Solution of 16-Queens Problem using 3-variable Affine Boolean functions (2^{n+1})

Ranjeet Kumar Rout
Amity School of Engineering and Technology
Amity University,Uttar Pradesh-201313
Email: ranjeetkumarrout@gmail.com

Shubham
Samsung Research Institute,
Uttar Pradesh-201301, India
Email: shubhamjindal_1994@yahoo.com

Jayanta Kumar Das
Applied Statistics Unit
Indian Statistical Institute,
203, B.T. Road kolkata-700108
Email: dasjayantakumar89@gmail.com

Pabitra Pal Choudhury
Applied Statistics Unit
Indian Statistical Institute,
203, B.T. Road kolkata-700108
Email: pabitrapalchoudhury@gmail.com

***Abstract*—The present paper focuses on solution to 16-Queens problem highlights the process of placing the 16-affine functions of 3-variable Boolean function as Queen. The classification procedure reported in [1] describes a method to find a solution for the 16-Queens Problem. A complete method has been devised for finding some possible solutions of 16-Queen problems using iterative method instead of backtracking.**

I. Introduction

Computer science problems are grouped into various classes on the basis of time and space taken to find all possible solutions. All the decision problems having polynomial time algorithm are placed in the P class. Some of the problems solved in non-deterministic computer or verified in polynomial time are placed in NP class. The theoretical computers are non-deterministic in nature, which are used to verify the solution of a problem in polynomial time. This type of computer does not exist [2] but this type of theoretical computer consist of infinite amount of asset to generate many processes depending upon the number of possible solution [2][3]. In order to accept the problems in NP, several heuristic methods are used [3][4].

Introduced by Gauss [5] (by taking $N = 8$) the N-Queens problem is about putting N-queens on an $N \times N$ chessboard in a way such that each row and each column of the chessboard have exactly one queen and each diagonal of the chessboard have at most one queen. In this arrangement no two queens are mutually attacking. A 16-Queens problem is a specific problem of the N-Queens problem where $N = 16$, i.e, 16 Queens need to placed in the 16×16 chessboard and the constraints of the N-Queens problem needs to be followed taking $N = 16$. Several mechanisms have already been proposed to solve the N-Queens problem like brute-force, backtracking, Ant Colony Optimization (ACO), Genetic Algorithm, Case Based Approach etc [6]. Generally, N-Queens problems are treated as intractable problem i.e. when the values of N is considered very large at that time the problems are not solved in polynomial time. This type of problems are placed in NP class. Only 92 solutions out of $4,426,165,368$ possible arrangements are exact in case of eight queen problem [2].

The ant colony optimization (ACO) is a meta-heuristic that is propelled by savvy practices of ants. ACO is a multi agent framework; a subterranean insect conduct portrays the conduct of an operator in the framework. The main insect calculation was produced in [7]. Ants in nature alter their environment by continually saving a concoction substance called pheromone. The pheromone is utilized as a circuitous correspondence among ants and aides them to find briefest way from their home to sustenance place. On the off chance that there are various ways to a nourishment place, ants pick one with high grouping of pheromone. An efficient algorithm has been proposed for N queen problem by the authors [2], which works efficiently for eight queen problem as the proposed model is very simple and the search space is very small in a given chess board.

Genetic algorithm is based on the three biological principals of selection, crossover and mutation for solving of searching and optimization problems. Some of the potential solutions are taken into consideration into the next level based on the fitness function defined for a problem being optimized. Then selection operator is applied to find certain individuals that survive into next generation. After selection process individuals are combined using crossover operation. A mutation operator is applied to change few individuals.

Multiple references having the same concept of genetic algorithm are there while having been implemented in different ways. In [8], it uses some form of parallelism to achieve greater speed of genetic algorithm, using a global parallel genetic algorithm which is one of the variants to genetic algorithm. The authors in [8] improved the efficiency of GA by using local search algorithm like minimal conflict algorithm.

Authors in [9] solved N-queen problem on General purpose computing on graphics processing units $(GPGPU)$ architecture using $OpenCL$ which extensively analyze the N-Queen problem with respect to local [10], global memory parameters and atomicity and synchronization issues in $OpenCL$ and its effects on performance. An efficient algorithm using local search and various heuristic searching techniques have been discussed in [11]. Various applications of N queen problems in

978-1-5386-1719-9/18/$31.00

real life has been discussed in [12], such as image processing, dead lock prevention, parallel processing etc.

The paper is organized in a precise methodical manner in the following sections. In Section II, the literature of Boolean functions of different variables relevant to our work is reviewed. In Section III, the method for solving 16-queen problem is organized. Section IV deals with concluding remarks accenting the main factors of the integral analysis.

II. Relevant review

An n-variable Boolean function f is a mapping from the set of all possible n-bit strings $\{0,1\}^n$ into $\{0,1\}$. For n-variable there are 2^{2^n} number of different n-variable Boolean functions. Each Boolean function can be represented with the help of truth table output as a binary string having length 2^n. The decimal equivalent of the binary string starting from bottom to top(least significant bit) in the truth table is called the rule number of that function [1][13]. The complement of f is denoted as $\bar{f}$.

A Boolean function with algebraic expression, where the degree is at most one is called an affine Boolean function. The general form for n-variable affine function is:

$f_{affine}(x_1, x_2, x_3, \ldots, x_n) = k_n x_n \oplus k_{n-1}x_{n-1} \oplus \cdots \oplus k_2x_2 \oplus k_1x_1 \oplus k_0,$ where the co-efficients are either zero or one.

If the constant term k_0 of an affine function is zero then the function is called a *linear Boolean function*. Thus, affine Boolean functions are either linear Boolean function or their complements. The number of different n-variable affine Boolean functions is 2^{n+1} out of which 2^n are linear. As an example, the 16 *affine Boolean functions* in 3-variables are 0, 60, 90, 102, 150, 170, 204, 240, 15, 51, 85, 105, 153, 165, 195, and 255 out of which, first eight are linear and remaining Boolean functions are their corresponding complements[1]. Here each affine functions are considered as queens.

In [1][13], the procedure of classification of Boolean functions of n-variable is obtained in such a coherent way that each class is of equal cardinality and contains only a single affine Boolean function. Two methods have been discussed. The first one is a recursive method used to classify 1-variable to n-variable using Cartesian and concatenation method [1]. The second method is based on changing of some predefined bit positions with respect to a affine Boolean function in that class [1]. Here, affine function are uniformly distributed in all the classes has been reported in [1]

III. Proposed method for solving 16-queen problem using classification of Boolean functions

The 16-Queens problem is the problem of placing of 16 queens on a chessboard that no two queens attack mutually. Thus, a solution mandates that no two queens are in the same row, column or diagonal. This problem cant be solved in polynomial time as the problem becomes intractable for large values of n and thus it is placed in NP class type of problem as was discussed in the introduction. Generally time complexity of this types of problems is exponential. In 3-variable Boolean function there are numbers of Boolean function. As per the classification, 3-variable Boolean functions have 16 numbers of classes and each class have 16 numbers of Boolean functions along with an affine Boolean function in each [1]. After sorting the Boolean functions in increasing order of the 16 classes we arranged in a chessboard as shown in Figure 1. From the Fig1, it has been observed that any two affine Boolean functions (indicated by red cell) are not in same row.

Classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	2	4	6	8	10	12	14	32	34	36	38	40	42	44	46
2	128	130	132	134	136	138	140	142	160	162	164	166	168	170	172	174
3	192	194	196	198	200	202	204	206	224	226	228	230	232	234	236	238
4	64	66	68	70	72	74	76	78	96	98	100	102	104	106	108	110
5	208	210	212	214	216	218	220	222	240	242	244	246	248	250	252	254
6	80	82	84	86	88	90	92	94	112	114	116	118	120	122	124	126
7	16	18	20	22	24	26	28	30	48	50	52	54	56	58	60	62
8	144	146	148	150	152	154	156	158	176	178	180	182	184	186	188	190
9	209	211	213	215	217	219	221	223	241	243	245	247	249	251	253	255
10	81	83	85	87	89	91	93	95	113	115	117	119	121	123	125	127
11	17	19	21	23	25	27	29	31	49	51	53	55	57	59	61	63
12	145	147	149	151	153	155	157	159	177	179	181	183	185	187	189	191
13	1	3	5	7	9	11	13	15	33	35	37	39	41	43	45	47
14	129	131	133	135	137	139	141	143	161	163	165	167	169	171	173	175
15	193	195	197	199	201	203	205	207	225	227	229	231	233	235	237	239
16	65	67	69	71	73	75	77	79	97	99	101	103	105	107	109	111

Fig. 1. Shows all 3-variable Boolean functions are within a 16 × 16 Chess Board with Affine Boolean function as Red colored

In this section, a method is proposed to evaluate different solutions with switch over of class positions within the 16×16 of chess board. The following steps are used for generating solutions of 16 queen problem.

Classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	2	4	6	8	10	12	14	32	34	36	38	40	42	44	46
2	193	195	197	199	201	203	205	207	225	227	229	231	233	235	237	239
3	81	83	85	87	89	91	93	95	113	115	117	119	121	123	125	127
4	144	146	148	150	152	154	156	158	176	178	180	182	184	186	188	190
5	145	147	149	151	153	155	157	159	177	179	181	183	185	187	189	191
6	80	82	84	86	88	90	92	94	112	114	116	118	120	122	124	126
7	192	194	196	198	200	202	204	206	224	226	228	230	232	234	236	238
8	1	3	5	7	9	11	13	15	33	35	37	39	41	43	45	47
9	208	210	212	214	216	218	220	222	240	242	244	246	248	250	252	254
10	17	19	21	23	25	27	29	31	49	51	53	55	57	59	61	63
11	129	131	133	135	137	139	141	143	161	163	165	167	169	171	173	175
12	64	66	68	70	72	74	76	78	96	98	100	102	104	106	108	110
13	65	67	69	71	73	75	77	79	97	99	101	103	105	107	109	111
14	128	130	132	134	136	138	140	142	160	162	164	166	168	170	172	174
15	16	18	20	22	24	26	28	30	48	50	52	54	56	58	60	62
16	209	211	213	215	217	219	221	223	241	243	245	247	249	251	253	255

Fig. 2. Shows all 3-variable Boolean functions are within a 16 × 16 Chess Board with Affine Boolean function as Red colored and the classes are renamed as per the position of affine function within that class

- *Step* I Sort the Class $1, 2, 3, \ldots, 16$ in ascending order.
- *Step* II Arrange all the classes of 3-variable Boolean functions in a 16 × 16 Chess board. All classes are assigned to a Class number according to the position of affine function in that class as shown in Fig 2
- *Step* III The resultant rows of 16 × 16 chess board is as follows:
  - a) $Row_i = Class\, k$, where $i = 1, 3, \ldots, 15$ and $k = 9, 10, \ldots, 16$
  - b) $Row_i = Class\, k$, where $i = 2, 4, \ldots, 16$ and $k = 1, 2, \ldots, 8$

Here, the Fig 3 with yellow colored Affine functions is a solution. similarly other solutions are shown in Fig 4 with blue colored Affine functions, in and Fig 5 with different solution.

Classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	208	210	212	214	216	218	220	222	240	242	244	246	248	250	252	254
2	0	2	4	6	8	10	12	14	32	34	36	38	40	42	44	46
3	17	19	21	23	25	27	29	31	49	51	53	55	57	59	61	63
4	193	195	197	199	201	203	205	207	225	227	229	231	233	235	237	239
5	129	131	133	135	137	139	141	143	161	163	165	167	169	171	173	175
6	81	83	85	87	89	91	93	95	113	115	117	119	121	123	125	127
7	64	66	68	70	72	74	76	78	96	98	100	102	104	106	108	110
8	144	146	148	150	152	154	156	158	176	178	180	182	184	186	188	190
9	65	67	69	71	73	75	77	79	97	99	101	103	105	107	109	111
10	145	147	149	151	153	155	157	159	177	179	181	183	185	187	189	191
11	128	130	132	134	136	138	140	142	160	162	164	166	168	170	172	174
12	80	82	84	86	88	90	92	94	112	114	116	118	120	122	124	126
13	16	18	20	22	24	26	28	30	48	50	52	54	56	58	60	62
14	192	194	196	198	200	202	204	206	224	226	228	230	232	234	236	238
15	209	211	213	215	217	219	221	223	241	243	245	247	249	251	253	255
16	1	3	5	7	9	11	13	15	33	35	37	39	41	43	45	47

Fig. 3. Shows all 3-variable Boolean functions are within a 16 × 16 Chess Board with Affine Boolean function as yellow colored with one solution.

Classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	3	5	7	9	11	13	15	33	35	37	39	41	43	45	47
2	209	211	213	215	217	219	221	223	241	243	245	247	249	251	253	255
3	192	194	196	198	200	202	204	206	224	226	228	230	232	234	236	238
4	16	18	20	22	24	26	28	30	48	50	52	54	56	58	60	62
5	80	82	84	86	88	90	92	94	112	114	116	118	120	122	124	126
6	128	130	132	134	136	138	140	142	160	162	164	166	168	170	172	174
7	145	147	149	151	153	155	157	159	177	179	181	183	185	187	189	191
8	65	67	69	71	73	75	77	79	97	99	101	103	105	107	109	111
9	144	146	148	150	152	154	156	158	176	178	180	182	184	186	188	190
10	64	66	68	70	72	74	76	78	96	98	100	102	104	106	108	110
11	81	83	85	87	89	91	93	95	113	115	117	119	121	123	125	127
12	129	131	133	135	137	139	141	143	161	163	165	167	169	171	173	175
13	193	195	197	199	201	203	205	207	225	227	229	231	233	235	237	239
14	17	19	21	23	25	27	29	31	49	51	53	55	57	59	61	63
15	0	2	4	6	8	10	12	14	32	34	36	38	40	42	44	46
16	208	210	212	214	216	218	220	222	240	242	244	246	248	250	252	254

Fig. 4. Shows all 3-variable Boolean functions are within a 16 × 16 Chess Board with Affine Boolean function as Blue colored with one solution.

IV. Conclusion

In this work a solution to 16-Queens Problem has been discussed, In the process of placing the 16-affine functions of 3-variable Boolean function as Queen. The classification procedure describes a method to find a solution for the 16-Queens Problem. A method has been devised for finding some possible solutions of 16-Queen problems using iterative method instead of backtracking. In future endeavor, this classification strategy can be followed in an indepth study in this paradigm for generating all possible solution to 16-queens problem using iterative instead of backtracking.

Classes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	193	195	197	199	201	203	205	207	225	227	229	231	233	235	237	239
2	144	146	148	150	152	154	156	158	176	178	180	182	184	186	188	190
3	80	82	84	86	88	90	92	94	112	114	116	118	120	122	124	126
4	1	3	5	7	9	11	13	15	33	35	37	39	41	43	45	47
5	17	19	21	23	25	27	29	31	49	51	53	55	57	59	61	63
6	64	66	68	70	72	74	76	78	96	98	100	102	104	106	108	110
7	128	130	132	134	136	138	140	142	160	162	164	166	168	170	172	174
8	209	211	213	215	217	219	221	223	241	243	245	247	249	251	253	255
9	0	2	4	6	8	10	12	14	32	34	36	38	40	42	44	46
10	81	83	85	87	89	91	93	95	113	115	117	119	121	123	125	127
11	145	147	149	151	153	155	157	159	177	179	181	183	185	187	189	191
12	192	194	196	198	200	202	204	206	224	226	228	230	232	234	236	238
13	208	210	212	214	216	218	220	222	240	242	244	246	248	250	252	254
14	129	131	133	135	137	139	141	143	161	163	165	167	169	171	173	175
15	65	67	69	71	73	75	77	79	97	99	101	103	105	107	109	111
16	16	18	20	22	24	26	28	30	48	50	52	54	56	58	60	62

Fig. 5. Shows all 3-variable Boolean functions are within a 16 × 16 Chess Board with Affine Boolean functions with different color.

References

[1] R. K. Rout, P. Pal Choudhury, and S. Sahoo, "Classification of boolean functions where affine functions are uniformly distributed," *Journal of Discrete Mathematics*, vol. 2013, 2013.

[2] S. Khan, M. Bilal, M. Sharif, M. Sajid, and R. Baig, "Solution of n-queen problem using aco," in *Multitopic Conference, 2009. INMIC 2009. IEEE 13th International*, pp. 1–5, IEEE, 2009.

[3] S. Khan, M. Bilal, M. Sharif, M. Sajid, and R. Baig, "Solution of n-queen problem using aco," in *Multitopic Conference, 2009. INMIC 2009. IEEE 13th International*, pp. 1–5, IEEE, 2009.

[4] I. Martinjak and M. Golub, "Comparison of heuristic algorithms for the n-queen problem," in *Information Technology Interfaces, 2007. ITI 2007. 29th International Conference on*, pp. 759–764, IEEE, 2007.

[5] J. Ginsburg, "Gausss arithmetization of the problem of 8 queens," *Scripta math*, vol. 5, pp. 63–66, 1939.

[6] M. Bozikovic, M. Golub, and L. Budin, "Solving n-queen problem using global parallel genetic algorithm," in *EUROCON 2003. Computer as a Tool. The IEEE Region 8*, vol. 2, pp. 104–107, IEEE, 2003.

[7] M. Dorigo, "Optimization, learning and natural algorithms," *Ph. D. Thesis, Politecnico di Milano, Italy*, 1992.

[8] J. E. A. Heris and M. A. Oskoei, "Modified genetic algorithm for solving n-queens problem," in *Intelligent Systems (ICIS), 2014 Iranian Conference on*, pp. 1–5, IEEE, 2014.

[9] K. Thouti and S. Sathe, "Solving n-queens problem on gpu architecture using opencl with special reference to synchronization issues," in *Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on*, pp. 806–810, IEEE, 2012.

[10] K. Thouti and S. Sathe, "Solving n-queens problem on gpu architecture using opencl with special reference to synchronization issues," in *Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on*, pp. 806–810, IEEE, 2012.

[11] M. Kovac, "N-queens problem by efficient non-backtracking algorithm using local search, heuristics and tabu search elements," in *Digital Technologies (DT), 2014 10th International Conference on*, pp. 159–163, IEEE, 2014.

[12] M. Kovac, "N-queens problem by efficient non-backtracking algorithm using local search, heuristics and tabu search elements," in *Digital Technologies (DT), 2014 10th International Conference on*, pp. 159–163, IEEE, 2014.

[13] R. K. Rout, P. P. Choudhury, S. Sahoo, and C. Ray, "Partitioning 1-variable boolean functions for various classification of n-variable boolean functions," *International Journal of Computer Mathematics*, vol. 92, no. 10, pp. 2066–2090, 2015.