Industrial Engineering Letters
ISSN 2224-6096 (print) ISSN 2225-0581 (online)
Vol 2, No.2, 2012

Application of Supply Chain Tools In Power Plant- A Case of Rayalaseema Thermal Power Plant

S. Shakeel Ahamed [1*] Dr.G.Rangajanardhana [2] E. L. Nagesh [3]

1. Department of Mechanical Engineering, Madina Engineering College, Kadapa, Andhra Pradesh, India.
2. Department of Mechanical Engineering, University College of engineering, JNTUK Vijayanagaram Campus ,Vijayanagaram, Andhra Pradesh, India
3. Nobel college of Engineering and Technology for women, Hyderabad, India.

* shakeelkdp@gmail.com

Abstract.

Inventories are considered to be one of the important weapons of supply chain to improve the efficiency of any manufacturing unit. Continuous availability of inventories are the prime requirement for uninterrupted working .To effectively manage inventory levels, it is essential to consider the appropriate reorder points as well as the optimized ordering quantity. The proposed system uses the Genetic algorithm to find the optimized ordering quantity at proper reorder point , by considering the power plant live data as a practical case study. The proposed approach is implemented in Matlab platform version 7.10

Key words. Inventories, supply chain, Genetic algorithm, Reorder point, ordering quantity, manufacturing unit, power plant, raw materials.

1.Proposed work

RTPP(Rayalaseema Thermal power plant) utilizes many raw materials in their working process. From these author has selected some of the raw materials in the proposed work. In this research work an attempt has been made to improve the performance of present methodology by applying the 'supply chain tools' using genetic algorithm. The raw materials that are use in this case are as follows.

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

2.Generation Of Associated Solution Demand Matrix

The associated solution demand matrix: $D2 = \{D2_{ij} \,|\, D2_{ij} < N_{\max} \;; i = 1, \cdots, |R|; 1 < j \le |M|\}$ which contains the expected solution demands for every raw material for the period of M is generated using the predicted demand rate $D1$ where $N_{\max} = Max(D1) + 0.20 \times Max(D1)$. The randomly generated solution demand rate for every raw material is less than $N_{\max}$ and every row of the associated solution demand matrix gives the expected ordering quantity of each raw material in R respectively.(i.e.) the first row of $D2$ is the expected solution demand for the raw material $R1$ for the period of 'M' months , the second row is the expected solution demand for the raw material $R2$ and so on.

Specimen calculations: N_{max} =350+0.20×350=420

Hence the generated demand values of Raw material Aluminum cladding sheet in every month(corresponding row) will be less than the value of N_{max}.

3. Population generation and chromosome representation

The genetic algorithm which incorporates a fitness function to numerically evaluate the quality of each chromosome within the population searches for the optimized ordering quantity and reorder point. The population consists of a group of individuals called chromosomes each of which represents a complete solution to the defined problem. Initially N_C No of chromosomes are randomly generated. Each gene represents a randomly generated number between 0 and $2^{|R|}-1$ which is subsequently encoded by employing a decimal to binary encoder where $|R|$ is the number of raw materials. The raw materials used in this case is 7, hence the population is generated as 2 -1=127.

12 chromosomes (12*12) gene values in binary form are generated as shown in Table 3

3.1 Fitness evaluation of chromosome

The genetic algorithm searches for the chromosome with highest fitness, where the fitness function is used to assess the quality of a given chromosome within the population. To find the optimized ordering quantity and reorder point the proposed system uses the $D1$ the demand matrix and $D2$, associated solution demand matrix for finding the fitness of the chromosome. The fitness of the chromosome is calculated as follows.

$$FC_k = \frac{|R|}{\sum FG_i}$$; where, $1 < i < M$, where R is the length of chromosome and

$$FG_i = \begin{cases} FGh_i & If \;\; P = Hc \\ FGs_i & If \;\; P = Sc \end{cases} \quad \text{and} \quad P = \begin{cases} Hc & If \;\; D2_{ij} > D1_{ij} \\ Sc & If \;\; D2_{ij} < D1_{ij} \\ 0 & If \;\; D2_{ij} = D1_{ij} \end{cases}$$

$$FGh_i = X_{ij}((30*(V_j + V_{j-1})*P) + (Pur_j * D2_{ji}) + Ord_j) \text{ and}$$

$$FGs_i = X_{ij}((30*(V_j)*P) + (Pur_j * D2_{ji}) + Ord_j) \text{ where } V_j = D1_{ij} - D2_{ij}$$

Depends on the deviation value of the $D2_{ij}$ with $D1_{ij}$, the fitness of the gene value changes. If $D2_{ij}$ the associated solution ordering quantity of the i[th] raw material for the j[th] month is greater than the $D1_{ij}$ then the fitness of the corresponding gene is calculated with the holding cost. If $D2_{ij}$ is less than the $D1_{ij}$, the gene value is calculated with the shortage cost. The fitness function is carried out for the every chromosome and the best two parent chromosomes are selected according to their high value in fitness.

3.2 Cross over

Crossover is also known as recombination of component materials due to mating. The outcome of crossover heavily depends on the chromosomes selected from the population. Crossover is a binary genetic operator acting on two parents. Different crossover operators have been developed for various purposes. In this work, the single point crossover operator selects a crossover point within a chromosome at random by using the cross over rate.

Subsequently, genes of the two parent chromosomes in between the point are interchanged to produce two new off springs. The crossover points c_1 is determined as follows.

$$C1 = | M | * R_a$$

R_a denotes the cross over rate and M denotes months Here a cross over rate of 0.4 is used to generate new offspring's. The 12 chromosomes that are produced by population generation(Table 3) are utilized for cross over and 6 new offspring's are generated as shown in the (Table3.2)

3.3 Mutation

One or more gene values in a chromosome from its initial state is altered by the genetic operator known as mutation, which may lead to entirely new gene values being added to the gene pool. The genetic algorithm may be capable of arriving at a better solution than the solution previously achieved, possible by employing these new gene values. Owing to the fact that mutation helps to prevent the population from stagnating at any local optima, it is considered as an important part of the genetic search. Mutation operator occurs in accordance with a user-definable mutation probability during the evolution.

The Mutation operation can be effectively performed using the index value "I" and the Mutation value "MV". The new offspring's produced from cross over operation are Mutated by randomly generated Index value and mutation value. The first offspring1 is Mutated with I=5, the randomly generated value and the value of MV is also generated randomly within 2-1=127 say MV=87. This value of 87 is converted in to binary form 1010111.As the Index value is taken as 5 the corresponding gene at the 5^{th} cell is replaced by the binary form of MV as shown in table 3.3. This process is followed for generating other offspring's and 6 mutated offspring's are generated as shown in table 3.4

4 . Generation of optimized chromosome

In order to produced the most efficient chromosomes, again the 1^{st} six parent chromosomes from population generation(Table 3) and six Mutated offspring's (Table 3.4) are applied with all the above operators of "G A".

5 . Generation of inverted chromosome

In order to produce the inverted chromosome 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'

6. Performance Evaluation

The performance of the proposed approach is evaluated using the different data set. The total cost of different data set is evaluated using the proposed approach as well as for the inverted chromosome obtained from the best chromosome. The optimized reordering point and the optimized ordering quantity is obtained by the best chromosome and the total cost is calculated, consequently inverted reordering point is generated from the best chromosome.(i.e.) 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' subsequently the total cost is calculated for the same inverted chromosome. The total cost of proposed system is denoted by T cost and the total cost of present system is denoted by T cost1.

7. conclusions

The main aspects of the inventory control in the manufacturing plant is to reduce the total cost. By applying the tools of genetic algorithm for the real data collected from RTPP the performance of the proposed system and the present system are compared and the results proved that the proposed is more efficient and also the cost can also be reduced.

8. References

Adel A. Ghobbar, Chris H. Friend, “The material requirements planning system for aircraft maintenance and inventory control: a note”, Journal of Air Transport Management, Vol.10, p.p.217–221, 2004.

Alp Muharremoglu, Nan Yang "Inventory Management with an Exogenous Supply Process" Operational research, issue 0030-364X, Vol. 58, No. 1, pp. 111–129, 2010.

Arumugam Mahamani and Karanam Prahlada Rao,"Development of a spreadsheet vendor managed inventory model for a single echelon supply chain: case study”, Serbian Journal of Management Vol.5 (2), p.p.199 - 211, 2010.

Aslam, Farrukh, A. R. Gardezi and Nasir Hayat,” Design, Development and Analysis of Automated Storage and Retrieval System with Single and Dual Command Dispatching using MATLAB”, World Academy of Science, Engineering and Technology 2009.

Azizul Baten and Anton Abdulbasah Kamil, "Direct solution of Riccati equation arising in inventory production control in a Stochastic manufacturing system",International Journal of the Physical Sciences Vol. 5(7), pp. 931-934, July 2010.

Behnam Fahimnia, Lee Luong, Remo Marian,” Optimization/simulation modeling of the integrated production distribution Plan: an innovative survey”, Wseas transaction on business and economics, No. 3, Vol. 5, March 2008.

Chin-Hsiung Hsu, Ching-Shih Tsou, and Fong-Jung Yu,” Multicritiria tradeoff in inventory control using memetic particle swarm optimization”, International Journal of Innovative Computing, Information and Control, No. 11(A), Vol. 5, November 2009.

Chitriki Thotappa, Dr. K.Ravindranath,” Data mining Aided Proficient Approach for Optimal Inventory Control in Supply Chain Management”, Proceedings of the World Congress on Engineering, Vol. 1, 2010.

Charu Chandra, Jānis Grabis,"Supply chain configuration using simulation based optimization", Proceedings of the 35th conference on winter simulation: driving innovation, 2003.

Haruhiko Tominaga, Tatsushi Nishi, and Masami Konishi, "Effects of inventory control on bullwhip in supply chain planning for multiple company”, International Journal of Innovative Computing, Information and Control, No. 3, Vol. 4, March 2008.

Ismail,E. Hashim,J.A.Ghani,R.Zulkifli,N.Kamilah,M. N. A. Rahman,"Implementation of EIS: A Study at Malaysian SMES",European Journal of Scientific Research, No.2 ,Vol..30, pp.215- 223 , 2009.

Jin-Hwa Song and Martin Savelsbergh, "Performance Measurement for Inventory Routing", Institute for Operations Research and the Management Sciences, Vol. 41, No.1, February 2007.

Luca Bertazz, Martin Savelsbergh, and M. Grazia Speranza "Inventory Routing",transportation Science, Vol.36 , p.p.44-54,February 2002 .

Ling-Feng Hsieh, Chao-Jung Huang and Chien-Lin Huang, "Applying Particle Swarm Optimization to Schedule Order Picking Routes in a Distribution Center", Asian Journal of Management and Humanity Sciences, Vol. 1, No. 4, p.p. 558-576, 2007.

M. Sreenivas, T.Srinivas,” Effectiveness of Distribution Network”, International Journal of Information Systems and Supply Chain Management, Int’l Journal of Information Systems and Supply Chain Management, Int’l Journal of Information Systems and Supply Chain Management, Vol.1(1), p.p.80-86 , January-March 2008.

Maria Sarmiento and Rakesh Nagiy,"A Review of Integrated Analysis of Production-Distribution Systems", IIE Transaction ,Vol. 31, No. 11, P.p. 1061 - 1074, 1999.

M.Zadieh and S.Molla-Alizadeh-Zavaedehi,"synchronized production and distribution scheduling with Due window ", Journal of applied sciences, Vol. 8(15), p.p. 2752-2757, 2008.

Patrick J. Rondeau Lewis A. literal, “Evaluation of manufacturing control system: from reorder point to enterprise resource planning", production and inventory management journals, 2001.

Peters mileff, Károly Nehez,Tibor Toth,” A new inventory control method for supply chain management”, In proceeding of the 12th International Conference on Machine Design and Production, September 2006.

P.Radhakrishnan , Dr. V.M.Prasad ,Dr. M. R. Gopalan,” Inventory Optimization in Supply Chain Management using Genetic Algorithm”, IJCSNS International Journal of Computer Science and Network Security, No.1,Vol.9, January 2009.

Philip Doganis, Eleni Aggelogiannaki, and Haralambos Sarimveis," A Model Predictive Control and Time Series orecasting Framework for Supply Chain Management", World Academy of Science, Engineering and Technology 2006.

QM. He a, E.M. Jewkes b, J. Buzacott c,"Optimal and near-optimal inventory control policies for a make-to-order inventory–production system "European Journal of Operational Research Vol.141 , p.p.113–132,2002.

Sanjoy Kumar Paul, Abdullahil Azeem,"Selection of the optimal number of shifts in fuzzy environment: manufacturing company's facility application", journals of industrial Engineering and management, vol.3, no.1. p.p. 54-67 ,2010.

Soheil Sadi-Nezhad a, Shima Memar Nahavandia and Jamshid Nazemia,"Periodic and continuous inventory models in the presence of fuzzy costs", International Journal of Industrial Engineering Computations, 2010.

SombaSt indhuchao,"A Very Large Scale Neighborhood (Vlsn) Search Algorithm for an Inventory- Routing Problem", ThammasatI nt. J.Sc.Tech. ,Vol.ll. , October-December 2006.

Steven P. Landry, Monterey,"Do Modern Japanese Inventory Methods Apply To Hong Kong?", International Business & Economics Research Journal ,Vol. 7, No. 4 April 2008.

S Shakeel ahamed, G. Ranga Janardhana,E.L.Nagesh,"GA Based Inventory Control for Manufacturing Unit", published in American Journal of Scientific Research,2011.

Thomas Fiig, Karl Isler , Craig Hopperstad ,Peter Belobaba ," Optimization of Mixed Fare Structures: Theory and Applications", Journal of Revenue and Pricing Management, 7th April 2009.

V. A. Temeng,P. A. Eshun,P. R. K. Essey, "Application of Inventory Management Principles to Explosive Products Manufacturing and Supply – A Case Study", International Research Journal of Finance and Economics, 2010.

Zuo-Jun Max Shen and Lian Qi,"Incorporating inventory and routing costs in strategic location models", European Journal of Operational Research 2006.

Table 1. Demand matrix (D1)

Raw materials	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
Aluminum cladding sheet(kgs)	100	100	100	100	150	150	200	250	350	300	100	100
M.S welding electrodes(nos)	5000	5000	5000	5000	6000	6000	10000	12000	12000	10000	2000	2000
S S welding electrodes (nos)	1000	2000	2000	1500	2500	2500	4500	4000	4500	3500	1000	1000
Fire proof wool (sq mts)	100	50	50	50	50	50	150	150	150	100	50	50
Gaskets(kgs)	5	3	5	5	4	1	12	10	10	15	5	5
Steam seals(kgs)	2	2.5	2.5	4	5	2	2	2	1	2	2.5	2.5
Bearings(nos)	200	200	250	250	200	150	200	150	100	100	50	150

The various cost of the raw materials incurred for RTPP are shown below.

Industrial Engineering Letters
ISSN 2224-6096 (print) ISSN 2225-0581 (online)
Vol 2, No.2, 2012

Table 1. 1. Various cost of raw materials.

S.no	Raw material	Purchasing cost	Transportation cost	Total cost T C	Holding cost 5% of T C	Shortage cost 4% of T C	Ordering cost
1	Al cladding sheet	185/-	15/-	200/-	10/-	8/-	2/-
2	MS welding electrodes	3.50 per piece	0.15	3.65/-	0.18/-	0.15/-	4/-
3	SS welding Electrodes	20/- per piece	1.60/-	21.60/-	1.08/-	0.9/-	6/-
4	Fire proof wool	160/-per sqm	13/-	173/-	9/-	7/-	2/-
5	Gaskets	350/- per kg	15/-	365/-	18.25/-	14.60/-	8/-
6	Steam seals	3200/- per kg	30/-	3230/-	161.50/-	130/-	2/-
7	Bearings	400/-	5/-	405/-	20.25/-	17/-	3/-

Table 2. Associated demand matrix

246	132	283	167	93	102	163	15	146	232	235	29
8380	6853	9532	11320	2737	13846	7705	13170	9298	2210	752	217
1061	2850	798	395	2997	5299	880	846	884	424	3880	2410
92	109	159	22	53	12	26	13	34	139	162	94
16	5	8	7	13	17	4	14	10	14	16	5
6	1	4	6	2	3	2	4	5	3	4	6
249	245	287	55	202	221	246	205	100	11	37	86

Table 3. population generation

'0001001'	'1100011'	'1101001'	'0011001'	'0110111'	'0011111'	'0110000'	'0101001'	'1010001'	'0101100'	'0101100'	'0110110'
'0101001'	'0110110'	'1111101'	'0110111'	'0101000'	'1110101'	'0011010'	'1100100'	'1111010'	'1100100'	'1100100'	'1111101'
'1000100'	'0001100'	'1011101'	'0111110'	'0010101'	'0100011'	'0111111'	'0111100'	'0011111'	'1010110'	'1010110'	'0100111'
'1010100'	'0100010'	'0101100'	'0010000'	'0010111'	'1100010'	'0101100'	'0000101'	'1010110'	'0000001'	'0000001'	'1011010'
'0110100'	'0010100'	'1001011'	'1001011'	'0110110'	'0011000'	'1111001'	'0010111'	'0100101'	'1001101'	'1001101'	'1010101'
'1101001'	'0100100'	'0001110'	'0011101'	'0001100'	'0100101'	'1110101'	'1011100'	'1010110'	'0110010'	'0110010'	'1000101'
'1011100'	'0111000'	'1110100'	'0110001'	'1001101'	'0001100'	'0000111'	'0111101'	'1011001'	'1110101'	'1110101'	'1011001'
'1111100'	'1000011'	'1110000'	'1001011'	'0111100'	'1001010'	'1011110'	'0010100'	'0001001'	'0000001'	'0000001'	'1010101'
'1000100'	'0111011'	'1101000'	'0100000'	'1011001'	'1010111'	'0100011'	'0101100'	'0100001'	'0111011'	'0111011'	'0010111'
'0101010'	'1110000'	'0100010'	'0100101'	'1011001'	'1000110'	'0110110'	'1001110'	'0011101'	'0110110'	'0110110'	'0010001'
'0001110'	'1000010'	'1001100'	'1001111'	'1010010'	'0110111'	'1000110'	'0011001'	'1010101'	'0111011'	'0111011'	'1111111'
'1001110'	'1111000'	'0000011'	'0100010'	'0000101'	'1010010'	'1111000'	'1011110'	'1101100'	'1100010'	'1100010'	'0010110'

Table 3.1 Fitness values of chromosome

1.87E-06	1.48E-06	1.47E-06	1.37E-06	1.34E-06	1.32E-06	1.32E-06	1.27E-06	1.25E-06	1.25E-06	1.15E-06	1.04E-06

Table3.1.1 process of cross over.

'0001001'	'0001001'		'0001001'	'1100011'	
'0101001'	'0101001'		'0101001'	'0110110'	
'1000100'	'1000100'		'1000100'	'0001100'	
'1010100'	'1010100'		'1010100'	'0100010'	
'0110100'	'0110100'		'0110100'	'0010100'	
'0100100'	'1101001'			'0100100'	'0100100'
'0111000'	'1011100'			'0111000'	'0111000'
'1000011'	'1111100'			'1000011'	'1000011'
'0111011'	'1000100'			'0111011'	'0111011'
'1110000'	'0101010'			'1110000'	'1110000'
'1000010'	'0001110'			'1000010'	'1000010'
'1111000'	'1001110'			'1111000'	'1111000'

Table3.2 Generation of offspring's

Offspring 1	**2**	**3**	**4**	**5**	**6**
'0001001'	'1100011'	'1101001'	'0011001'	'0110111'	'0011111'
'0101001'	'0110110'	'1111101'	'0110111'	'0101000'	'1110101'
'1000100'	'0001100'	'1011101'	'0111110'	'0010101'	'0100011'
'1010100'	'0100010'	'0101100'	'0010000'	'0010111'	'1100010'
'0110100'	'0010100'	'1001011'	'1001011'	'0110110'	'0011000'
'0100100'	'0001110'	'0011101'	'0001100'	'1101001'	'1101001'
'0111000'	'1110100'	'0110001'	'1001101'	'1011100'	'1011100'
'1000011'	'1110000'	'1001011'	'0111100'	'1111100'	'1111100'
'0111011'	'1101000'	'0100000'	'1011001'	'1000100'	'1000100'
'1110000'	'0100010'	'0100101'	'1011001'	'0101010'	'0101010'
'1000010'	'1001100'	'1001111'	'1010010'	'0001110'	'0001110'
'1111000'	'0000011'	'0100010'	'0000101'	'1001110'	'1001110'

Table 3.3 Mutation process

	I=5		**I=5**		**I=2**		**I=2**		**I=5**		**I=4**
Offspring 1	**MV=87 1010111**	**Offsing2**	**MV=121 1111001**	**Offspring 3**	**MV=91 1011011**	**Offspring4**	**MV=16 0010000**	**Offspring 5**	**MV=58 0111010**	**Offspring 6**	
'0001001'	'0001001'	'1100011'	'1100011'	'1101001'	'1101001'	'0011001'	'0011001'	'0110111'	'0110111'	'0011111'	'0011111'
'0101001'	'0101001'	'0110110'	'0110110'	'1111101'	'1011011'	'0110111'	'0010000'	'0101000'	'0101000'	'1110101'	'1110101'
'1000100'	'1000100'	'0001100'	'0001100'	'1011101'	'1011101'	'0111110'	'0111110'	'0010101'	'0010101'	'0100011'	'0100011'
'1010100'	'1010100'	'0100010'	'0100010'	'0101100'	'0101100'	'0010000'	'0010000'	'0010111'	'0010111'	'1100010'	'1010101'
'0110100'	'1010111'	'0010100'	'1111001'	'1001011'	'1001011'	'1001011'	'1001011'	'0110110'	'0111010'	'0011000'	'0011000'
'0100100'	'0100100'	'0001110'	'0001110'	'0011101'	'0011101'	'0001100'	'0001100'	'1101001'	'1101001'	'1101001'	'1101001'
'0111000'	'0111000'	'1110100'	'1110100'	'0110001'	'0110001'	'1001101'	'1001101'	'1011100'	'1011100'	'1011100'	'1011100'
'1000011'	'1000011'	'1110000'	'1110000'	'1001011'	'1001011'	'0111100'	'0111100'	'1111100'	'1111100'	'1111100'	'1111100'
'0111011'	'0111011'	'1101000'	'1101000'	'0100000'	'0100000'	'1011001'	'1011001'	'1000100'	'1000100'	'1000100'	'1000100'
'1110000'	'1110000'	'0100010'	'0100010'	'0100101'	'0100101'	'1011001'	'1011001'	'0101010'	'0101010'	'0101010'	'0101010'
'1000010'	'1000010'	'1001100'	'1001100'	'1001111'	'1001111'	'1010010'	'1010010'	'0001110'	'0001110'	'0001110'	'0001110'
'1111000'	'1111000'	'0000011'	'0000011'	'0100010'	'0100010'	'0000101'	'0000101'	'1001110'	'1001110'	'1001110'	'1001110'

Table 3.4 Mutated Offspring's

'0001001'	'1100011'	'1101001'	'0011001'	'0110111'	'0011111'
'0101001'	'0110110'	'1011011'	'0010000'	'0101000'	'1110101'
'1000100'	'0001100'	'1011101'	'0111110'	'0010101'	'0100011'

'1010100'	'0100010'	'0101100'	'0010000'	'0010111'	'1010101'
'1010111'	'1111001'	'1001011'	'1001011'	'0111010'	'0011000'
'0100100'	'0001110'	'0011101'	'0001100'	'1101001'	'1101001'
'0111000'	'1110100'	'0110001'	'1001101'	'1011100'	'1011100'
'1000011'	'1110000'	'1001011'	'0111100'	'1111100'	'1111100'
'0111011'	'1101000'	'0100000'	'1011001'	'1000100'	'1000100'
'1110000'	'0100010'	'0100101'	'1011001'	'0101010'	'0101010'
'1000010'	'1001100'	'1001111'	'1010010'	'0001110'	'0001110'
'1111000'	'0000011'	'0100010'	'0000101'	'1001110'	'1001110'

Table 4. New chromosomes ,6 from population generation and 6 from mutated offspring's.

'0001001'	'1100011'	'1101001'	'0011001'	'0110111'	'0011111'	'0001001'	'1100011'	'1101001'	'0011001'	'0110111'	'0011111'
'0101001'	'0110110'	'1111101'	'0110111'	'0101000'	'1110101'	'0101001'	'0110110'	'1111101'	'0110111'	'0101000'	'1110101'
'1000100'	'0001100'	'1011101'	'0111110'	'0010101'	'0100011'	'1000100'	'0001100'	'1011101'	'0111110'	'0010101'	'0100011'
'1010100'	'0100010'	'0101100'	'0010000'	'0010111'	'1100010'	'1010100'	'0100010'	'0101100'	'0010000'	'0010111'	'1100010'
'0110100'	'0010100'	'1001011'	'1001011'	'0110110'	'0011000'	'0110100'	'0010100'	'1001011'	'1001011'	'0110110'	'0011000'
'1101001'	'0100100'	'0001110'	'0011101'	'0001100'	'0100101'	'0100100'	'0001110'	'0011101'	'0001100'	'1101001'	'1101001'
'1011100'	'0111000'	'1110100'	'0110001'	'1001101'	'0001100'	'0111000'	'1110100'	'0110001'	'1001101'	'1011100'	'1011100'
'1111100'	'1000011'	'1110000'	'1001011'	'0111100'	'1001010'	'1000011'	'1110000'	'1001011'	'0111100'	'1111100'	'1111100'
'1000100'	'0111011'	'1101000'	'0100000'	'1011001'	'1010111'	'0111011'	'1101000'	'0100000'	'1011001'	'1000100'	'1000100'
'0101010'	'1110000'	'0100010'	'0100101'	'1011001'	'1000110'	'1110000'	'0100010'	'0100101'	'1011001'	'0101010'	'0101010'
'0001110'	'1000010'	'1001100'	'1001111'	'1010010'	'0110111'	'1000010'	'1001100'	'1001111'	'1010010'	'0001110'	'0001110'
'1001110'	'1111000'	'0000011'	'0100010'	'0000101'	'1010010'	'1111000'	'0000011'	'0100010'	'0000101'	'1001110'	'1001110'

Table 4.1 optimized chromosome.

1	**0**	**0**	**1**	**0**	**0**	**0**
0	**0**	**0**	**1**	**1**	**1**	**0**
0	**0**	**0**	**1**	**1**	**0**	**0**
0	**0**	**1**	**1**	**0**	**0**	**0**
0	**0**	**0**	**1**	**0**	**0**	**0**
0	**0**	**0**	**1**	**1**	**0**	**1**
1	**0**	**0**	**1**	**1**	**1**	**0**
1	**0**	**0**	**0**	**1**	**0**	**0**
0	**0**	**1**	**0**	**1**	**1**	**0**
1	**0**	**0**	**1**	**1**	**0**	**1**
0	**1**	**0**	**0**	**0**	**1**	**0**
1	**0**	**1**	**0**	**1**	**0**	**0**

Table 5. Inverted chromosome

0	**1**	**1**	**0**	**1**	**1**	**1**
1	**1**	**1**	**0**	**0**	**0**	**1**
1	**1**	**1**	**0**	**0**	**0**	**1**
1	**1**	**0**	**0**	**1**	**1**	**1**
1	**1**	**1**	**0**	**1**	**1**	**1**
1	**1**	**1**	**0**	**0**	**1**	**0**
0	**1**	**1**	**0**	**0**	**0**	**1**
0	**1**	**1**	**1**	**0**	**1**	**1**
1	**1**	**0**	**1**	**0**	**0**	**1**
0	**1**	**1**	**0**	**0**	**1**	**0**
1	**0**	**1**	**1**	**1**	**0**	**1**
0	**1**	**0**	**1**	**0**	**1**	**1**

Table 6. Performance Comparison results of total cost.

Data Set:1, Iteration Limitation = 50	
Total cost of Proposed ordering point(T cost)	Total cost of inverse Ordering point(T cost1)
5460742.0000000	9366567.5000000
3161336.20000000	9983010.80000000
3771384.0000000	9864896.0000000
4733264.90000000	9575622.10000000
5500463.50000000	10031103.5000000

7. Convergence graph of total cost for various Iterations.

X axis= Iterations

Y axis = Fitness values

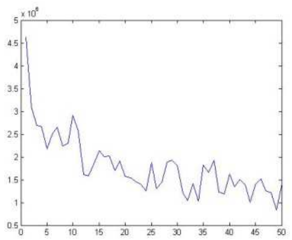

Figure 1. Graph

7.1 Bar chart showing the difference between the proposed system and present system.

Tcost1(Total ordering cost of present system) = 93,66,567.500

T cost(Total ordering cost of proposed system) = 54,60,742.00

Difference(savings by the proposed system) = 39,05,825.00

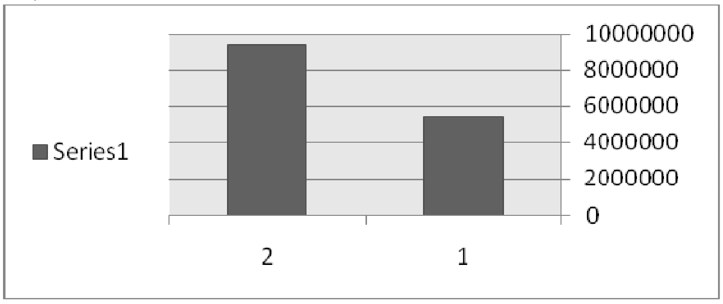

Figure 2. Bar chart comparison.

Interpretation. The above bar chart clearly represents that the total ordering cost of proposed system is less compared to the total ordering cost of present system.