UNIVERSITY OF ALBERTA

Analysis of Evolutionary Optimization Algorithms in Automated Test Pattern Generation for Sequential Circuits

KING KHALID UNIVERSITY

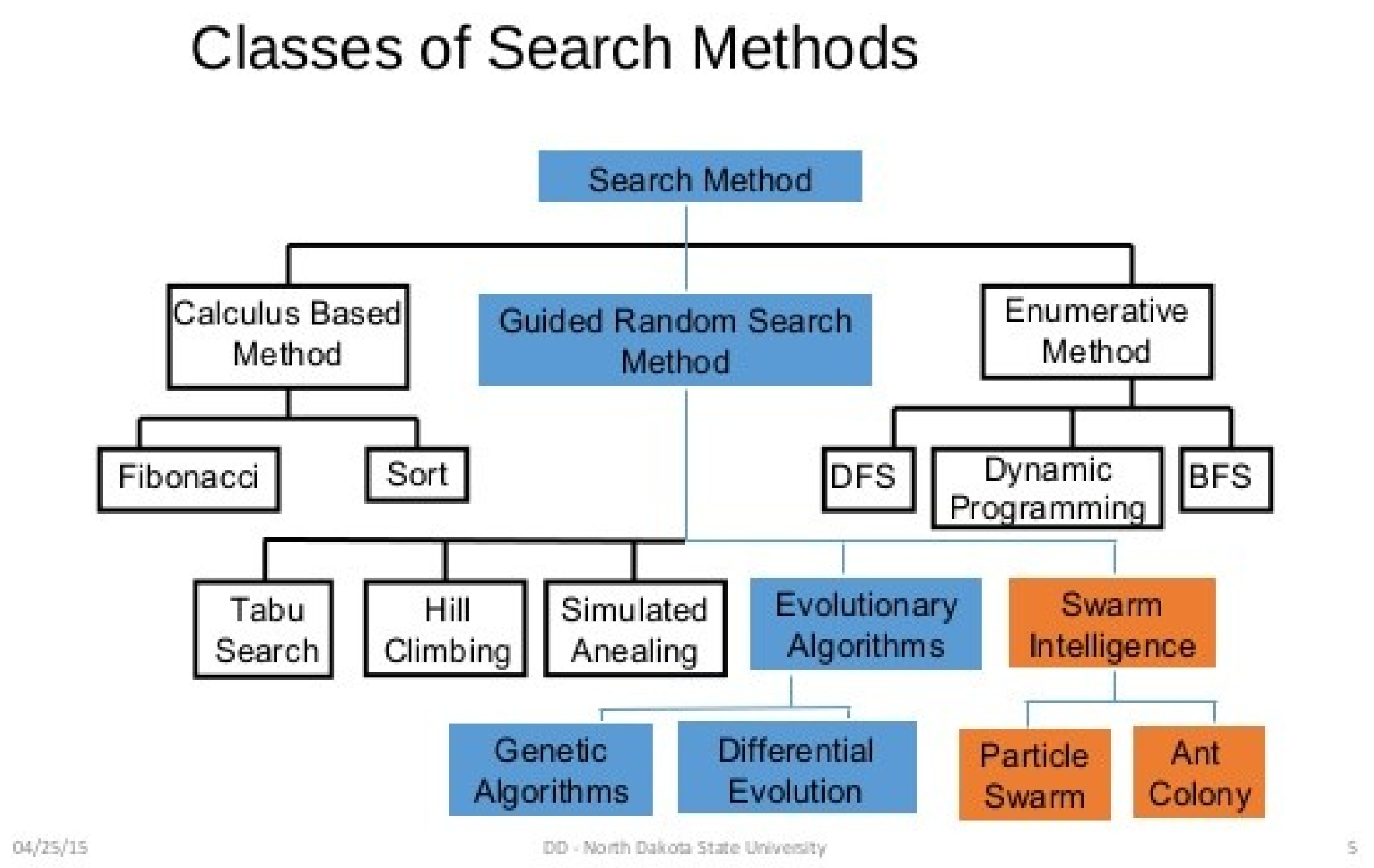
Majed M. Alateeq
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta, Canada
alateeq@ualberta.ca

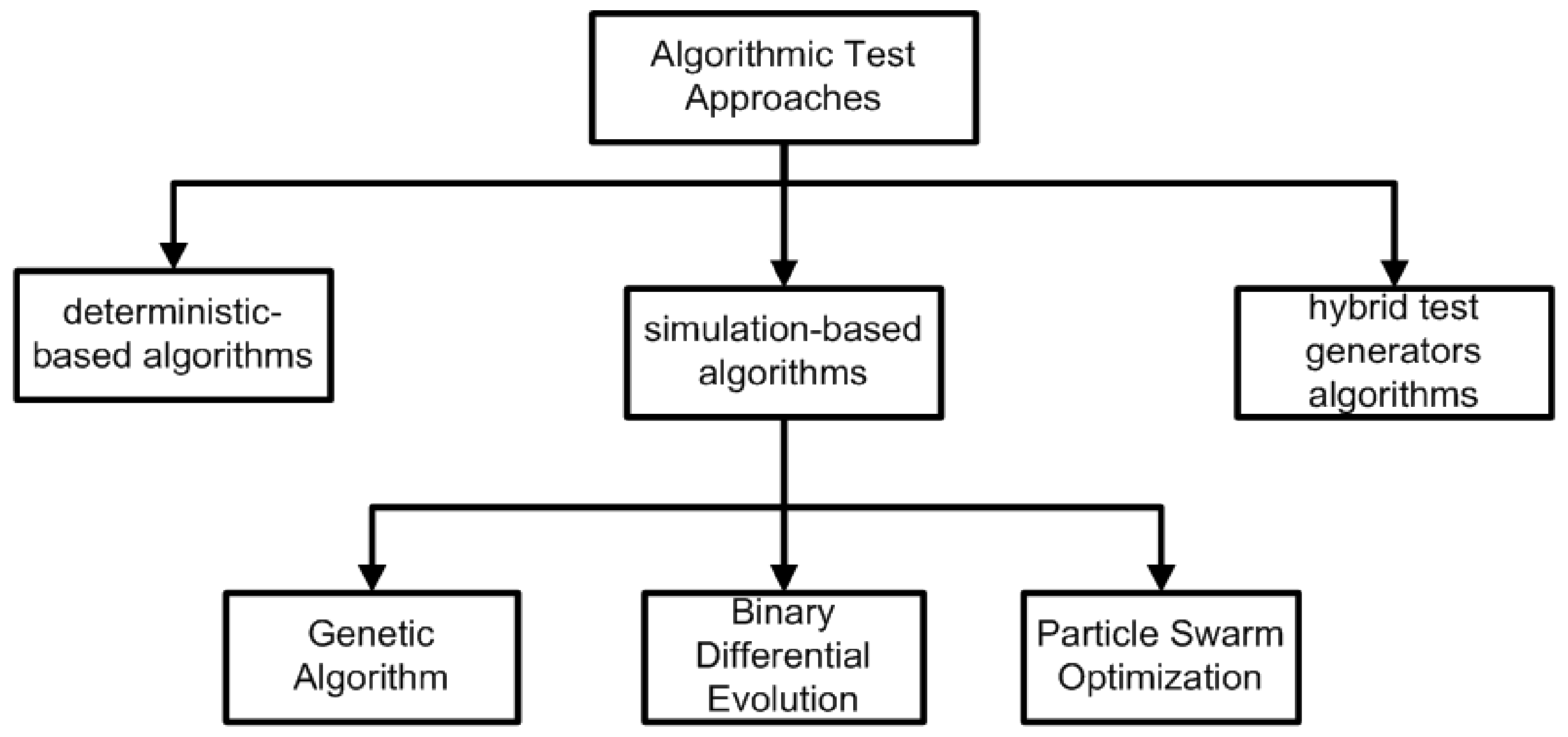
Witold Pedrycz
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta, Canada
wpedrycz@ualberta.ca

Introduction

Evolutionary algorithms are capable of generating and optimizing efficient test vectors for combinational and sequential digital circuits. The basic concept behind these two searching classes is as follows: they start with an initial population of individuals, strings of bits, each bit is mapped to a primary input and an individual is evaluated with a fitness function. Better individuals will evolve through several generations until a stopping criteria has been reached. The incremental complexity of VLSI due to continues increase in number of transistor per chip increases the significance of finding close-to-perfection algorithms to test sequential circuits. A test generation algorithm has to be able to detect faults at the manufacturing level at the earliest point possible to increase yield probability and decrease the cost of production.

In this work, three optimization algorithms, namely: genetic algorithm (GA), particle swarm optimization (PSO) and differential evolution (DE), were studied for the purpose of generating optimized test sequence sets. Furthermore, it investigated the broad use of evolutionary algorithms and swarm intelligence in automated test pattern generation to expand the analysis of the subject. The obtained experimental results demonstrated the improvement in terms of testing time, number of test vectors, and fault coverage compared with previous optimization-based test generators. In addition, the experiments highlight the weakness of each optimization algorithm in the test pattern generation (TPG) and offer some constructive methods of improvement.




Algorithmic Test Approaches
deterministic-based algorithms
simulation-based algorithms
hybrid test generators algorithms
Genetic Algorithm
Binary Differential Evolution
Particle Swarm Optimization

Research Objectives

The overall objective when dealing with ATPG is to find the minimum number of test sequences that detect all testable faults in the shortest test time possible. This project primarily analyzes evolutionary algorithms and swarm intelligence algorithms in ATPG for synchronous sequential circuits by studying in details the parameters of each algorithm and understanding the type of interactions between them. The analysis will lead to determining the optimal values for each optimization algorithm used in this paper.

Implementation

ISCAS89 sequential benchmark circuits were used to be tested in this research because of their high adaption in the literature. The selected circuits range from small-scale circuits to large-scale ones. In this research, we are considering single stuck-at faults model in synchronous sequential circuits under zero gate delay model. HOPE fault simulator is used to simulate each test vector and compute its fitness.

	Circuit	PIs	Sequential Depth	Total Faults	POs
1	S298	3	8	308	6
2	S344	9	6	342	11
3	S349	9	6	350	11
4	S510	19	2	564	7
5	S641	35	6	467	24
6	S713	35	6	581	23
7	S953	16	3	1079	23
8	S1196	14	4	1242	14
9	S35932	35	35	39094	320

A. Genetic Algorithm

The processing realized by the sequential GA-based test generator is divided into two processes and several sub-processes:

1. GA Pre-process:
   - •Initialize all FFs.
   - •Preprocess and partition the circuit.
2. GA-Process
   - •Generate random sequence (vector series) as an initial population.
   - •Compute the fitness of each sequence.
   - •The evolutionary processes of GA are used to generate new population from the existing population.

B. Binary Differential Evaluation

BDE in ATPG for sequential circuits aims to limit the search space in the range 0-1. It starts with a randomly generated initial population and each individual is evaluated through a fitness function. A new population is then generated through the process of crossover and mutation according to a mutation strategy

X_3	X_2	X_1	Results
0	0	0	0
0	0	1	1- X_3 with some probability
0	1	0	1- X_3 with some probability
0	1	1	0
1	0	0	1
1	0	1	1- X_3 with some probability
1	1	0	1- X_3 with some probability
1	1	1	1

C. Particle Swarm Optimization

The initial population is made up of randomly initialized particles. Particles revise their own velocities and positions based on a predefined fitness function of its own and other particles in a population. In other words, a particle modifies its movement according to its own experience and its neighboring particle experience. Position and velocity are to be modified in each iteration of PSO algorithm to find the optimum solution.

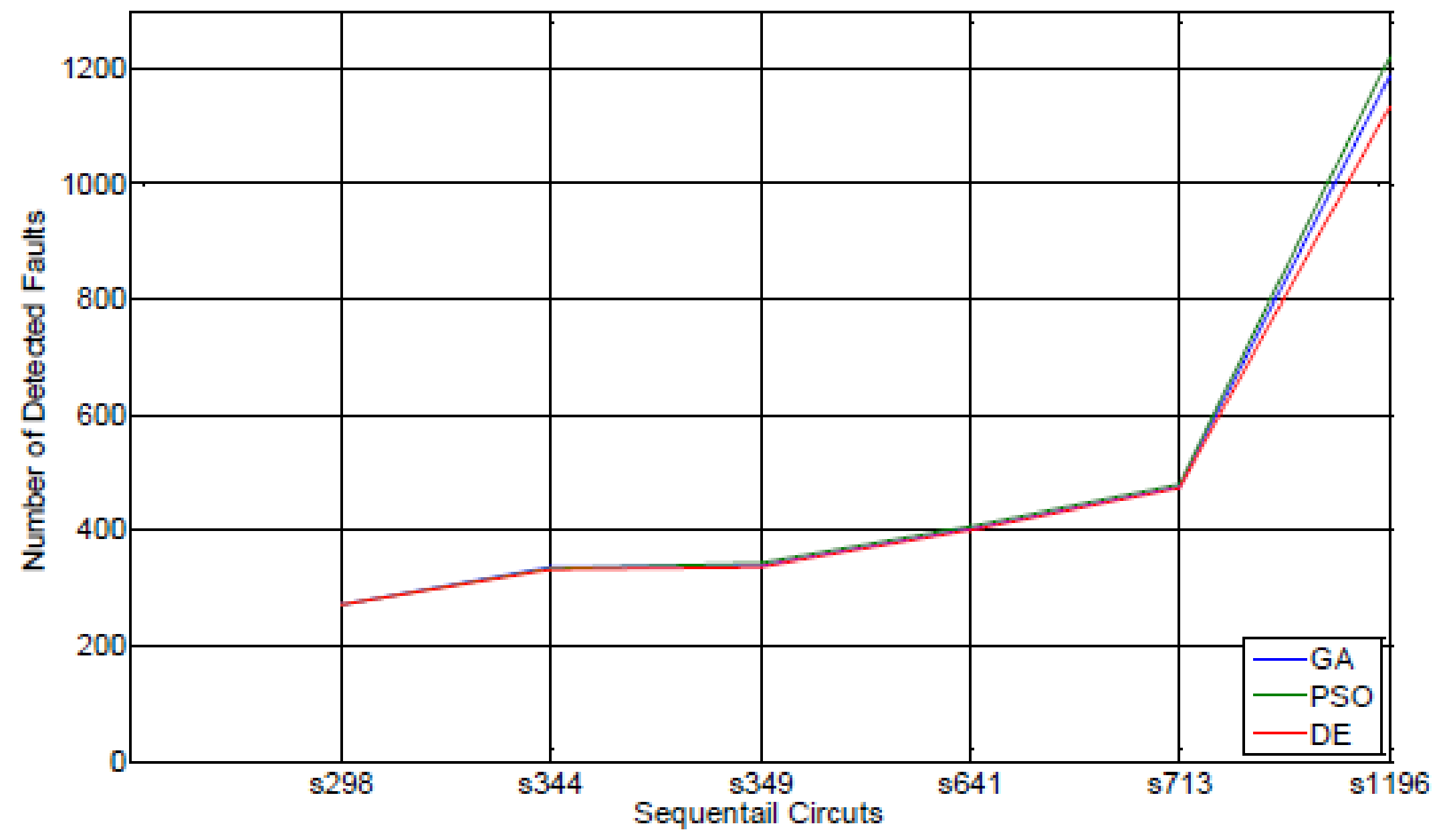
Fitness Function

$$\text{Fitness} = \text{number of faults detected} + \frac{\text{number of faults propogated to flip\_flops}}{(\text{number of faults})(\text{number of flip\_flops})(\text{sequnce length})}.$$

Results

PSO shows superiority in performance over other optimization algorithms and it is expected that other SI algorithms will show similar results because of similarities in the search mechanism which is highly guided since it lets a particle rely on its own experience as well as the whole group experience. In contrast, the GA search mechanism lets individuals move to optimum solutions as a group which leads to incremental increases in testing time which concludes that GA is slower than PSO.

Circuit	Total number of faults	GA		PSO		DE	
		Fault Detected	Number of Vectors	Fault Detected	Number of Vectors	Fault Detected	Number of Vectors
S298	308	272	144	272	142	271	154
S344	342	337	91	335	90	333	89
S349	350	340	90	343	90	338	89
S510	564	564	263	564	239	564	317
S641	467	404	134	407	138	400	142
S713	581	475	119	480	118	474	119
S1196	1242	1188	362	1220	348	1134	567
S35932	39094	34011	197	33730	201	33230	328



BDE must have a proper binary mutation strategy that does not add significant computation overhead when it operates with long test sequences in large sequential circuits. The mutation strategy needs to have a mutant rate that excludes a part of the test sequence from being mutated to reduce the testing time. Otherwise, the testing time will increase rapidly as the length of the test sequence increases.

EAs show an increase in testing time for all sequential circuits while SI shows an optimized testing time and higher fault coverage due to the nature of the searching process. The results of PSO, as a representative of SI, in ATPG raise the significance of implementing other SI-based algorithms in ATPG, such as Ant Colony.

Acknowledgement

The assistance and support provided by The King Khalid University (KKU), Saudi Arabia is fully acknowledged.