[Aman* *et al.*, 5(9): September, 2016]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

ANALYSIS & COMPARISION OF INVENTORY COST BY GENETIC ALGORITHM, INVERTED METHOD AND FITNESS FUNCTION

Mohd Aman*, Prabhat Kumar Sinha, Anshuka Srivastava
M.Tech. Scholar, Dept. of Mechanical Engineering, SHIATS, Allahabad, U.P., India
Assistant Professor, Dept. of Mechanical Engineering, SHIATS Allahabad, U.P., India
Professor, Dept. of Mechanical Engineering, SHIATS Allahabad, U.P., India

DOI: 10.5281/zenodo.61483

ABSTRACT

Inventory is the list of movable items which are required for manufacturing the product or assembly the system to maintain the plant. Inventory is directly related with the manufacturing cost. Inventory control refers to making the desired item of required quality and in required quantity. In this paper we are going to optimize inventory cost in Bharat Pumps & Compressor Limited Naini, Allahabad by visiting there and selecting seven inventory which is highly used. Here I have taken seven raw material and try to find their exact reorder point by considering some item holding and some item ordering at the same time. Here we proposed Genetic Algorithm approach to find the optimize cost of the inventory used. In Genetic Algorithm we used crossover, mutation and inverted method to get the cost of item and compare them with the fitness function value. The nearest value of the fitness function will be best value in our analysis. The whole calculation would be performed on Microsoft excel.

KEYWORDS: Inventory management, Economic order quantity, Genetic algorithm, Inverted chromosome.

INTRODUCTION

No business or plant can successfully operate without inventory management [1]. Inventory in company consist of thousands of different items in stock, to control on these item have the serious problem to the management [2]. Genetic Algorithm is used to optimize the value of inventory cost by considering proper reorder point. Genetic algorithm have different methods through which it helps in optimizing the inventory cost. It provide a way through which items can be order and hold for proper functioning without delay. It consider an economic order quantity for its analysis.

we are used the Genetic Algorithm[3] to find the optimized ordering quantity at the proper reorder point. GA is an optimization method on the basis of the genetic concept. It is a method for solving the multivariable optimization issues which are regarded to be complicated by conventional optimization methods. As seen in nature, Genetic Algorithms are similar adaptive search techniques which simulate an evolutionary process on the basis of the ideas of selection of the fittest, crossing and mutation.

LITERATURE REVIEW

S R Singh [4] have made optimization inventory with the help of supply chain management. In this paper, we propose an efficient approach that effectively utilizes the Genetic Algorithm for optimal inventory control. This paper reports a method based on genetic algorithm to optimize inventory in supply chain management. G. Ramya and M. Chandrasekaran [5] have try to minimize total holding cost with no tardiness in sequencing for scheduling analysis. They defined scheduling which is an allocation of tasks to the time intervals on the machines. The aim is to find a schedule that minimizes the overall completion time, which is called the makespan. In the job shop scheduling problem P different jobs have to be processed on Q different machines. Kuan Yew Wong [6] has discussed flexible job shop manufacturing method. Scheduling is significantly investigated in manufacturing systems. Scheduling has been applied to meet customer demand, which plays an important role in customer satisfaction. Additionally, providing a better schedule influences a system's performance. P.Radhakrishnan [7] has made design optimization method for supply chain management using lead time. Inventory management is considered to be a very important area in Supply chain management. Efficient and effective management of inventory throughout the supply chain significantly improves the ultimate service provided to the customer.

A CASE STUDY

Data of Raw Materials

In BPCL Naini plant; many raw materials comes in working process. Here we have selected some of raw materials in the proposed work. The raw materials that are used are given below

1. Al cladding sheet
2. Fastener
3. Electrodes
4. Gaskets
5. Valve
6. Pressure Gauge
7. Bearings.
8. Quantity of all 7 raw materials which is used in all 12 months (entire year) are given and the costs of these inventories as purchasing cost, holding cost, ordering cost are also mentioned in this problem.

Demand Matrix

The quantity of raw materials which are used in different months in a year is shown in table no.1

Table no. 1: Demand of Raw material

Raw material/ month	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1st	20	20	20	20	30	30	40	25	70	60	20	20
2nd	500	500	500	500	600	600	1000	1200	1200	1000	200	200
3rd	100	200	200	150	250	250	450	400	450	350	100	100
4th	50	25	25	25	25	25	75	75	75	50	25	25
5th	5	3	5	5	4	1	12	10	10	15	5	5
6th	2	3	3	4	5	2	2	2	1	2	3	3
7th	40	40	50	50	40	30	40	30	20	20	10	30

Various Cost of Raw Materials

The purchasing, holding, ordering costs of various materials are shown in table no 2.

Table no. 2: Cost components

Raw material	Purchasing Cost	Holding Cost	Ordering Cost
Al cladding sheet	200	10	1
Fastener	3.65	0.18	2
Electrodes	21.6	1.08	3
Gaskets	173	9	1
Valve	365	18.25	4
Pressure Gauge	3230	161.5	1
Bearings	405	20.25	1.5

Population Generation and Chromosome In our problems we have taken 7 raw materials from the BPCL plant which has to deliver in all 12 months of the year. As I know that each chromosome incorporates genes. Here the population has to represents every month with different genes which will make by adding all 12 genes; as chromosome.

Here we consider that each gene consists of 7 numerical value because we have 7 raw materials those are already selected from plant.

Table no 3. Randomly generated chromosome.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
S1	0001001	1100011	1101001	0011001	0110111	0011111	0110000	0101001	1010001	0101100	0101100	0110110
S2	0101001	0110110	1111101	0110111	0101000	1110101	0011010	1100100	1111010	1100100	1100100	1111101
S3	1000100	0001100	1011101	0111110	0010101	0100011	0111111	0111100	0011111	1010110	1010110	0100111
S4	1010100	0100010	0101100	0010000	0010111	1100010	0101100	0000101	1010110	0000001	0000001	1011010
S5	0110100	0010100	1001011	1001011	0110110	0011000	1111001	0010111	0100101	1001101	1001101	1010101
S6	1101001	0100100	0001110	0011101	0001100	0100101	1110101	1011100	1010110	0110010	0110010	1000101
S7	1011100	0111000	1110100	0110001	1001101	0001100	0000111	0111101	1011001	1110101	1110101	1011001
S8	1111100	1000011	1110000	1001011	0111100	1001010	1011110	0010100	0001001	0000001	0000001	1010101
S9	1000100	0111011	1101000	0100000	1011001	1010111	0100011	0101100	0100001	0111011	0111011	0010111
S10	0101010	1110000	0100010	0100101	1011001	1000110	0110110	1001110	0011101	0110110	0110110	0010001
S11	0001110	1000010	1001100	1001111	1010010	0110111	1000110	0011001	1010101	0111011	0111011	1111111
S12	1001110	1111000	0000011	0100010	0000101	1010010	1111000	1011110	1101100	1100010	1100010	0010110

Initial Population

A set of possible solutions are generated randomly [8] and length 12 in string is equal to the no. of 12 month. The costs of raw materials in every month are evaluated and shown in table no. 4

Table no.4: Cost of every genes of initial 12 string

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total Annual
S1	42042	41418	48147	49962	54281	38124	67162	56963	58740	55711	28220	36636	577406
S2	42952	42690	48460	50359	55211	39467	64421	57758	60539	55571	28240	36175	581840
S3	42941	41597	47550	51097	53992	38950	65320	58151	58467	54102	27587	35881	575633
S4	43133	42348	49193	51100	53189	39243	65527	55236	58812	54130	27940	35963	575813
S5	44223	42181	46755	48852	55031	39022	65452	55683	60548	52976	27489	36011	574222
S6	42772	42707	47802	49891	54062	39257	65881	55742	58812	56676	28202	35819	577621
S7	42733	42934	49597	51072	53042	38528	63236	57588	58140	55868	28245	35882	576863
S8	43643	40508	49668	48852	55634	37951	63890	56567	57906	54130	27940	36011	572699
S9	42941	41782	49084	51722	53579	38054	65227	57383	60690	55901	27814	35709	579885
S10	43381	42954	48983	50713	53579	38137	66670	54653	58628	56462	28131	36262	578550
S11	42400	41258	48103	48781	53726	39416	63626	55547	58598	55901	27814	35693	570862
S12	42220	42754	47135	51080	53512	38631	66202	55421	59693	55464	27830	36272	576211

For considering minimum cost and maximum cost from the above table we have arrange the cost of every population in ascending order by using Microsoft excel which is shown in table no. 5

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total Annual Cost
S11	42400	41258	48103	48781	53726	39416	63626	55547	58598	55901	27814	35693	570862
S8	43643	40508	49668	48852	55634	37951	63890	56567	57906	54130	27940	36011	572699
S5	44223	42181	46755	48852	55031	39022	65452	55683	60548	52976	27489	36011	574222
S3	42941	41597	47550	51097	53992	38950	65320	58151	58467	54102	27587	35881	575633
S4	43133	42348	49193	51100	53189	39243	65527	55236	58812	54130	27940	35963	575813
S12	42220	42754	47135	51080	53512	38631	66202	55421	59693	55464	27830	36272	576211
S7	42733	42934	49597	51072	53042	38528	63236	57588	58140	55868	28245	35882	576863
S1	42042	41418	48147	49962	54281	38124	67162	56963	58740	55711	28220	36636	577406
S6	42772	42707	47802	49891	54062	39257	65881	55742	58812	56676	28202	35819	577621
S10	43381	42954	48983	50713	53579	38137	66670	54653	58628	56462	28131	36262	578550
S9	42941	41782	49084	51722	53579	38054	65227	57383	60690	55901	27814	35709	579885
S2	42952	42690	48460	50359	55211	39467	64421	57758	60539	55571	28240	36175	581840

Table no. 5: Cost of every genes in ascending order.

THOMSON REUTERS ENDNOTE
[Aman* *et al.*, 5(9): September, 2016] ISSN: 2277-9655
IC™ Value: 3.00 Impact Factor: 4.116

Evaluation of fitness function:

Fitness function is specific kind of objective function that enumerates the optimality of a solution in a genetic algorithm. A fitness function is a particular type of objective function that is used to summarise as how to close a given solution is to achieving the set goals. Fitness functions ensure that the evolution is towards optimization by calculating the fitness value for each individual in the population. Fitness function always have given best value and no value can be best than that. The fitness value evaluates the performance of each individual in the population [9].

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,
C_o = Ordering Cost,
C_h = Holding Cost,
Q = economic order quantity
where Q = $\sqrt{\frac{2.D.C_0}{c_h}}$

total cost of all seven material will come by adding all the value as given below.

1. For 1st raw material (Al cladding sheet)
   D = 375, C = 200, C_h = 10, C_o = 1
   Then Q = $\sqrt{\frac{2.D.C_0}{c_h}}$ = √ (2x 375x1/10) = 8.66
   Now Total inventory cost is
   = $D * C \ (D / Q) * Co + (Q / 2) * Ch$
   = 375x200 + (375/8.66)x1 + (8.66 / 2)x10
   = 75086
2. For 2nd raw material (Fasteners)
   D = 8000, C = 3.65, C_h = 0.18, C_o = 2
   Then Q= √(2x8000x2/0.18)= 421.63
   Now Total inventory cost is
   =8000x3.65+ (8000/421.63) x2+(421.63/2) x 0.18 = 29275
3. For 3rd raw material (Electrodes)
   D = 3000, C = 21.60, C_h = 1.08, C_o = 3
   Then Q = √ (2x3000x3/1.08) = 129
   Now Total inventory cost is
   = 3000x21.60+(3000/129)x3+(129/ 2) * 1.08 = 64939
4. For 4th raw material (Gaskets)
   D = 500, C = 173, C_h = 9, C_o = 1
   Then Q = √ (2x 500x1 / 9) = 10.5
   Now Total inventory cost is
   = 500x173 + (500 / 10.5)x1 + (10.5/2)x9
   = 86594
5. For 5th raw material (valve)
   D = 80, C = 365, C_h = 18.25, C_o = 4
   Then Q = √ (2 x 80 x 4 / 18.25) = 5.92
   Now Total inventory cost is
   = 80x365+(80/5.92) x 4 + (5.92 / 2) x18.25
   = 29,308
6. For 6th raw material (Pressure Gauge)
   D = 32, C = 3230, C_h = 161.50, C_o = 1
   Then Q =√ (2x 32x 1 / 161.50) = 0.62
   Now Total inventory cost is =32x3230+(32/0.62)x1+ (0.62 /2) x161.5
   = 103,461

ISSN: 2277-9655
Impact Factor: 4.116

7. For 7th raw material (Bearings)

$D = 400$, $C = 405$, $C_h = 20.25$, $C_o = 1.5$

Then $Q = \sqrt{(2x\,400x1.5 / 20.25)} = 7.69$

Now Total inventory cost is

= 400x405+(400/7.69)x1.5+(7.69/2)x20.2

= 162,155 **Total cost= 550,818**

Crossover :

Crossover is very important part of GA to get better characteristics from fittest solution among generations [10]. Crossover is a recombination of components due to mating [12].

A single point crossover operation is used in this problem. Here we break the two chromosomes and after breaking the genes which are at right of the crossover point in the two chromosomes are interchanged thus crossover operation is done. After each crossover operation two new chromosomes are generated.

Example-

Before crossover	after crossover operation
0001¦110	0001¦100
0110¦100	0110¦110

Now we select first 3 strings (11, 8 and 5) for the final evaluation as it is, because they have minimum cost among all 12 strings.

We will perform single point crossover operation among four string (3, 4, 12, 7,).

Table No.6. 12 string of crossover

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
S1	1000100	0001010	1011100	0111000	0010111	0100010	0111100	0111101	0011110	1010001	1010001	0100010
S2	1000100	0100100	0101101	0010110	0010101	1100011	0101111	0000100	1010111	0000110	0000110	1011111
S3	1000110	0001000	1011011	0111010	0010101	0100010	0111000	0111110	0011100	1010010	1010010	0100110
S4	1001100	1111100	0000101	0100110	0000101	1010011	1011100	1011001	1101111	1100110	1100110	0010111
S5	1000100	0001000	1011100	0110011	1001100	0001010	0111101	0011100	0011001	1010101	1010101	0100001
S6	1011100	0111100	1110101	1001001	0111101	1001100	0111100	1011001	1011111	1110110	1110110	1011111
S7	1010110	0100000	0101011	0010010	0010101	1100010	0000110	0010110	1010100	0000010	0000010	1011110
S8	1001100	1111010	0000100	0100000	0000111	1010010	1011101	1011111	1101110	1100001	1100001	0010010
S9	1010100	0100000	0101100	0010001	0010101	1100100	0000101	0010100	1010001	0000101	0000101	1011001
S10	1011100	0111010	1110100	0010001	1001111	0001010	0111101	1011111	1011110	1110001	1110001	1011010
S11	1001100	1111000	0000100	0100001	0000101	1010100	1011101	1011100	1101001	1100101	1100101	0010001
S12	1011110	0111000	1110011	0110010	1001101	0001010	0111110	1011110	1011100	1110010	1110010	1011110

Mutation

One or more gene values in a chromosome from its initial state is altered by the genetic operator known mutation [11]. By performing the mutation operation, a new chromosome will be generated. This is done by a random generation of two points and performing interchange between both the genes [13]. The way in which the value are interchanged in the mutation should uniformly follow in the population.

Example-

Before mutation	after mutation operation
0001110	1001010
0001001	0011000

We do mutation operation among last 5 initial strings (1, 6, 10, 9, 2). Here we perform first place mutation, second place mutation. By conducting these two mutation operation, we get 10 new strings which is shown in table no. 7

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
S1	0001001	0100111	0101101	0011001	1110011	1011011	0110000	0101001	0010101	1101000	1101000	1110010
S2	0001001	1100011	1001011	0011001	0110111	0111101	0010010	0001011	1010001	0001110	0001110	0110110
S3	0101101	1100000	1001010	1011001	1001000	1100001	1110101	1011100	1010110	0110010	0110010	1000101
S4	1001011	0000110	0101100	0011101	0001100	0000111	1010111	1011100	1110100	0110010	0110010	1000101
S5	0101010	0110100	0100010	1100001	0011101	1000110	1110010	1001110	1011001	1110010	1110010	0010001
S6	0101010	1010010	0100010	0000111	1011001	1100100	0110110	1101100	0011101	0110110	0110110	0010001
S7	1000100	0111011	0101100	0100000	0011101	1010111	0100011	1101000	0100001	0111011	0111011	1010011
S8	1000100	0111011	1001010	0000010	1011001	1110101	0100011	0001110	0000011	0111011	0111011	0110101
S9	0101001	1110010	1111101	1110011	0101000	1110101	0011010	1100100	0111110	1100100	1100100	1111101
S10	0001011	0110110	1011111	0110111	0001010	1010111	0111000	1000110	1111010	1000110	1000110	1011111

Table No. 7: 10 strings after mutation

Generation of optimized chromosome :

In order to produce the most efficient chromosomes, we are taking first 3 strings (chromosome) from population generation which have minimum cost, 5 strings which have minimum cost from crossover generation and 4 strings which have minimum cost from mutation generation and shown in table no. 8 and table no. 9 shows the binary code of the minimum cost obtained.

Now we pick the genes from all columns (from M1 month to M12 month) whichever have minimum cost for example 41541 is a minimum cost in M1 month (first column). Same as for M2 month (second column), cost is 40508 and so on. We do same thing further 12 columns (all months). Minimum cost of all 12 months is indicating below table no. 8 and it is indicating in yellow color.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total Annual
S11	42400	41258	48103	48781	53726	39416	63626	55547	58598	55901	27814	35693	570862
S8	43643	40508	49668	48852	55634	37951	63890	56567	57906	54130	27940	36011	572699
S5	44223	42181	46755	48852	55031	39022	65452	55683	60548	52976	27489	36011	574222
S2 c	42941	42707	48256	50387	53992	38680	64456	55799	58437	53970	27575	35329	572527
S4 c	42541	42711	47546	51009	53512	38068	64211	55322	59157	55250	27759	35709	572793
S7 c	42812	42750	47845	50458	53992	39243	63986	56246	58973	54184	27646	35892	574024
S8 c	42541	42352	48483	51722	52709	38631	63461	54858	59532	55410	28124	36343	574165
S5 c	42941	41640	48487	50430	53792	38221	65641	55967	58770	54048	27881	36434	574250
S2m	42042	41418	46755	49962	54281	39537	65021	54458	58740	53570	27375	36636	569795
S10m	41721	42690	47068	50359	53316	38054	66562	55253	60539	53430	27395	35329	571714
S8 m	42941	41782	47693	50170	53579	39467	65227	54878	58346	55901	27814	36555	574351
S4 m	41541	41396	49193	49891	54062	37844	63740	55742	61157	56676	28202	35819	575260

Table no.8: cost of optimized chromosome.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
S11	0001110	1000010	1001100	1001111	1010010	0110111	1000110	0011001	1010101	0111011	0111011	1111111
S8	1111100	1000011	1110000	1001011	0111100	1001010	1011110	0010100	0001001	0000001	0000001	1010101
S5	0110100	0010100	1001011	1001011	0110110	0011000	1111001	0010111	0100101	1001101	1001101	1010101
S2 c	1000100	0100100	0101101	0010110	0010101	1100011	0101111	0000100	1010111	0000110	0000110	1011111
S4 c	1001100	1111100	0000101	0100110	0000101	1010011	1011100	1011001	1101111	1100110	1100110	0010111
S7 c	1010110	0100000	0101011	0010010	0010101	1100010	0000110	0010110	1010100	0000010	0000010	1011110

S8 c	1001100	1111010	0000100	0100000	0000111	1010010	1011101	1011111	1101110	1100001	1100001	0010010
S5 c	1000100	0001000	1011100	0110011	1001100	0001010	0111101	0011100	0011001	1010101	1010101	0100001
S2m	0001001	1100011	1001011	0011001	0110111	0111101	0010010	0001011	1010001	0001110	0001110	0110110
S10m	0001011	0110110	1011111	0110111	0001010	1010111	0111000	1000110	1111010	1000110	1000110	1011111
S8 m	1000100	0111011	1001010	0000010	1011001	1110101	0100011	0001110	0000011	0111011	0111011	0110101
S4 m	1001011	0000110	0101100	0011101	0001100	0000111	1010111	1011100	1110100	0110010	0110010	1000101

Table no. 9: cost of optimized chromosome in binary code.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
S11 in	1110001	0111101	0110011	0110000	0101101	1001000	0111001	1100110	0101010	1000100	1000100	0000000
S8 in	0000011	0111100	0001111	0110100	1000011	0110101	0100001	1101011	1110110	1111110	1111110	0101010
S5 in	1001011	1101011	0110100	0110100	1001001	1100111	0000110	1101000	1011010	0110010	0110010	0101010
S2 c in	0111011	1011011	1010010	1101001	1101010	0011100	1010000	1111011	0101000	1111001	1111001	0100000
S4 c in	0110011	0000011	1111010	1011001	1111010	0101100	0100011	0100110	0010000	0011001	0011001	1101000
S7 c in	0101001	1011111	1010100	1101101	1101010	0011101	1111001	1101001	0101011	1111101	1111101	0100001
S8 c in	0110011	0000101	1111011	1011111	1111000	0101101	0100010	0100000	0010001	0011110	0011110	1101101
S5 c in	0111011	1110111	0100011	1001100	0110011	1110101	1000010	1100011	1100110	0101010	0101010	1011110
S2 m in	1110110	0011100	0110100	1100110	1001000	1000010	1101101	1110100	0101110	1110001	1110001	1001001
S10 m in	1110100	1001001	0100000	1001000	1110101	0101000	1000111	0111001	0000101	0111001	0111001	0100000
S8 m in	0111011	1000100	0110101	1111101	0100110	0001010	1011100	1110001	1111100	1000100	1000100	1001010
S4 m in	0110100	1111001	1010011	1100010	1110011	1111000	0101000	0100011	0001011	1001101	1001101	0111010

Table no 10. Optimized inverted chromosome

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total Annual Cost
S11 in	43364	42141	48429	52010	54404	38272	65812	57437	60305	53751	27876	36632	580432
S8 in	42121	42891	46864	51939	52496	39737	65548	56417	60996	55522	27751	36315	578595
S5 in	41541	41218	49777	51939	53099	38666	63986	57300	58355	56676	28202	36315	577072
S2 c in	42823	40692	48277	50404	54138	39008	64982	57185	60465	55682	28116	36996	578767
S4 c in	43223	40688	48987	49782	54618	39620	65227	57662	59745	54402	27932	36616	578501
S7 c in	42952	40650	48687	50333	54138	38445	65452	56738	59930	55468	28045	36434	577270
S8 c in	43223	41047	48049	49069	55421	39057	65977	58125	59370	54242	27567	35983	577129
S5 c in	42823	41760	48045	50361	54338	39467	63797	57017	60132	55604	27810	35892	577044
S2 m in	43722	41981	49777	50829	53849	38151	64417	58526	60162	56082	28316	35690	581499
S10 m in	44043	40710	49464	50432	54814	39634	62876	57731	58364	56222	28296	36996	579580
S8 m in	42823	41617	48840	50621	54551	38221	64211	58106	60557	53751	27876	35771	576943
S4 m in	44223	42004	47339	50900	54068	39844	65698	57242	57746	52974	27489	36507	576034

Table no 11. Cost of Optimized inverted chromosome

GENERATION OF INVERTED CHROMOSOME

In order to produce the inverted chromosome[14] all the positive status '1' of the best chromosome are changed to negative ordering status '0' and the negative status '0' is changed to positive status '1'.By this way we will get the new matrix of inverted chromosome of (12*12) which is shown in table 10 and their respective cost is shown in table 11.

Optimized chromosome

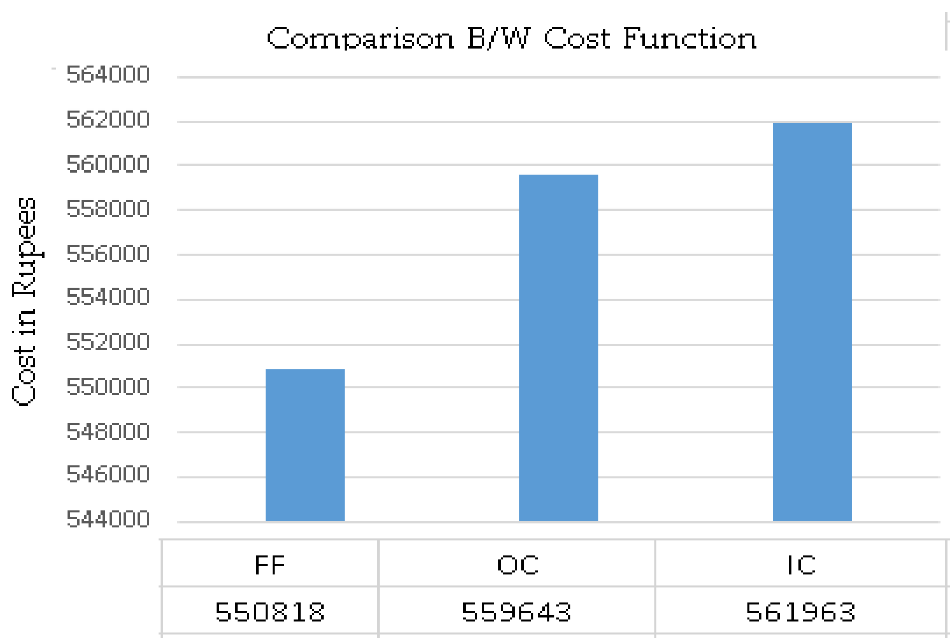
Basically we are trying to get the cost of optimized chromosome which is very close by the cost of fitness function (minimum total inventory cost) means total inventory cost of all 7 raw materials must be near by 550,818. Here we get the total optimized minimum inventory cost by genetic algorithm is 559,643 and minimum cost by optimized inverted chromosome is 561,963 of which it is clear that optimized chromosome is nearer to the fitness function (minimum total inventory cost 550,818) .

0ptimized chromosome

	part1	p2	p3	p4	p5	p6	p7
Jan	1	0	0	1	0	1	1
Feb	1	0	0	0	0	1	1
Mar	1	0	0	1	0	1	1
Apr	1	0	0	1	1	1	1
May	0	0	0	0	1	1	1
Jun	0	0	0	0	1	1	1
Jul	1	0	1	1	1	0	1
Aug	0	0	0	1	0	1	1
Sep	0	0	0	1	0	0	1
Oct	1	0	0	1	1	0	1
Nov	0	0	0	1	1	1	0
Dec	1	0	1	1	1	1	1

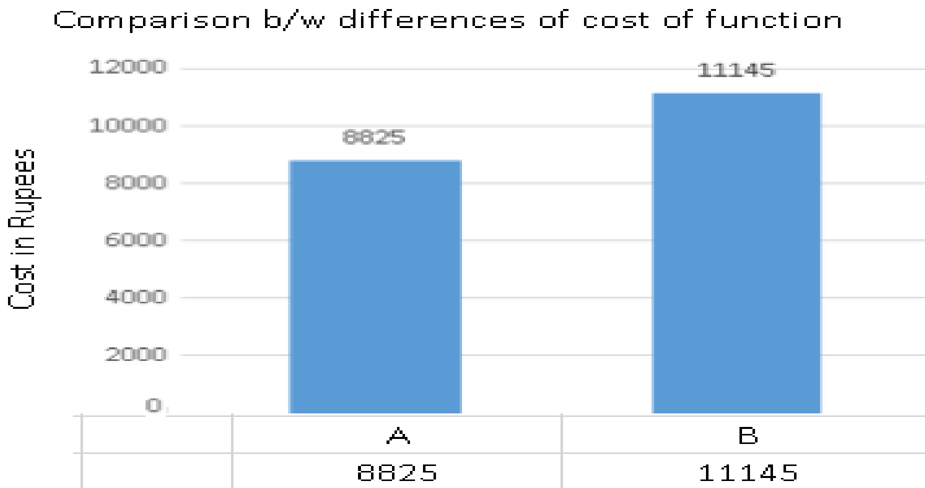
RESULT

By solving the problem by GA we got an optimized chromosome cost(OC) is 559,643 and Fitness function cost(FF) is 550,818 which we got by solving by EOQ formula and optimized inverted chromosome cost (IC) is 561,963. The optimized chromosome is the best way for ordering and holding the item which we obtained by calculation and it is very near the fitness function cost. The value of inverted chromosome is for away from fitness function cost. So we have get our best optimized chromosome by Genetic algorithm approach. Now we do comparison among these costs by bar chart in given below figure.



comparison among costs of function FF, OC and IC.

ISSN: 2277-9655
Impact Factor: 4.116



comparison between differences of cost of function A, B

CONCLUSION

The main aim of this paper to control inventory cost of the BPCL Naini plant . Here we proposed genetic algotithm approach to minimize the cost of selected seven items. In this approach we used crossover, mutation and inverted method and compare with the fitness function according to economic order quantity. This approach helps in ordering and holding of items at proper time by considering their reorder point. Whole calculation is successfully performed on the Microsoft excel.

REFERENCES

[1] Mr.S.Godwin barnabas, Mr.S.Thandeeswaran, M.Ganeshkumar, K.Kani raja, B.Selvakumar, "SPARE PARTS INVENTORY OPTIMIZATION FOR AUTO MOBILE SECTOR" European Journal of Business and Management ISSN 2222-1905 (Paper) ISSN 2222-2839 (Online) Vol 4, No.17, 2012

[2] S. Godwin Barnabas, I. Ambrose Edward, and S.Thandeeswaran, "Spare Parts Inventory Model for Auto Mobile Sector Using Genetic Algorithm" 3rd International Conference on Trends in Mechanical and Industrial Engineering (ICTMIE'2013) January 8-9, 2013 Kuala Lumpur (Malaysia)

[3] P.Radhakrishnan ** Prof.N.Jeyanth, "Application of Genetic Algorithm to Supply Chain Inventory Optimization" JOURNAL OF CONTEMPORARY RESEARCH IN MANAGEMENT January - March, 2010

[4] S.R. Singh, Tarun Kumar, "Inventory Optimization in Efficient Supply Chain Management" International Journal of Computer Applications in Engineering Sciences [VOL I, ISSUE IV, DECEMBER 2011] [ISSN: 2231-4946]

[5] G. Ramya and M. Chandrasekaran, "An Efficient Heuristics for Minimizing Total Holding Cost with no Tardy Jobs in Job Shop Scheduling" International Journal of Computer and Communication Engineering, Vol. 2, No. 2, March 2013

[6] Sayedmohammadreza Vaghefinezhad1, Kuan Yew Wong, "A Genetic Algorithm Approach for Solving a Flexible Job Shop Scheduling Problem"

[7] P.Radhakrish thi nan and Dr. M. R. Gopalan N.Jeyan , "Design of Genetic Algorithm Based Supply Chain Inventory Optimization with Lead Time" IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.4, April 2010

[8] Imran Ali Chaudhry, "A Genetic Algorithm Approach for Process Planning and Scheduling in Job Shop Environment" Proceedings of the World Congress on Engineering 2012 Vol III WCE 2012, July 4 - 6, 2012, London, U.K

[9] P. Radhakrishnan, V.M. Prasad and M.R. Gopalan, "Optimizing Inventory Using Genetic Algorithm for Efficient Supply Chain Management" Journal of Computer Science 5 (3): 233-241, 2009 ISSN 1549-3636

[10] 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.

[11] 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.

[12] 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.

[13] 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" J soc sci, 23(2):(2010), 135-142.

[14] P. Radhakrishnan, Dr. V. M. Prasad, Dr. 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.