SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X VOLUME 3, ISSUE 12, 2025

OPTIMIZED RECONSTRUCTION OF BOOLEAN-VALUED INFORMATION SYSTEM FROM AGGREGATES THROUGH GENETIC ALGORITHM BASED ON PROBABILISTIC INITIAL POPULATION SELECTION(O-BISGA)

Muhammad Sadiq Khan[1], Salman Ali[2], Nizamud Din [3], Attaullah[*4], Muhammad Uzair Khan[5]

[1,3,4,]*Department of Computer Science, University of Chitral, Seen Lasht Chitral, Pakistan*
[2]*Departmentof Computer Science, City University of Science and Technology (CUSIT), Peshawar Pakistan*
[5]*Department of Mathematics University of Buner, Swari, Buner, KP, Pakistan*

uzair@ubuner.edu.pk

DOI: https://doi.org/10.5281/zenodo.23393200

Keywords
Boolean-valued information systems (BIS); Boolean-valued Information Systems using the Genetic Algorithm (BISGA); Incomplete Information; Data Reconstruction;Soft Set

Article History
Received: 01 November 2025
Accepted: 15 December 2025
Published: 31 December 2025

Copyright @Author
Corresponding Author: *
Attaullah, Muhammad Uzair Khan

Abstract
Incomplete Boolean-valued information systems (BIS) provide significant challenges in reconstructing information due to missing or ambiguous components. Whereas the approach of recalculating Boolean-valued Information Systems using the Genetic Algorithm (BISGA) presents a solution by employing aggregate sets and random beginning populations, its reliance on stochastic initialisation typically leads to inefficient convergence and variable accuracy. This study proposes a deterministic, probability-based initial population selection technique to improve BISGA performance. The suggested technique (O-BISGA) distributes 1s to high-probability cells based on weighted averages from row, column, and diagonal aggregates, guaranteeing that chromosomes comply closely to aggregate prerequisites. Experimental findings on BIS matrices (4×4 to 7×7) show a 35% reduction in convergence iterations (8.05 vs. 12.4) compared to BISGA, coupled with increased fitness ratings. The technique reduces randomness-induced inefficiencies while maintaining BISGA's essential limitations, providing a scalable solution for reconstructing incomplete BIS.

1. INTRODUCTION

In an era where data-driven decision-making is the norm, the difficulty of processing ambiguous or insufficient information persists in industries such as healthcare, engineering, and the social sciences [1-3]. Alternative frameworks such as fuzzy sets [4], rough sets [5], and soft sets have been used since normal mathematical procedures such as calculus and statistics frequently fail when dealing with ambiguous information. Among them, Molodtsov's 1999 exposition of soft set theory [6] has grown into a powerful tool for defining parameterised data, especially when using Boolean-valued information systems (BIS).These platforms store data in binary matrices, with objects and properties assessed as 0s or 1s, allowing

for more efficient decision-making [1]. A soft set is employed in numerous real-world applications and decision-making processes [7-9]. However, real-world BIS is usually incomplete because to failures in data input, transportation, or storage, resulting in outcomes that are inaccurate.

Traditional solutions to incomplete BIS are divided into two categories: preprocessed and unprocessed approaches [10]. Preprocessed approaches, such as parity bits [11] and aggregate sets [12, 13], rely on retrieving auxiliary data before reconstructing missing values. While successful for partial data loss, these methods have scalability concerns with bigger systems. Unprocessed approaches, such as weighted averages [14] and probability-based imputation [10, 15], use existing data but struggle with completely missing values. BISGA [16], a significant improvement, recalculates whole missing datasets with aggregate sets and genetic algorithms [17]. However, BISGA's reliance on random beginning populations increases uncertainty in chromosomal fitness, resulting in inefficient convergence and worse accuracy—particularly when BIS dimensions increase.

To overcome these constraints, this paper proposes a structured initial population formation technique for BISGA that replaces random selection with a deterministic, probability-based approach. The O-BISGA starts by creating a zero-matrix that matches the dimensions of the desired BIS. It then computes weighted averages for each cell by dividing row, column, and diagonal aggregate values by their lengths, resulting in a probabilistic distribution of possible 1s and 0s. These averages are added together to form a table, which is then sorted row by row in descending order. To construct high-fitness initial chromosomes, the method systematically distributes 1s to places with the highest probability, starting with values aligned with row aggregates and progressing to columns and diagonals. This process not only minimizes the randomness inherent in BISGA but also ensures that the initial population adheres closely to the constraints imposed by the aggregate sets. The method's efficacy is demonstrated through a manually solved 4×4 BIS example. Implemented in Python, the approach was systematically tested on larger matrices ranging from 4×4 to 7×7. The results indicate a 35% reduction in the average number of iterations required for convergence (8.05 vs. 12.4), effectively addressing the scalability and efficiency limitations of prior methods

Some terminologies used in this paper are similar. For example, a cell of a table is like a gene of a chromosome or table, or BIS, therefore a cell and gene are same and a table or a BIS or a chromosome are also same. Therefore, the readers should not get confused with the use of these terminologies alternatively.

The remainder of this paper is structured as follows: Section 2 reviews soft sets, BIS, and genetic algorithms; Section 3 details the proposed methodology; Section 4 presents experimental results; and Section 5 concludes with future research directions.

2. Preliminaries and Related Works

Preliminaries and works related to this study are categorized into five types in this section.

2.1 Foundational Works in Soft Set Theory

The theoretical groundwork for handling uncertain data was established through soft set theory[6], which provided a framework for parameterized decision-making. Early applications demonstrated how Boolean-valued information systems could represent object-feature relationships through binary matrices[1]. These systems became particularly valuable for problems requiring multi-criteria evaluation, where traditional methods like fuzzy sets showed limitations in parameter handling. The development of parameter reduction techniques addressed computational challenges in large datasets, though these approaches remained constrained by their dependence on complete information systems[18].

2.2 Techniques for Incomplete Boolean-Valued Information Systems

Initially researchers recognised the frequency of missing data in real-world applications, two independent methodological methods evolved for dealing with incomplete Boolean-valued information systems. Preprocessed category

approaches, such as parity bits [11] and aggregate sets [12, 13], provided novel methods for preserving and recovering data. The idea of aggregate sets was especially important, with row aggregates, column aggregates, and diagonal aggregates offering several dimensions for data reorganisation. The unprocessed category strategies provided alternate solutions using methods such as weighted average [14] and probability-based approaches [19, 20]. These approaches worked directly on available data and did not require pre-extracted information, making them more adaptive to unanticipated data loss circumstances. However, their usefulness was reduced when faced with significant data gaps, since they lacked methods for comprehensive system reconstruction. The distinction between these two types becomes critical in comprehending the trade-offs between preparation overhead and adaptive capabilities in incomplete data processing.

2.3 Advancements in Entire System Reconstruction

The area made tremendous progress with the advent of specialised approaches for reconstructing completely Boolean-valued information systems. The idea of universal, empty, and hybrid diagonals introduced new methods for determining system completeness. Universal diagonals, where the total equalled the cell count, and empty diagonals, with a zero sum, provided unambiguous reconstruction routes in particular instances. Hybrid diagonals posed more difficult issues, necessitating repetitive techniques that occasionally resulted in computing bottlenecks. These developments laid the groundwork for more sophisticated reconstruction algorithms but also highlighted the need for more efficient computation methods[12].

2.4 Genetic Algorithms in Computational Optimization

Genetic algorithms[17] emerged as powerful metaheuristics for complex optimization problems[21-23], inspired by biological evolution principles. These algorithms operate through generational improvement of candidate solutions, employing three core operations: selection[24, 25] identifies promising solutions based on fitness evaluation, crossover[26] combines features of parent solutions, and mutation[27] introduces controlled randomness. The adaptability of genetic algorithms made them particularly suitable for problems with large search spaces and multiple local optima, though their effectiveness heavily depended on proper parameter tuning and initialization strategies[2, 3, 28].

2.5 BISGA Implementation and Limitations

The BISGA adapted these evolutionary principles specifically for data reconstruction[16]. BISGA uses the random selection for the initial population, due to which the fitness of chromosomes varies between the best and worst fit. When a chromosome is worse fit, there will be more iteration for making it best fit and accuracy and efficiency decrease. There will be the issue of more iteration to get the solution when the size of the BIS's increases, and finding the original one became very difficult due to increase in the number of equivalent BIS. These are the identified drawbacks of the approach.

It is worth mentioning that by applying the O-BISGAfor the initial population creation the iterations become less and the efficiency and accuracy of the required result enhanced.

3. Proposed Methodology

As the proposed approach focuses on the selection of chromosomes that have high fitness compared to BISGA. We explain it with the help of an example that how chromosomes with high fitness are selected through proposed technique of probability. Consider a 6×6 BIS given in Table 1, the BIS has sixobjects, and each object has six parameters represented by capital alphabets from A to F. Table 1 also has its columns and row aggregates calculated by summing its number of 1 for each column and row respectively.

Table 1: A sample6×6BIS with the aggregates of rows and columns

U|P	A	B	C	D	E	F	Row Aggregate
1	1	1	0	0	1	0	3
2	0	1	1	0	0	0	2
3	1	1	0	1	1	1	5
4	0	0	1	1	0	0	2
5	1	0	1	0	1	0	3
6	1	1	0	1	0	1	4
Column Aggregates	4	4	3	3	3	2	

Left to Right and Right to Left diagonals are calculated for sample BIS in Table 2 and Table 3 respectively. Same color for each diagonal with its aggregate may help the readers in following a diagonal easily.

Table 2: Left to Right Diagonal Aggregates for sample BIS of Table 1

U|P	A	B	C	D	E	F	Left to Right Diagonal Aggregates
1	1	1	0	0	1	0	
2	0	1	1	0	0	0	0
3	1	1	0	1	1	1	1
4	0	0	1	1	0	0	1
5	1	0	1	0	1	0	1
6	1	1	0	1	0	1	3
							5
							2
							3
							0
							2
							1

Table 3: Right to Left Diagonal Aggregates for sample BIS of Table 1

Right to Left Diagonal Aggregates	A	B	C	D	E	F	U|P
	1	1	0	0	1	0	1
1	0	1	1	0	0	0	2
1	1	1	0	1	1	1	3
2	0	0	1	1	0	0	4
2	1	0	1	0	1	0	5
2	1	1	0	1	0	1	6
3							
4							
1							

2		
0		
1		

To summarize Table 1, Table 2 and Table 3, the following four aggregate set(1)is obtained

C = (4, 4, 3, 3, 3, 2),
R = (3, 2, 5, 2, 3, 4), ⋯⋯⋯ Aggregates Set (1)
LR = (0, 1, 1, 1, 3, 5, 2, 3, 0, 2, 1),
RL = (1, 1, 2, 2, 2, 3, 4, 1, 2, 0, 1)

where C represents column aggregates, R represents rows aggregates, LR represents left to right diagonal aggregates and RL represents right to left diagonal aggregates. The purpose is to recalculate the original BIS from the sets of these aggregatesin set (1) through GA while the proposed method focuses on the selection of chromosomes that have relatively better fitness than BISGA. How BISGA will select chromosomes for this BIS is illustrated in below example.

Example 1: Chromosomes selection for BIS of Table 1 using BISGA

BISGA will define the size of chromosomes based on the size of columns (C) and rows (R) by seeing the sets of aggregates in set (1). In this case |C| = 6 and |R| = 6, it means that the size of BIS to be calculated is a 6×6 table. According to BISGA, the BIS will be filled with random numbers of 0s and 1s but an initial constraint of at least one aggregate satisfaction will be imposed on these random numbers. Suppose we impose the initial constraint on rows aggregate satisfaction; it means that our BIS must satisfy the row aggregate for all its chromosomes as two chromosomes are given in Table 4 and Table 5.

Table 4: Chromosome (1) selected through BISGA and satisfying row constraint

U|P	A	B	C	D	E	F	Row Aggregate
1	1	0	1	0	1	0	3
2	1	0	0	1	0	0	2
3	0	1	1	1	1	1	5
4	0	1	1	0	0	0	2
5	1	1	0	1	0	0	3
6	1	0	1	1	1	0	4
Column Aggregate	4	3	4	4	3	1	

Set of aggregates for chromosome 1 are given below

C = (4, 3, 4, 4, 3, 1),
R = (3, 2, 5, 2, 3, 4), ⋯⋯⋯ Aggregates set (2)
RL = (0, 1, 1, 3, 1, 2, 5, 2, 2, 1, 1),
LR = (1, 1, 1, 1, 5, 4, 1, 3, 1, 1, 0)

To calculate the fitness of chromosome (1), first, we must subtract the number of each aggregate in the original set (1) from its corresponding aggregate in the obtained set (2). Then we must add the absolute values of all aggregates after subtraction. For set (1), C is (4, 4, 3, 3, 3, 2) while for set (2) C is (4, 3, 4, 4, 3, 1), therefore its absolute difference is (0, 1, 1, 1, 0, 1). R is same for both Set (1) and set (2)because we have deliberately imposed this initial constraint on chromosome (1), therefore, its absolute difference is 0. Similarly, the absolute difference for RL is (0,

0, 0, 2, 2, 3, 3, 1, 2, 1, 0) and the absolute difference for LR is (0, 0, 1, 1, 3, 1, 3, 2, 1, 1, 1). Summing the absolute differences of C, LR and RL gives us 31 as the fitness of chromosome (1). Closer this fitness to zero means more fittest chromosomes and a fitness equal to zero means a hundred percent fittest chromosome.

Table 5: Chromosome (2) selected through BISGA and satisfying row constraint

U|P	A	B	C	D	E	F	Row Aggregate
1	1	0	0	1	0	1	3
2	0	0	0	1	0	1	2
3	1	1	0	1	1	1	5
4	1	0	1	0	0	0	2
5	1	1	0	0	0	1	3
6	1	1	0	1	1	0	4
Column Aggregate	5	3	1	4	2	4	

Aggregate's set for chromosome (2) is also given below:

C= (5, 3, 1, 4, 2, 4)
R = (3, 2, 5, 2, 3, 4) ~~~~~~~ Aggregates set (3)
LR = (1, 1, 2, 2, 2, 1, 3, 2, 2, 2, 1)
RL = (1, 0, 1, 3, 2, 5, 3, 1, 1, 2, 0)

Fitness for chromosome (2) is calculated in Table 6 using the method already explained for chromosome 1. The fitness of chromosome (2) is 30 which is one time more fit than chromosome (1) having fitness as 31.

Table 6: Fitness calculation for chromosome (2)

Aggregate set	Column Aggregate	Row Aggregate	LR Aggregate	RL Aggregate
Set (1)	(4,4,3,3,3,2)	(3,2,5,2,3,4)	(0,1,1,1,3,5,2,3,0,2,1)	(1,1,2,2,2,3,4,1,2,0,1)
Set (3)	(5,3,1,4,2,4)	(3,2,5,2,3,4)	(1,1,2,2,2,1,3,2,2,2,1)	(1,0,1,3,2,5,3,1,1,2,0)
Absolute Difference of corresponding aggregates	(1,1,2,1,1,2)	(0,0,0,0,0,0)	(1,0,1,1,1,4,1,1,2,0,0)	(0,1,1,1,0,2,1,0,1,2,1)
Total of each aggregates set	8	0	12	10
Total Fitness	30			

After explaining the chromosomes selection of BISGA in example 1. Now, we are going to discuss the selection of more fit chromosomes through the proposed approach. In addition to one aggregate initial constraint of BISGA, we implement another constraint of probability on selection of more fit chromosomes. Instead of randomly assigning 0s and 1s, we assign it through using a probabilistic method. In O-BISGA the weight of each aggregate is calculated first for each cell in the BIS which is called genes in GA. To calculate the weight for an aggregate, the total number of 1s in that aggregate is divided by the total length of that aggregate and put in the respective gene/cell of the

chromosome. As each gene is part of all four aggregates, the respectiveweight of each aggregate is added in each gene for finding the probability of 1s and 0s in each gene. A gene is assigned 1s according to the initial constraint of any one BISGA but instead of random assignment unlike BISGA the O-BISGA assigns 1s to the highest probability genes and the remaining genes are filled with 0s. This way, a more fit chromosome is obtained unlike BISGA. This process is explained step by step in the example below.

Example 2: selecting the chromosomes through O-BISGA.

Consider again the aggregates set (1), first we calculate the weight for each aggregate of 1st gene of object (1) having parameter A as given in Table 7. In aggregates set (1), the first row has three 1s while the length of the row is 6. Dividing the number of 1s by row length gives us 0.5 which is entered first in the first gene. Similarly, there are four 1st in first column so dividing it by column length gives us 0.67 which is added to the first calculated weight in same gene. For left to right diagonal, five out of six are 1s in the diagonal and the ratio become 0.83, which is also added to the first two weights of the same cell. Similarly, the right to left diagonal has one length and its value is 1, yielding in weight 1, which is added to the previous three weight resulting in 3 as a grand weight of this cell. Similarly, the weights of all other genes are calculated as shown in Table 7 and represented in Table 8 separately.

Table 7: Weights calculation for each gene through probability

U | P	A	B	C	D	E	F
1	0.5+0.67+0.83 +1=3	0.5+0.67+0.6 +0.5=2.27	0.5+0.5+0.25 +0.67=1.92	0.5+0.5+0.33 +0.5=1.83	0.5+0.5+0.5+ 0.4=1.9	0.5+0.33+0+0. 5=1.33
2	0.33+0.67+0.4 +0.5=1.9	0.33+0.67+0. 83+0.67=2.5	0.33+0.5+0.6 +0.5=1.93	0.33+0.5+0.2 5+0.4=1.48	0.33+0.5+0.3 3+0.5=1.67	0.33+0.33+0.5 +0.8=1.66
3	0.83+0.67+0.7 5+0.67=2.92	0.83+0.67+0. 4+0.5=2.4	0.83+0.5+0.8 3+0.4=2.56	0.83+0.5+0.6 +0.5=2.43	0.83+0.5+0.2 5+0.8=2.38	0.83+0.33+0.3 3+0.25=1.74
4	0.33+0.67+0+ 0.5=1.5	0.33+0.67+0. 75+0.4=2.15	0.33+0.5+0.4 +0.5=1.73	0.33+0.5+0.8 3+0.8=2.46	0.33+0.5+0.6 +0.25=1.68	0.33+0.33+0.2 5+0.67=1.58
5	0.5+0.67+1+0. 4=2.57	0.5+0.67+0+0 .5=1.67	0.5+0.5+0.75 +0.8=2.55	0.5+0.5+0.4+ 0.25=1.65	0.5+0.5+0.83 +0.67=2.5	0.5+0.33+0.6+ 0=1.43
6	0.67+0.67+1+ 0.5=2.85	0.67+0.67+1+ 0.8=3.14	0.67+0.5+0+ 0.25=1.42	0.67+0.5+0.7 5+0.67=2.59	0.67+0.5+0.4 +0=1.57	0.67+0.33+0.8 3+1=2.83
Sequence of Probability Calculation: Row + Colum+ LR Diagonal +RL Diagonal						

We can see in Table 8 that each gene has its own weight which reflects its probability of 1s in that cell aggregate wise. We combine this probability with the initial constraint of an aggregate in our approach to get more fit chromosome. In other words, the initial constraint of BISGA used to fix the number of 1s for a specific diagonal while the O-BISGA also specifies the gene positions of those 1s in that diagonal. Suppose we combine row constraint with probability of each row, then according to row aggregate R = (3, 2, 5, 2, 3, 4) of aggregate set (1),the first row has three 1s. The top three probabilities of first row in Table 8 are first, second and third cellshaving the values 3, 2.27 and 1.92 respectively. Therefore, we assign 1s to the first three cells of our new BIS as shown in Table 9 and fill the remaining genes with 0s. Similarly, for second row there should be two 1s according to row constraint and the top two cells are second and last having 2.5 and 1.96 probabilities respectively. Therefore, we assign 1s to second and last cell of second row and fill the remaining cells of same row with 0s. In Same way, the remaining rows are filled as given in Table 9.

Table 8: Probability of 1s according to aggregate set (1)

3	2.27	1.92	1.83	1.9	1.33
1.9	2.5	1.93	1.48	1.66	1.96
2.92	2.4	2.56	2.43	2.38	1.74
1.5	2.15	1.73	2.46	1.68	1.58
2.57	1.67	2.55	1.65	2.5	1.43
2.84	3.14	1.42	2.59	1.57	2.83

Table 9: Chromosome (p) satisfying row constraint calculated through O-BISGA

U|P	A	B	C	D	E	F	Row Aggregate
1	1	1	1	0	0	0	3
2	0	1	1	0	0	0	2
3	1	1	1	1	1	0	5
4	0	1	0	1	0	0	2
5	1	0	1	0	1	0	3
6	1	1	0	1	0	1	4
Column Aggregates	4	5	4	3	2	1	

Aggregates for chromosome (p) calculated through O-BISGA are given below aggregate set (4)

R = (3, 2, 5, 2, 3, 4)
C = (4, 5, 4, 3, 2, 1) ---------- Aggregates set (4)
LR = (0, 0, 0, 2, 3, 6, 1, 4, 0, 2, 1)
RL = (1, 1, 3, 2, 3, 2, 4, 0, 2, 0, 1)

The fitness of chromosome (p) is calculated in Table 10 with its fitness equal to 14.

Table 10: Fitness calculation for Chromosome (p) selected through O-BISGA

Aggregate set	Column Aggregate	Row Aggregate	LR Aggregate	RL Aggregate
Set (1)	(4,4,3,3,3,2)	(3,2,5,2,3,4)	(0,1,1,1,3,5,2,3,0,2,1)	(1,1,2,2,2,3,4,1,2,0,1)
Set (4)	(4,5,4,3,2,1)	(3,2,5,2,3,4)	(0,0,0,2,3,6,1,4,0,2,1)	(1,1,3,2,3,2,4,0,2,0,1)
Absolute Difference of corresponding aggregates	(0,1,1,0,1,1)	(0,0,0,0,0,0)	(0,1,1,1,0,1,1,1,0,0,0)	(0,0,1,0,1,1,0,1,0,0,0)
Total of each aggregates set	4	0	6	4
Total Fitness	14			

Now, we select another chromosome (q) using the proposed method that satisfies another constraint of LR.Considering the probability Table 8, the chromosome (q) constructed through O-BISGA satisfying LR aggregates is given in Table 11.

Table 11: Chromosome (q) satisfying LR constraint calculated through proposed approach

U|P	A	B	C	D	E	F	Row Aggregate
1	1	1	0	1	0	0	3
2	1	1	1	0	0	1	4
3	1	1	1	1	1	0	5
4	0	0	0	0	0	0	0
5	1	0	1	0	1	0	3
6	1	1	0	1	0	1	4
Column Aggregates	5	4	3	3	2	2	

Aggregates for chromosome (q) calculated through O-BISGA are given in aggregate set (5).

R = (3, 4, 5, 0, 3, 4)
C = (5, 4, 3, 3, 2, 1) ~~~~~~~ Aggregates set (5)
LR = (0, 1, 1, 1, 3, 5, 2, 3, 0, 2, 1)
RL = (1, 2, 2, 3, 2, 2, 4, 0, 2, 0, 1)

The fitness of chromosome (q) is calculated in Table 12 with its fitness equal to 12.

Table 12: Fitness calculation for Chromosome (q) selected through O-BISGA

Aggregate set	Column Aggregate	Row Aggregate	LR Aggregate	RL Aggregate
Set (1)	(4,4,3,3,3,2)	(3,2,5,2,3,4)	(0,1,1,1,3,5,2,3,0,2,1)	(1,1,2,2,2,3,4,1,2,0,1)
Set (5)	(5,4,3,3,2,1)	(3,4,5,0,3,4)	(0,1,1,1,3,5,2,3,0,2,1)	(1,2,2,3,2,2,4,0,2,0,1)
Absolute Difference of corresponding aggregates	(1,0,0,0,1,1)	(0,2,0,2,0,0)	(0,0,0,0,0,0,0,0,0,0,0)	(0,1,1,1,0,1,0,1,0,0,0)
Total of each aggregates set	3	4	0	5
Total Fitness	12			

It is worth noting that zero fitness of LR aggregate is due to the initial constraint imposed on LR aggregate in chromosome (q).

In Example 1, we generated two chromosomes through BISGA which have the fitness of 31 and 29, while in Example 2 we generated two chromosomes through proposed method O-BISGA which have the fitness of 14 and 12. These examples not only explain the mechanisms of O-BISGA and BISGA but also show the efficiency of O-BISGA over BISGA. On average, BISGA generated chromosomes of 30 fitness while O-BISGA generated chromosomes of 13 fitness, which shows that generating fittest chromosomes, O-BISGA is 56% more efficient than BISGA in these examples.

After these explanations, we are going to present the algorithm for O-BISGA as given Algorithm -1.

Algorithm- 1 O-BISGA

Input: Aggregate of rows, aggregate of columns, aggregate of left-right diagonal, aggregate of left-right diagonal.

Output: Chromosomes with high fitness

1. Create a table of the size of the solution with number of rows equal to size of row aggregateset and number of columns equal to set of column aggregateset.
2. Find the average values for each aggregate, by dividing each aggregate value by the length of aggregate.
3. Add the average values to the corresponding cell of the table to get the probability of 1s in each gene.
4. Insert 1s in the cells having highest probability values according to count present in row aggregate set and insert 0s to the rest of the genes.
5. Repeat Step 4 for column, left-right diagonal and right-left diagonal aggregate sets (to get three more chromosomes)

4. Results and Discussions

We implemented bothO-BISGA and BISGA in Python and created chromosomes of different BIS sizes through it.Chromosomes for BIS of 4×4, 5×5, 6×6, and 7×7 were generated tentimes through BISGA and O-BISGA. The fitness of each chromosome generated through these experiments is given in Table 13.

Table 13: Fitness obtained from experiments through BISGA and proposed method

BIS Size	Fitness Calculated Through	Number of Experiments										Average
		1	2	3	4	5	6	7	8	9	10	
4×4	BISGA	6	8	12	16	10	8	4	14	4	12	9.4
	O-BISGA	0	4	8	10	4	0	0	6	0	0	3.2
5×5	BISGA	10	18	12	14	8	4	4	14	4	12	10
	O-BISGA	8	10	8	10	4	0	0	6	0	8	5.4
6×6	BISGA	10	20	22	14	8	10	8	14	12	24	14.2
	O-BISGA	8	10	16	10	4	8	6	8	8	16	9.4
7×7	BISGA	10	24	22	12	18	10	14	14	16	20	16
	O-BISGA	8	26	22	8	20	8	8	6	14	22	14.2
Average Fitness of BIS of all sizes					BISGA							12.4
					O-BISGA							8.05

The average results show that the proposed approach generates 35% more fittest chromosomes than BISGA on average.These results are alsographically shown in Figure 1
.

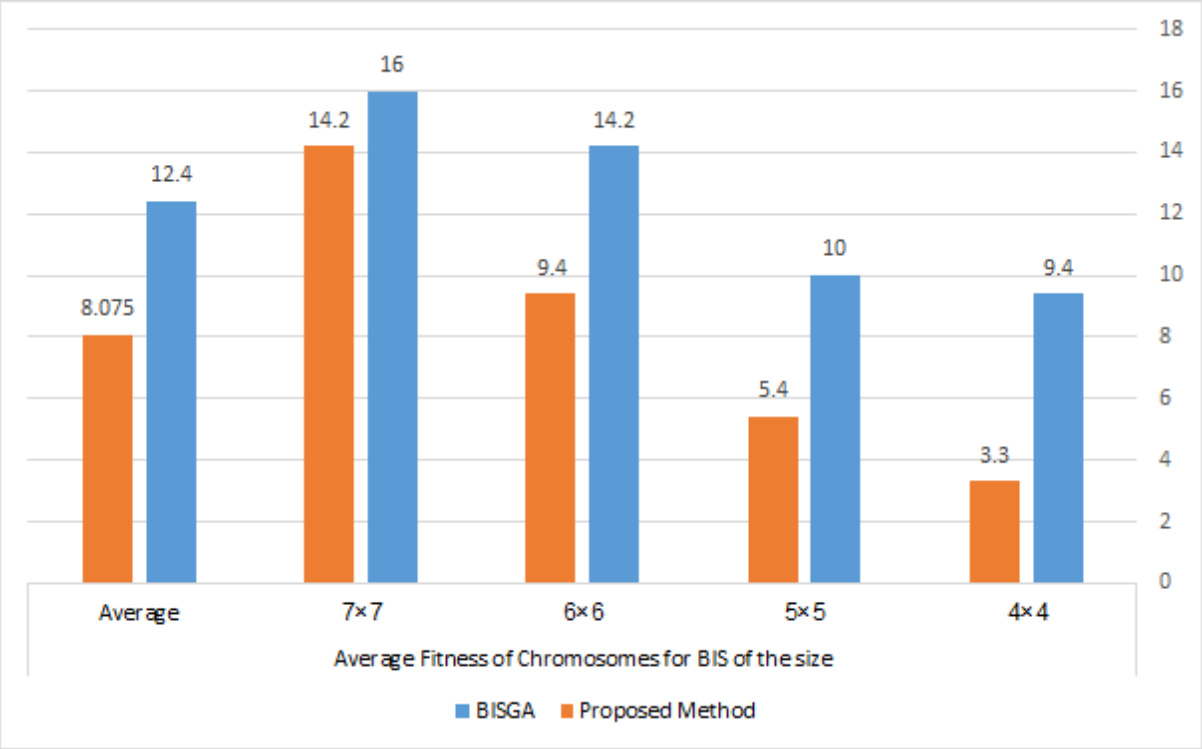


Figure 1: Graphical representation of the Fitness of O-BISGA in comparison to BISGA

It is obvious from the results that proposed method generates more fittest chromosomes for BISof 7×7 size and less, therefore the GA will take less iterations to converge and find the solution quickly.

After the presentation of these results, we are going to present a few discussions related to BISGA and O-BISGA which may be useful and interesting for the readers. Firstly, as BISGA is based on random chromosomes generation with initial constraint of one aggregate satisfaction. This random assignment of value varies from best to worst fitness therefore in some cases BISGA may yield better results than O-BISGA. Three such cases can be observed in the 2nd, 5th and 10th experiments of 7×7 BIS in Table 13. The purpose of the O-BISGA is to minimize the generation of the worst chromosomes and maximize the probability of the best ones which is reflected in all results other than the mentioned three cases.

Secondly, the initial constraint on one aggregate proposed by BISGA is among the major contributions of BISGA and without this constraint the random assignment of 0s and 1s would have generated more worst chromosomes as usually generated by different GAs. This constraint has fixed the number of 0s and 1s in any chromosome and it can be observed in all sets of aggregates that the number of 1s are 19 in for each set and chromosome. The proposed method has maintained this powerful constraint of BISGA and has tried to maximize fitness.

Lastly, it can also be observed in the results that the performance of O-BISGA in selecting the best chromosomes decreases from small size chromosomes to large size. This is the weakness of this approach, and it needs to be overcome in future. Interestingly, this weakness of O-BISGA is identical to the weakness of BISGA, and same like O-BISGA, BISGA is unable to recalculate larger than 7×7 entire BIS from aggregates.

5. Conclusions and Future Work

This study addresses the limitations of BISGA by replacing its random initial population generation with a structured probabilistic method, significantly improving convergence efficiency and chromosome fitness. Key contributions include:

1. **Deterministic Initialization**: Weighted averages derived from aggregate sets guide 1s/0s placement, reducing randomness and enhancing fitness.
2. **Performance Gains**: Experiments show a 35% reduction in iterations and higher-fitness chromosomes across BIS sizes (4×4 to 7×7).
3. **Constraint Preservation**: The method retains BISGA's aggregate constraints while minimizing suboptimal solutions.

Future work should address scalability for larger BIS matrices and explore hybrid techniques to further optimize reconstruction accuracy. This advancement bridges the gap between probabilistic rigor and evolutionary computation, offering a robust framework for incomplete data recovery.

Data Availability: All the data are available in the manuscript.

Author contributions: The authors Muhammad Sadiq Khana, Salman Ali, Nizamud Din, and Attaullah contributed equally to the manuscript.

Competing interests: The authors declare no competing interests.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Funding declaration: There is no funding source available for this research.

REFERENCES

[1] P. K. Maji, A. R. Roy, and R. Biswas, "An application of soft sets in a decision-making problem," Computers & mathematics with applications, vol. 44, pp. 1077-1083, 2002.

[2] P. R. Srivastava and T.-h. Kim, "Application of genetic algorithm in software testing," International Journal of software Engineering and its Applications, vol. 3, pp. 87-96, 2009.

[3] B. Tripathy, T. Sooraj, and R. N. Mohanty, "Application of soft set in game theory," in Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction, ed: IGI global, 2019, pp. 421-435.

[4] J. A. Goguen, "LA Zadeh. Fuzzy sets. Information and control, vol. 8 (1965), pp. 338–353.-LA Zadeh. Similarity relations and fuzzy orderings. Information sciences, vol. 3 (1971), pp. 177–200," The Journal of Symbolic Logic, vol. 38, pp. 656-657, 1973.

[5] Z. Pawlak, "Rough sets," International journal of computer & information sciences, vol. 11, pp. 341-356, 1982.

[6] D. Molodtsov, "Soft set theory—first results," Computers & mathematics with applications, vol. 37, pp. 19-31, 1999.

[7] T. Fujita and F. Smarandache, "An introduction to advanced soft set variants: Superhypersoft sets, indetermsuperhypersoft sets, indetermtreesoft sets, bihypersoft sets, graphicsoft sets, and beyond," Neutrosophic Sets and Systems, vol. 82, pp. 817-843, 2025.

[8] A. SezgíN, A. O. Atagün, and N. Cagan, "A complete study on and-product of soft sets," Sigma journal of engineering and natural sciences, vol. 43, pp. 1-14, 2025.

[9] A. Sezgin and E. Şenyiğit, "A new product for soft sets with its decision-making: Soft star-product," Big data and computing visions, vol. 5, pp. 52-73, 2025.

[10] M. Sadiq Khan, M. A. Al-Garadi, A. W. A. Wahab, and T. Herawan, "An alternative data filling approach for prediction of missing data in soft sets (ADFIS)," SpringerPlus, vol. 5, pp. 1-20, 2016.

[11] A. N. Mohd. Rose, H. Hassan, M. I. Awang, T. Herawan, and M. M. Deris, "Solving incomplete datasets in soft set using parity bits of supported sets," in International Conference on Ubiquitous Computing and Multimedia Applications, 2011, pp. 33-43.

[12] M. S. Khan, T. Herawan, A. W. A. Wahab, G. Mujtaba, and M. A. Al-Garadi, "Concept of entire Boolean values recalculation from aggregates in the preprocessed category of incomplete soft sets," IEEE Access, vol. 5, pp. 11444-11454, 2016.

[13] A. N. M. Rose, H. Hassan, M. I. Awang, N. A. Mahiddin, H. M. Amin, and M. M. Deris, "Solving incomplete datasets in soft set using supported sets and aggregate values," Procedia Computer Science, vol. 5, pp. 354-361, 2011.

[14] Y. Zou and Z. Xiao, "Data analysis approaches of soft sets under incomplete information," Knowledge-Based Systems, vol. 21, pp. 941-945, 2008.

[15] H. Qin, X. Ma, T. Herawan, and J. M. Zain, "DFIS: A novel data filling approach for an incomplete soft set," International Journal of Applied Mathematics and Computer Science, vol. 22, pp. 817-828, 2012.

[16] S. Ali, M. S. Khan, H. Shah, H. Garg, and A. Alsheddy, "BISGA: Recalculating the Entire Boolean-Valued Information System from Aggregates Using a Genetic Algorithm," Complexity, vol. 2023, p. 1539563, 2023.

[17] J. H. Holland, "Genetic algorithms," Scientific american, vol. 267, pp. 66-73, 1992.

[18] K. Zhi, W. Jia, G. Zhang, and L. Wang, "Normal parameter reduction in soft set based particle swarm optimization algorithm," Applied mathematical modeling, vol. 39, pp. 4808-4820, 2015.

[19] Z. Kong, Q. Lu, L. Wang, and G. Guo, "A simplified approach for data filling in incomplete soft sets," Expert Systems with Applications, vol. 213, p. 119248, 2023.

[20] Z. Kong, J. Zhao, L. Wang, and J. Zhang, "A new data filling approach based on probability analysis in incomplete soft sets," Expert Systems with Applications, vol. 184, p. 115358, 2021.

[21] K. Pandey and D. Sharma, "Novel image encryption algorithm utilizing hybrid chaotic maps and Elliptic Curve Cryptography with genetic algorithm," Journal of Information Security and Applications, vol. 89, p. 103995, 2025.

[22] A. Thompson and J. Suomalainen, "GAOR: Genetic Algorithm-Based Optimization for Machine Learning Robustness in Communication Networks," Network (2673-8732), vol. 5, 2025.

[23] D. Waysi, B. T. Ahmed, and I. M. Ibrahim, "Optimization by nature: A review of genetic algorithm techniques," The Indonesian Journal of Computer Science, vol. 14, 2025.

[24] A. Shukla, H. M. Pandey, and D. Mehrotra, "Comparative review of selection techniques in genetic algorithm," in 2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE), 2015, pp. 515-519.

[25] T. Pencheva, K. Atanassov, and A. Shannon, "Modelling of a roulette wheel selection operator in genetic algorithms using generalized nets," International Journal Bioautomation, vol. 13, p. 257, 2009.

[26] A. J. Umbarkar and P. D. Sheth, "Crossover operators in genetic algorithms: a review," ICTACT journal on soft computing, vol. 6, 2015.

[27] S. M. Lim, A. B. M. Sultan, M. N. Sulaiman, A. Mustapha, and K. Y. Leong, "Crossover and mutation operators of genetic algorithms," International journal of machine learning and computing, vol. 7, pp. 9-12, 2017.

[28] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia tools and applications*, vol. 80, pp. 8091-8126, 2021.