

# A GENETIC ALGORITHM APPROACH FOR MINIMIZING TOTAL INVENTORY COST IN A JOB-SHOP MANUFACTURING UNIT

Kapil Kumar Gupta<sup>1</sup>, Deepak Singh<sup>2</sup>, Prashant Kumar Singh<sup>3</sup>

<sup>1</sup>Assistant Professor, Dept. of Mechanical Engineering, MCSIEM Lucknow, U.P., India

<sup>2</sup>Assistant Professor, Dept. of Mechanical Engineering, MCSCET Lucknow, U.P., India

<sup>3</sup>Ph.D. Scholar, MNNIT Allahabad, U.P., India

## Abstract

Inventories are idle resources which are maintained by every organisation for efficient and smooth running of its operations. So here we are going to control the inventory cost by minimizing the inventory cost. For maintaining optimum inventory, one must consider the appropriate reorder point as well as the economic order quantity. We propose an efficient genetic algorithm approach to find economic order quantity at a proper reorder point. Genetic algorithm is an optimization method used for solving multi-variable optimization problems. Job-shops handle a variety of jobs, where each job is different and different inventory levels are to be maintained. A case study of a power plant where different types of raw materials are used. The proposed approach is tested on C-programme.

**Keywords:** Inventory, Reorder point, Economic order quantity, Genetic algorithm, Optimization

-----\*\*\*-----

## 1. INTRODUCTION

Inventory is a stock of items or resources used in an organisation, stock in hand at a given time and acts as a reserve of goods for future demand[11]. Determination of ordering time and ordering quantity is the main objective of inventory management and include appropriate safety stock for minimizing the stock-outs.

A sufficient supply of items without too much over supply is known as inventory control and one thing is that preventing the supply chain members from getting affected by excess as well as shortage of inventory by controlling the inventory in an effective manner to decrease the total inventory cost.

Here in this study the reorder point technique and the economic order quantity to control the inventory are considered. The reorder point is a point when we replenishment our stock (inventory) before comes down to zero for preventing shortage stock.

Genetic algorithm is randomized search technique which is based on evolutionary technique [10]. Starting from an initial set of solutions, we generate new solutions by applying search operators like crossover and mutation. These operators are applied to randomly selected solutions from current set of solutions, where the selection probability is proportional to the solutions objective function value [2].

## 2. LITERATURE REVIEW

P. Radhakrishnan et. al.[1] have discussed the effective and efficient approach that works on Genetic Algorithms in order to determine the most probable excess stock level and shortage level required for Inventory optimization in the supply chain such that the total supply chain cost is

minimized. S. Sakeel Ahmad et. al.[7] have proposed a method to find out the optimized ordering quantity at proper reorder point by using genetic algorithm approach. They considered some raw materials from the power plant and GA is implemented in C - programme. Mr. S. Godwin barnabas et. at.[3] have analysed the problem to determine optimal allocation of spare parts of auto mobile sector with replacement of defective parts. In this paper they tried to solve a problem of spare parts management system, concentrated to find optimal demand for a given spare parts management system to determine optimal inventory level in order to reduce cost.

## 3. A CASE STUDY

### 3.1 Data of Raw Materials

In thermal powerplant; many raw materials comes in working process. Here we have selected some of raw materials in the proposed work [7]. The raw materials that are used are given below

1. Al cladding sheet
2. MS Electrodes
3. SS Electrodes
4. Fire proof wool
5. Gaskets
6. Steam seals
7. Bearings.

### 3.2 Demand Matrix

The quantity of raw materials which are used in different months in a year is shown in table 1.

**Table 1: Demand of Raw material**

Raw material / month	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1 <sup>st</sup>	100	100	100	100	150	150	200	250	350	300	100	100
2 <sup>nd</sup>	5000	5000	5000	5000	6000	6000	10000	12000	12000	10000	2000	2000
3 <sup>rd</sup>	1000	2000	2000	1500	2500	2500	4500	4000	4500	3500	1000	1000
4 <sup>th</sup>	100	50	50	50	50	50	150	150	150	100	50	50
5 <sup>th</sup>	5	3	5	5	4	1	12	10	10	15	5	5
6 <sup>th</sup>	2	2.5	2.5	4	5	2	2	2	1	2	2.5	2.5
7 <sup>th</sup>	200	200	250	250	200	150	200	150	100	100	50	150

### 3.3 Various Cost of Raw Materials

The purchasing, holding, ordering costs of various material are shown in table 2.

**Table 2: Cost components**

Raw material	Purchasing Cost	Holding Cost	Ordering Cost
Al cladding sheet	200	10	2
MS Electrodes	3.65	0.18	4
SS Electrodes	21.60	1.08	6
Fire proof wool	173	9	2
Gaskets	365	18.25	8
Steam seals	3230	161.50	2
Bearings	405	20.25	3

### 3.4 Population Generation and Chromosome Representation

Fitness function which is a part of genetic algorithm is used to calculate the quantity of each chromosome within the population for economic order quantity and reorder point. The population consists of group of individuals called chromosomes. Each of chromosome represents a complete solution of the problem.

In our problems there are 7 raw materials which will have to deliver in all 12 months. So here we represent every months with a different genes and 12 genes made a chromosome. Each gene consists of 7 numerical value because we have 7 raw materials. Now these 7 numerical values are to be encoded to binary (0,1). So when we place 1 will represents ordering cost and when we place 0 will represents holding cost[4].

So initially 12 chromosomes are generated randomly which will make (12\*12 = 144) genes and is shown in table 3.

**Table 3: Randomly generated chromosomes**

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1	0001001	1100011	1101001	0011001	0110111	0011111	0110000	0101001	1010001	0101100	0101100	0110110
2	0101001	0110110	1111101	0110111	0101000	1110101	0011010	1100100	1111010	1100100	1100100	1111101
3	1000100	0001100	1011101	0111110	0010101	0100011	0111111	0111100	0011111	1010110	1010110	0100111
4	1010100	0100010	0101100	0010000	0010111	1100010	0101100	0000101	1010110	0000001	0000001	1011010
5	0110100	0010100	1001011	1001011	0110110	0011000	1111001	0010111	0100101	1001101	1001101	1010101
6	1101001	0100100	0001110	0011101	0001100	0100101	1110101	1011100	1010110	0110010	0110010	1000101
7	1011100	0111000	1110100	0110001	1001101	0001100	0000111	0111101	1011001	1110101	1110101	1011001
8	1111100	1000011	1110000	1001011	0111100	1001010	1011110	0010100	0001001	0000001	0000001	1010101
9	1000100	0111011	1101000	0100000	1011001	1010111	0100011	0101100	0100001	0111011	0111011	0010111
10	0101010	1110000	0100010	0100101	1011001	1000110	0110110	1001110	0011101	0110110	0110110	0010001
11	0001110	1000010	1001100	1001111	1010010	0110111	1000110	0011001	1010101	0111011	0111011	1111111
12	1001110	1111000	0000011	0100010	0000101	1010010	1111000	1011110	1101100	1100010	1100010	0010110

### 3.5 Initial Population

A set of possible solutions are generated randomly[8] and length 7 in string is equal to the no. of 7 month. The costs of raw materials in every month are evaluated and shown in table 4.

**Table 4:** Cost of every genes of initial 12 strings

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	TOTAL
1	170629.25	203739.75	224955.00	207782.25	254752.00	200267.00	366432.00	333085.50	316184.00	291283.00	99336.25	146732.50	2815178.50
2	189729.25	217799.00	234743.75	226543.00	246390.50	222656.00	326863.00	334620.50	362539.50	289583.00	98886.25	143393.75	2893747.50
3	173928.00	188907.75	215643.75	230505.50	232629.50	211247.25	361490.00	355250.00	317642.00	268284.00	95767.50	139225.00	2790520.25
4	178848.00	207989.75	230016.25	212444.75	231832.00	212634.75	343119.00	288193.00	317647.00	252211.75	91235.00	137993.75	2704165.00
5	198748.00	199097.75	205456.25	198964.25	258202.00	203183.75	360332.00	307554.00	342581.50	258958.00	90033.76	136103.75	2759215.01
6	188929.25	208357.75	210517.50	207731.00	223429.50	211556.00	361259.00	307410.50	317647.00	309037.75	104258.75	131183.75	2781317.75
7	178148.00	217878.50	239406.25	227232.25	218779.50	190873.50	302200.00	352663.00	315134.00	305078.00	102943.75	135805.00	2786141.75
8	197248.00	184639.75	239457.50	198964.25	258649.50	189364.75	325140.00	310460.50	295794.00	252211.75	91235.00	136103.75	2679268.75
9	173928.00	214029.75	229267.50	224164.75	231120.50	199417.00	340523.00	335570.50	342684.00	308884.00	104207.50	136505.00	2840301.50
10	192860.25	217428.50	230018.75	219801.00	231120.50	189704.50	365990.00	287411.50	317831.50	308884.00	104207.50	136955.00	2802213.00
11	173709.00	188089.75	210116.25	198913.00	234123.00	223537.00	304050.00	306925.50	316081.50	306612.75	103046.25	142995.00	2708199.00
12	172909.00	217078.50	206606.75	223526.75	220329.50	202014.75	363782.00	307091.50	340456.50	289417.75	98538.75	139092.50	2780844.25

After finding out costs of all genes of population generation by using C – programme, we will arrange these costs in ascending order so that easily we can recognise the

minimum cost string and maximum cost and in shown in table 5.

**Table 5:** Cost of 12 strings in ascending order

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	TOTAL
1	0101010	1110000	0100010	0100101	1011001	1000110	0110110	1001110	0011101	0110110	0110110	0010001	2802213.00
2	1111100	1000011	1110000	1001011	0111100	1001010	1011110	0010100	0001001	0000001	0000001	1010101	2679268.75
3	1010100	0100010	0101100	0010000	0010111	1100010	0101100	0000101	1010110	0000001	0000001	1011010	2704165.00
4	0001110	1000010	1001100	1001111	1010010	0110111	1000110	0011001	1010101	0111011	0111011	1111111	2708199.00
5	0110100	0010100	1001011	1001011	0110110	0011000	1111001	0010111	0100101	1001101	1001101	1010101	2759215.00
6	1001110	1111000	0000011	0100010	0000101	1010010	1111000	1011110	1101100	1100010	1100010	0010110	2780844.25
7	1101001	0100100	0001110	0011101	0001100	0100101	1110101	1011100	1010110	0110010	0110010	1000101	2781317.75
8	1011100	0111000	1110100	0110001	1001101	0001100	0000111	0111101	1011001	1110101	1110101	1011001	2786141.75
9	1000100	0001100	1011101	0111110	0010101	0100011	0111111	0111100	0011111	1010110	1010110	0100111	2790520.25
10	0001001	1100011	1101001	0011001	0110111	0011111	0110000	0101001	1010001	0101100	0101100	0110110	2815178.50
11	1000100	0111011	1101000	0100000	1011001	1010111	0100011	0101100	0100001	0111011	0111011	0010111	2840301.50
12	0101001	0110110	1111101	0110111	0101000	1110101	0011010	1100100	1111010	1100100	1100100	1111101	2893747.50

### 3.6 Fitness Function

Fitness functions ensure that the evolution is toward optimization by calculating the fitness value for each individual in the population. The fitness value evaluates the performance of each individual in the population [3].

Fitness function for minimizing the total inventory cost is that

$$F.F. = D * C + (D / Q) * Co + (Q / 2) * Ch$$

Where D = Demand, C = Purchasing Cost  
 Co = Ordering Cost Ch = Holding Cost

Q = economic order quantity where

$$Q = \sqrt{\frac{2.D.Co}{c_h}}$$

### 3.7 Crossover

Crossover is very important part of GA to get better characteristics from fittest solution among generations [2]. Crossover is a recombination of components due to mating [4]. A single point crossover operation is used in this problem. Here we break the two chromosomes and after breaking the genes that are right of the crossover point in the two chromosomes are interchanged thus crossover operation is done. After crossover operation two new chromosomes are generated.

Example-

Before crossover  
0001110  
0110100

After crossover  
0001100  
0110110

Now we take first 2 strings as it is, because they have minimum cost among 12 strings. We do single point crossover operation among six string ( 11,5,12,6,7,3 ) and are shown in table 6.

**Table 6:** 30 strings after crossover

S. No.	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1	0001100	1000100	1001011	1001011	1010110	0110000	1000001	0011111	1010101	0111101	0111101	1111101
2	0110110	0010010	1001100	1001111	0110010	0011111	1111110	0010001	0100101	1001011	1001011	1010111
3	0001110	1000000	1001011	1001010	1010101	0110010	1000000	0011110	1010100	0111010	0111010	1111110
4	1001110	1111010	0000100	0100111	0000010	1010111	1111110	1011001	1101101	1100011	1100011	0010111
5	0001001	1000100	1001110	1001101	1010100	0110101	1000101	0011100	1010110	0111010	0111010	1111101
6	1101110	0100010	0001100	0011111	0001010	0100111	1110110	1011001	1010101	0110011	0110011	1000111
7	0001100	1000000	1001100	1001001	1010101	0110100	1000111	0011101	1010001	0111101	0111101	1111001
8	1011110	0111010	1110100	0110111	1001010	0001111	0000110	0111001	1011101	1110011	1110011	1011111
9	0001100	1000100	1001101	1001110	1010101	0110011	1000111	0011100	1010111	0111110	0111110	1111111
10	1000110	0001010	1011100	0111111	0010010	0100111	0111110	0111001	0011101	1010011	1010011	0100111
11	0110110	0010000	1001011	1001010	0110101	0011010	1111000	0010110	0100100	1001010	1001010	1010110
12	1001100	1111100	0000011	0100011	0000110	1010000	1111001	1011111	1101101	1100101	1100101	0010101
13	0110001	0010100	1001110	1001101	0110100	0011101	1111101	0010100	0100110	1001010	1001010	1010101
14	1101100	0100100	0001011	0011011	0001110	0100000	1110001	1011111	1010101	0110101	0110101	1000101
15	0110100	0010000	1001100	1001001	0110101	0011100	1111111	0010101	0100001	1001101	1001101	1010001
16	1011100	0111100	1110011	0110011	1001110	0001000	0000001	0111111	1011101	1110101	1110101	1011101
17	0110100	0010100	1001101	1001110	0110101	0011011	1111111	0010100	0100111	1001110	1001110	1010111
18	1000100	0001100	1011011	0111011	0010110	0100000	0111001	0111111	0011101	1010101	1010101	0100101
19	1001001	1111100	0000110	0100101	0000100	1010101	1111101	1011100	1101110	1100010	1100010	0010101
20	1101110	0100000	0001011	0011010	0001101	0100010	1110000	1011110	1010100	0110010	0110010	1000110
21	1001100	1111000	0000100	0100001	0000101	1010100	1111111	1011101	1101001	1100101	1100101	1000110
22	1011110	0111000	1110011	0110010	1001101	0001010	0000000	0111110	1011100	1110010	1110010	1011110
23	1001100	1111100	0000101	0100110	0000101	1010011	1111111	1011100	1101111	1100110	1100110	0010111
24	1000110	0001000	1011011	0111010	0010101	0100010	0111000	0111110	0011100	1010010	1010010	0100110
25	1101100	0100000	0001100	0011001	0001101	0100100	1110111	1011101	1010001	0110101	0110101	1000001
26	1011001	0111100	1110110	0110101	1001100	0001101	0000101	0111100	1011110	1110010	1110010	1011101
27	1101100	0100100	0001101	0011110	0001101	0100011	1110111	1011100	1010111	0110110	0110110	1000111
28	1000001	0001100	1011110	0111101	0010100	0100101	0111101	0111100	0011110	1010010	1010010	0100101
29	1011100	0111100	1110101	0110110	1001101	0001011	0000111	0111100	1011111	1110110	1110110	1011111
30	1000100	0001000	1011100	0111001	0010101	0100100	0111111	0111101	0011001	1010101	1010101	0100001

### 3.8 Mutation

One or more gene values in a chromosome from its initial state is altered by the genetic operator known mutation [7]. By performing the mutation, a new chromosome will be generated. This is done by a random generation of two points and performing interchange between both the genes [1].

Example-

Before mutation  
0001110  
0001001

After mutation  
1001010  
0011000

We do mutation operation among last 3 initial strings (1,9,2 ).Here we perform first place mutation, second place

mutation and third place mutation. By conducting these three mutation operation, we get 9 new strings and shown in table 7.

**Table 7:** 9 strings after mutation

S. No.	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1	0001001	0100111	0101101	0011001	1110011	1011011	0110000	0101001	0010101	1101000	1101000	1110010
2	1000100	0111011	0101100	0100000	0011101	1010111	0100011	1101000	0100001	0111011	0111011	1010011
3	0101001	1110010	1111101	1110011	0101000	1110101	0011010	1100100	0111110	1100100	1100100	1111101
4	0001001	1100011	1001011	0011001	0110111	0111101	0010010	0001011	1010001	0001110	0001110	0110110
5	1000100	0111011	1001010	0000010	1011001	1110101	0100011	0001110	0000011	0111011	0111011	0110101
6	0001011	0110110	1011111	0110111	0001010	1010111	0111000	1000110	1111010	1000110	1000110	1011111
7	0011000	1110010	1111000	0011001	0110111	0011111	0100001	0111000	1010001	0101100	0101100	0100111
8	1000100	0111011	1101000	0100000	1011001	1010111	0110010	0101100	0110000	0111011	0111011	0010111
9	0111000	0100111	1111101	0110111	0101000	1110101	0001011	1100100	1101011	1100100	1100100	1111101

**3.9 Generation of Chromosome for New Iteration**

In order to produce the most efficient chromosomes, we take first 3 strings (chromosome) from population generation, 6

strings which have minimum cost from crossover generation, 3 strings with minimum cost from mutation generation and shown in table 8.

**Table 8:** costs of optimized chromosome

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	TOTAL
1	1111000	1000011	1110000	1001011	0111100	1001010	1011110	0010100	0001001	0000001	0000001	1010101	2679268.75
2	1010100	0100010	0101100	0010000	0010111	1100010	0101100	0000101	1010110	0000001	0000001	1011010	2704165.00
3	0001110	1000010	1001100	1001111	1010010	0110111	1000110	0011001	1010101	0111011	0111011	1111111	2708199.00
4	0001100	1000000	1001100	1001001	1010101	0110100	1000111	0011101	1010001	0111101	0111101	1111001	2687331.75
5	0001100	1000100	1001011	1001011	1010110	0110000	1000001	0011111	1010101	0111101	0111101	1111101	2700567.25
6	0001001	1000100	1001110	1001101	1010100	0110101	1000101	0011100	1010110	0111010	0111010	1111101	2710707.75
7	0001110	1000000	1001011	1001010	1010101	0110010	1000000	0011110	1010100	0111010	0111010	1111110	2717713.75
8	0001100	1000100	1001101	1001110	1010101	0110011	1000111	0011100	1010111	0111110	0111110	1111111	2707460.75
9	0110100	0010000	1001100	1001001	0110101	0011100	1111111	0010101	0100001	1001101	1001101	1010001	2751911.75
10	0001001	1100011	1001011	0011001	0110111	0111101	0010010	0001011	1010001	0001110	0001110	0110110	2687683.00
11	1000100	0111011	1001010	0000010	1011001	1110101	0100011	0001110	0000011	0111011	0111011	0110101	2737551.00
12	0001011	0110110	1011111	0110111	0001010	1010111	0111000	1000110	1111010	1000110	1000110	1011111	2745636.75

Now we pick the gene from first column (M1 month) which has minimum cost. We do same thing further 12 columns (all months) & adding these costs of 12 genes, then we get cost is 2515510.75. Basically we are trying to get the cost of fitness value of chromosome i.e. total inventory cost of all 7 raw materials should be minimum.

**4. OPTIMUM CHROMOSOME**

M1	0	0	0	1	0	1	1
M2	1	0	0	0	0	1	1
M3	1	0	0	1	0	1	1
M4	1	0	0	1	1	1	1
M5	1	0	1	0	1	1	0
M6	1	0	0	1	0	1	0
M7	1	0	0	0	1	0	1
M8	0	0	0	1	1	1	0
M9	0	0	0	1	0	0	1
M10	1	0	0	1	1	0	1
M11	1	0	0	1	1	0	1
M12	1	0	1	1	1	1	1

**5. CONCLUSION**

A GA approach is proposed to adding the optimal inventory level. The program is coded in C and is tested on a case study problem. In this paper, all the work that have done, is mainly concern with optimum inventory level, so genetic algorithm approach were used here for solving inventory control problem. The algorithm is found to terminate well within 100 genes by using crossover and mutation operators so we achieved our objectives which are minimizing the total inventory cost by optimal ordering quantity at reorder point using genetic algorithm approach.

**REFERENCES**

[1]. P.Radhakrishnan, V.M.Prasad, M. R. Gopalan, "Inventory Optimization in Supply Chain Management using Genetic Algorithm" International Journal of Computer Science and Network Security, No.1, Vol.9, January 2009, 33-40.  
 [2]. Tapan p. bagchi, "Multiobjective scheduling by Genetic algorithm" Kluwer academic publishers.

- [3]. S. Godwin barnabs, S. Thandeeswaran, M. Ganeshkumar, K. Kani raja, B. Selvakumar, "Spare parts inventory optimization for auto mobile sector" European journal of business and management Vol. 4, No. 17,2012, 168-176.
- [4]. S Shakeelahamed, G. RangaJanardhana,E.L.Nagesh, "GA Based Inventory Control for Manufacturing Unit" published in American Journal of Scientific Research,2011, 56-69.
- [5]. S.L.Adeyemi, A.O. Salami, "Inventory Management: A Tool Optimizing Resources in a Manufacturing Industry A Case study of Coca-Cola Bottling Company, Ilorin Plant,Jsocsci, 23(2): (2010), 135-142.
- [6]. S. Narmadha, Dr. V.Selladurai, G.Sathish, "Multi-product Inventory Optimization using Uniform Crossover Genetic Algorithm"International journal of Computer Science and Information Security, Vol. 7, No.1, 2010, 170-177.
- [7]. S. Shakeel Ahamed, Dr.G.Rangajanardhana, E. L. Nagesh, "Application of Supply Chain Tools In Power Plant- A Case of Rayalaseema Thermal Power Plant", Industrial Engineering LettersVol 2, No.2, 2012, 1-10.
- [8]. C. Darwin, "On the Origin of Species by Means of Natural Selection" Published: November 24, 1859 London.
- [9]. I.K. Moon, B. C. Cha, S. K. Kim, "Offsetting inventory cycles using mixed integer programming and genetic algorithm" , International journal of industrial engineering, 15(3), 245-256, 2008
- [10]. Melanie Michell, "An Introduction to Genetic Algorithm", Fifth printing, 1999, first MIT Press paperback edition, 1998.
- [11]. Prem kumar gupta, D.S. Hira, "Operations research", S. chand & company ltd., first edition 1976, reprint 2010.