle are injected into the circuit. Both fail and pass patterns are simulated in presence of these faults and the responses are compared with the tester responses. If the two responses for a particular test pattern match, the test pattern is said to be fully explained by the particle. For each particle, the number of test patterns explained by the particle is used as its fitness.

2) Objective Function 2: The second objective is to measure coupling capacitance between different layers and interconnects. Various steps involved in the measurement are

i. Define the material properties permittivity and resistivity of different layers and interconnects.
ii. Create an electrostatic finite capacitor model and assign physics attributes to each region within the model.
iii. Apply boundary conditions and loads.
iv. Calculate the capacitance matrix.

D. Global Best and Particle Best

Global Best (gbest) is the best solution achieved so far. Particle Best (pbest) for each particle is its best solution found so far. Both gbest and pbest are updated after each iteration.

E. Mask Operator and New Position of a Particle

Two mask operators are calculated separately based on pbest and gbest positions. In Fig. 2, an example of mask operator calculation is shown. Mask operator is calculated by comparing bit-by-bit the pbest with the particle's current position. If ith fault is absent/present in both pbest and the particle, ith bit is set to "0," otherwise it is set to "1." After the mask operator has been calculated, a bitwise-XOR is performed between particle's current position and the mask operator. The rest of the bits are found by XOR-ing the current position with the mask operator.

1	1	0	0	1	1	1	pbest
1	0	0	0	1	0	1	Particle position
0	1	0	0	0	1	0	Mask Operator

Figure 2: Example of Mask Operator

After applying the mask operator for pbest, the particle positions go through the same process for their respective gbest also. The final position obtained after applying the gbest mask operator is considered as the particle's new position. The new position of particle i at (k+1)th iteration is calculated as follows

$$P^{i}_{k+1} = (C_1 * I \otimes C_2 * (P_k \rightarrow pbest^i) \otimes C_3 * (P_k \rightarrow gbest^i)) * P^{i}_{k} \quad (4)$$

The operator → represents the mask operator. The operator ⊗ is the fusion operator. The fusion operator applied on two exclusive-or sequences, a ⊗ b is equal to the sequence in which the sequence of exclusive-or operations in a is followed by those in b. The constants C_1, C_2, and C_3 are the inertia, self confidence, and swarm confidence values. For the circuit s5378, it is found that the combination c1 = 1.0, c2 =0.08, and c3 =0.1 produces good results in most of the cases. It can be found that the convergence condition for this DPSO is given by

$$\left(1-\sqrt{C_1}\right)^2 \leq C_2 + C_3 \leq \left(1+\sqrt{C_1}\right)^2 \quad (5)$$

F. Termination Conditions

The algorithm terminates after 30 generations, even if the solution is still improving. If best result that is the particle with maximum fitness is obtained, we stop running the algorithm. We also use a maximum iteration condition, after which we stop, even if solution is still improving.

The pseudo code of the proposed PSO is provided in Algorithm.

```
Algorithm PSO-Pseudocode
begin
    Initialize all particles;
    while Max iterations not reached and gbest has changed in
    last 30 generations do
        for all particles do
            Calculate fitness value;
            if fitness is better than current pbest then
                Update pbest;
            if fitness is better than current gbest then
                Update gbest;
        for all Particles do
            Find mask operators and use it to find the new
            position of particle;
end
```

III. RESULTS AND DISCUSSION

The proposed diagnostic algorithm has been implemented in MATLAB. We have used the full-scan version of s5378 benchmark circuit. All the results are obtained by setting maximum possible faults to six. For each circuit, we have performed 10 random fault injections to get different faulty circuits. The circuits are then simulated using test pattern set generated by an Automatic Test Pattern Generator (ATPG) tool and collected the failure responses. Those failure responses and pass patterns are fed into our algorithm. Our PSO-based diagnostic algorithm gives sets of faults as output.

Efficiency of diagnostic algorithms can be obtained by several parameters. Resolution (Res.) for an algorithm is measured as a ratio of actual faults present in the circuit to the total number of reported fault candidates. Diagnosability (Dia.) of an algorithm is a measure of the fraction of defects that can be correctly identified. The candidate faults identified by the algorithm are arranged in a specific order depending on their probability. First Hit Rank (FHR) compares the ordered list of faults found by the algorithm with the first fault that matches an injected fault. The next step of the diagnostic process is to use a microscope to examine the candidate sites in the reported order.

Table I presents the results obtained by the proposed method for stuck-at-faults for the circuit s5378. The first column indicates the number of faults injected. The parameters: FHR, Diagnosability and Resolution have been reported for the circuit.

TABLE I
RESULTS OBTAINED BY THE PROPOSED APPROACH FOR STUCK-AT-FAULTS

Injected Faults	FHR	Dia.	Res.
1	1.00	1.02	2.10
2	1.05	1.00	1.40
3	1.10	0.96	1.20
4	1.20	0.92	1.50
5	1.25	0.90	1.26
6	1.15	0.91	1.28

Table II gives the results for rising transition faults for the circuit s5378. The proposed approach can handle both rising and failing faults in the circuit simultaneously.

TABLE II
RESULTS OBTAINED BY THE PROPOSED APPROACH FOR TRANSITION FAULTS

Injected Faults	FHR	Dia.	Res.
1	1.00	1.00	2.10
2	1.00	0.98	1.29
3	1.08	0.98	1.11
4	1.13	0.95	1.20
5	1.19	0.95	1.15
6	1.21	0.98	1.25

In PSO, particles start with random positions. So, initially, the gbest and pbest fitness values will be less. After each iteration, the particles will modify their positions and move closer to the optimal solution. In Fig. 3, we have plotted the change in average detection rate with iterations for a problem instance of the circuit s5378. Here, PSO first finds the optimal solution at generation 20 and it remains unchanged for the next iterations.

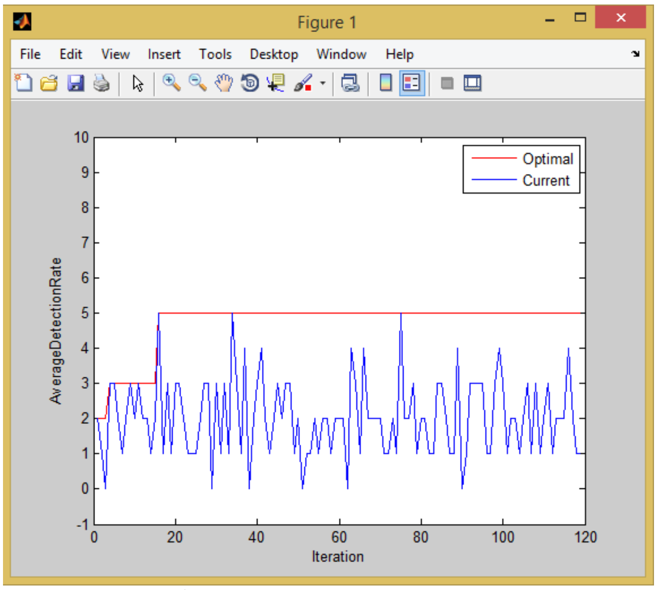

Figure.3: Variation of Average Detection Rate With Iterations.

In Table III, experimental results for crosstalk defects are shown for the circuit s5378.

TABLE III
RESULTS OBTAINED BY THE PROPOSED APPROACH FOR CROSSTALK DEFECTS

Faults	FHR	Dia.	Res.
1	1.93	1.00	1.23
2	1.82	0.85	1.80
3	1.68	0.87	1.28
4	1.15	0.82	1.26
5	1.02	0.78	1.29
6	1.05	0.84	1.25

IV. CONCLUSION

Fault diagnosis is important to reduce the cost and time to market and manufacture chips. An efficient multiple fault diagnosis methodology using multi-objective particle swarm optimization for the diagnosis of crosstalk defects along with conventional faults is proposed. The algorithm has a very high FHR and diagnosability with small resolution compared with previous works. PSO based diagnostic algorithm can be extended for different fault models, such as bridging faults and stuck open faults. Diagnosis of open interconnect fault can be included with the help of model free fault diagnosis. In future, PSO based multi-objective optimization algorithm can be proposed by including the measurement of temperature effect. The use of different mutation operators which act on different subdivisions of the swarm will improve its efficiency. Since, the approach is effect-cause based, different fault models can easily be incorporated into the proposed framework.

V. REFERENCES

[1] H. Takahashi, K. O. Boateng, K. K. Saluja, and Y. Takamatsu, "On diagnosing multiple stuck-at faults using multiple and single fault simulation in combinational circuits," *IEEE Trans. Computer Aided Design Integr. Circuits Syst.*, (Mar. 2002), vol. 21, no. 3, pp. 362–368.

[2] D. B. Lavo, B. Chess, T. Larrabee, and F. J. Ferguson, "Diagnosing realistic bridging faults with single stuck-at information," *IEEE Transactions on Computer Aided Des. Integr. Circuits Syst.*, (Mar. 1998), vol. 17, no. 3, pp. 255–268.

[3] V. J. Mehta, M. Marek-Sadowska, K.H. Tsai, and J. Rajski, "Timing-aware multiple-delay-fault diagnosis," *IEEE Trans. Computer Aided Design Integrated Circuits Syst.*, (Feb. 2009), vol. 28, no. 2, pp. 245–258.

[4] X. Fan, W. Moore, C. Hora, and G. Gronthoud, "Stuck-open fault diagnosis with stuck-at model," *in Proc. 10th IEEE Eur. Symposium* (May 2005), pp. 182–187.

[5] B. Chess and T. Larrabee, "Creating small fault dictionaries", Vol. 18, No. 3, pp.346-356, *IEEE Trans. Computer-Aided Design*, 1999.

[6] D. Lavo and T. Larrabee, "Making cause-effect effective: low-resolution fault dictionaries", *in Proc. Int. Test Conf. (ITC)*, (2001), pp. 278-286.

[7] Y. Higami, K. K. Saluja, H. Takahashi, S. Kobayashi, and Y. Takamatsu, "Compaction of pass/fail-based diagnostic test vectors for combinational and sequential circuits," *in Proc. ASPDAC*, (2006), pp. 75– 80.

[8] Z. Wang, M. Marek-Sadowska, K.-H. Tsai, and J. Rajski, "Analysis and methodology for multiple-fault diagnosis," *IEEE Trans. Computer Aided Design Integr. Circuits Syst* (Mar. 2006), vol. 25, no. 3, pp. 558–575.

[9] J.B. Liu and A. Veneris, "Incremental Fault Diagnosis," IEEE Trans. Computer Aided Design of Integrated Circuits and Systems (Feb. 2005), vol. 24, no. 2, pp. 240-251.

[10] M. Abramovici and M. A. Breuer, "Multiple fault diagnosis in combinational circuits based on an effect-cause analysis*," IEEE Trans. Computers* (June 1980), vol. 29, pp. 451–460.

[11] A. Smith, A. Veneris, M. Ali, and A. Viglas, "Fault diagnosis and logic debugging using Boolean satisfiability," *IEEE Trans. Computer Aided Design Integrated Circuits Syst.* (Oct. 2005), vol. 24, no. 10, pp. 1606–1621.

[12] B. Bosio, P. Girard, S. Pravossoudovitch, and A. Virazel, "A comprehensive framework for logic diagnosis of arbitrary defects," *IEEE Trans. Computers* (Mar. 2010), vol. 59, no. 3, pp. 289–300.

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," *in Proc. IEEE Int.Conf. Neural Netw.*, vol.4. Nov.–Dec. 1995, pp. 1942–1948.

[14] L. Wang, C. Wu, and X. Wen, *"VLSI Test Principles and Architectures: Design for Testability*," (2006), 1st ed. Amsterdam, The Netherlands: Elsevier.

[15] A. Rubio, N. Itazaki, X. Xu, and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits," *IEEE Trans. Computer Aided Des. Integrated Circuits Syst.*(1994), vol. 13, no. 3, pp. 387–395.

[16] M. Reyes-Sierra and C. Coello, "Multi-objective particle swarm optimizers: A survey on the state-of-the-art," *International Journal of Computational Research* (2006), 2(3):287-308.

[17] M. Abramovici, P. R. Menon, and D. T. Miller, "Critical path tracing: An alternative to fault simulation," *in IEEE Design Test Comput. Mag.* (Feb. 1984), vol. 1, pp. 89–93.

[18] K. Wang, L. Huang, C. Zhou, and W. Pang, "Particle swarm optimization for traveling Salesman problem," *in Proc. 2nd Int. Conf. Mach. Learn. Cyberm*. (Nov. 2003), vol. 3, pp. 1583–1585.

[19] Z. Wang, M. Marek-Sadowska, K. H. Tsai, and J. Rajski, "Delay-fault diagnosis using timing information," *IEEE Trans. Computer Aided Design Integrated Circuits Syst.*(Sep. 2005), vol. 24, no. 9, pp. 1315–1325.