©2010 International Journal of Computer Applications (0975 - 8887)
Volume 1 – No. 21

Optimizing Maintenance Activities Using HGA and Monte Carlo Simulation

Mahadevan ML
Asst. Professor,
Dept of Mechanical Engineering,
Thiagarajar College of Engineering,
Madurai – 625015, India

Paul Robert T
Professor
Dept of Industrial Engineering,
College of Engineering Guindy,
chennai – 600 025, India

Vignesh kumar V
Sridhar S
Graduate Student,
Dept of Mechanical Engineering,
Thiagarajar College of Engineering,
Madurai – 625015, India

ABSTRACT
The present industrial environment needs proper maintenance for effective functioning of the system underlining the need for an optimal maintenance planning. Maintenance planning is a complex and an inherently stochastic process. This paper presents maintenance planning problem for a process industry. The problem is formulated to determine which of the possible actions viz. maintenance or replacement is to be carried out for the subsystems during the planning period. Maintenance is carried out by analyzing improvement in the parameters (viz. MTBF & MTTR) during the planning period. The objective is to minimize the present value of total costs that are incurred by the decision taken during the planning period. The problem is effectively solved by hybrid genetic algorithm (HGA) technique.

Categories and Subject Descriptors
I.6.4 [Computing Methodologies]: Model Validation and Analysis.

General Terms
Design, verification.

Keywords
Maintenance planning, MTBF, MTTR, Hybrid Genetic algorithm, Optimization.

1. INTRODUCTION
Maintenance is the combination of any action carried out to retain an item or to restore it to an acceptable working condition. It can be achieved by the support of management and effective scheduling process. The purpose of maintenance is to extend the lifetime of the equipment or the mean time between two consecutive failures (MTTF) whose repair may be costly. The maintenance scheduling problem is concerned with scheduling essential maintenance activities over a fixed planning horizon for the critical equipments minimizing the maintenance costs and providing enough capacity to meet the anticipated demand.

The cost and effectiveness of the combined production and maintenance system has been analyzed by a simulation programme [4]. Comparison approach has been made between Genetic Algorithm and Simulated Annealing in optimizing the output of biological pathways [8]. The need for the usage of Hybrid Genetic Algorithm has been described [1]. A high speed achieving architecture of GA and SA has been proposed [5]. The proposed architecture realizes flexibility for many genetic operations on GA. The implementation of Hybrid Genetic Algorithm for the standard Traveling salesman problem has been done [2]. It has been designed by combining a variant of already existing crossover operator with heuristic algorithm. A Genetic and Simulated Annealing based algorithm for solving the flow assignment problems in computer networks has been proposed [10]. A Hybrid Genetic Algorithm using probabilistic selection has been proposed [7]. Simulated annealing and genetic algorithms for minimizing mean flow time in an open shop has been proposed [6]. Hybrid simulated annealing/genetic algorithm approach to short-term hydro-thermal scheduling with multiple thermal plants has been proposed [9]. Synthesis of multi-stream heat exchanger network for multi-period operation with genetic/simulated annealing algorithms has been proposed [11].

2. PROBLEM DESCRIBTION
In a cement plant, raw-mill process is one of the critical processes. In the raw-mill process, the lime ore is pulverized in different stages and supplied to the clinkerization process. A schematic diagram of a raw-mill process system for a cement process industry is shown in figure 1. Failure and repair data are required for a system performance study. The failure rates per year and mean time to repair of each of the components in the system were referred from the in-house plant records. Various corrective maintenance actions on the critical components of the raw-mill system are proposed in the study to maximize the system performance at minimum cost. In practice, these two corrective maintenance policies viz. increase in MTBF and/or decrease in MTTR need to be exclusive. They may be extreme policies for all possible levels of improvements to both MTBF and MTTR or some combination of two with each policy, all of which can be implemented to a different extent. Feasible improvements to the MTBF and/or MTTR at 5%, 10% and 15 % levels in components are considered, if the choice will be maintenance action. So the objective of the study is to determine the most cost effective maintenance policy

The identified subsystems are:

(1) Booster fan (2) Conveyor roller assembly (3) Air slide - membrane system (4) Silo feed elevator-drive system (5) Separator system (6) Impact crusher - rotor system (7) Raw-mill gear system

The problem considered is a maintenance optimization problem for a process industry. There are seven subsystems sequentially operated in series configuration. Maintenance or replacement activity in any of these subsystem results in cessation of the entire process. In this problem 6 periods are considered and for each period the combination of components to undergo maintenance or replacement activity is to be obtained. Maintenance and replacement costs, time to repair, downtime cost, failure cost and standby cost are all included in the calculation.

Nomenclature

- X_{ij} A binary variable for maintenance choice for subsystem 'i' at j^{th} period
- Y_{ij} A binary variable for the replacement for subsystem '*i*' at j^{th} period
- M_{ij} The Maintenance cost for subsystem '*i*' at period '*j*'
- R_{ij} The Replacement cost for subsystem '*i*' at period '*j*'
- M_{ci} Current maintenance cost for subsystem '*i*'
- m Rate of inflation for maintenance
- R_{ci} Current replacement cost for subsystem '*i*'
- r Rate of inflation for replacement
- s Number of stages of improvement in MTTR or MTBF
- k_1 Discount factor per period
- P_n Improvement stages for MTBF (0%, 5%, 10%, 15%)
- S_c Standby cost
- Q_n Improvement stages for MTTR (0%, 5%, 10%, 15%)
- d_{jp} Downtime due to scheduled PM during 'j^{th}' period (in hours)
- C_{sd} Downtime cost of the system per hour
- d_{jr} Downtime due to unscheduled repair during 'j^{th}' period (in hours)
- $R(t)_{ij}$ Reliability of the subsystem '*i*' at j^{th} period
- $R(t)_{target}$ Reliability target of the subsystem '*i*' at j^{th} period
- K Present value factor
- t_{ij} Age of the subsystem at the end of the period '*j*'
- β_i ,η_i Weibull parameters for subsystem '*i*'

Objective function

Minimize Total cost, $$Z = \sum_{j=1}^{6} \frac{(C_j + (d_{jp} \text{x} C_{sd})) + (d_{jr} \text{x} C_{sd})}{(1+k)^j} \quad (1)$$

subject to the reliability constraint,

$$R(t)_{ij} \geq R(t)_{target} \quad (2)$$

where,

$$C_j = \sum_{i=1}^{7} \{X_{ij} M_{ij} + Y_{ij} R_{ij} + X_{ij} (B_{ij} + MR_{ij})\} \quad (3)$$

$\forall j = 1, 2, \ldots 6$

$$M_{ij} = M_{ci} (1+m)^j \quad (4)$$

$$R_{ij} = R_{ci} (1+r)^j \quad (5)$$

$$B_{ij} = \sum_{n=1}^{s} k_1 (1+P_n)^n \quad (6)$$

$$MR_{ij} = k_1 * S_c \quad (7)$$

$$MR_{ijs} = MR_{ijs-1} + MR_{ijs-1}(1+Q_n) \qquad \forall s = 2,3 \quad (8)$$

$$R(t)_{ij} = \text{Exp}(-(t_{ij} / \beta_i)^{\eta_i}) \quad (9)$$

3. METHODOLOGY

3.1 Genetic Algorithm

Genetic algorithm (GA) is known to be an efficient search and optimization mechanism which incorporates the rules of natural selection [3]. Genetic algorithms are randomized search and optimization techniques, guided by the principles of evolution and natural genetics, which search large and complex landscapes using implicit parallel searching capability and can provide optimal or near optimal solution. For every GA procedure there are a number of parameters which, in the coded form, represent an infinitely different solution. The target is to find the best possible representative solution maintained by a finite sized population. The value of the objective function for a particular solution is determined by the corresponding parameter values. Basically, three genetic operators are used in GAs namely selection, crossover and mutation. A fixed population size is considered. Each member of the population is called a chromosome. The initial population is generated randomly or using some domain specific knowledge. A chromosome is a concatenation of the encoded parameter values. Encoding is done in binary for basic GA.

Simple Genetic Algorithm

```
begin
    g = 0
    initialize P (g)
    evaluate P (g) using fitness function
    termination_condition = false
    while (NOT termination_condition) do
    begin
        g = g + 1
        select parents from P (g)
        crossover
        mutation
        evaluate P (g + 1) using fitness function
    end
end
```

3.2 Hybrid Genetic Algorithm

There are many algorithm techniques and their variations being used for solving complex optimization problems. Due to the inherent parallel search capability, genetic algorithms are considered as one of the best tools for solving such problems. Hybrid genetic algorithm is the combination of search methods like GA and SA. In order to develop a hybrid algorithm for a better solution, we have incorporated an SA selection criterion in a GA framework.

3.2.1 SA concept (selection)
SA is a local search algorithm that exploits an analogy between the ways in which a metal cools and freezes to a minimum energy crystalline structure (Annealing process). This process is used as a selection operator in HGA to search for a minimum in a more general system. SA's major advantage over other methods is its ability to avoid becoming trapped at local minima.

3.2.2 Crossover
Crossover operates on two chromosomes at a time and generates offspring by combining both chromosomes features. In single point crossover of two chromosomes, a point is selected in both chromosomes and the part of the chromosome before or after this point is replaced by the similar part from the other chromosome. A higher crossover rate (p_c) allows the exploration of large solution space and reduces the chances of false optimum

3.2.3 Mutation
Mutation involves flipping a single bit in the chromosome. It replaces the genes lost from the population during the selection process or provides the genes that were absent in the initial population. The mutation rate (p_m) controls the rates at which new genes are introduced into the population. Very low mutation rate would neglect many useful genes and a very high value would result in large amount of random perturbation, loss of parent-offspring resemblance and the algorithm will finally lose the ability to learn from the history of search.

Hybrid Genetic Algorithm

Notations:

pop	population size
gen	number of generations
p_c	Probability of cross over
p_m	Probability of mutation
C_L	chromosome length
RN	Random number
ch_{ij}	chromosome vector of generation 'i' and population 'j'
Z_b	Best objective function value

Input *pop, gen,* p_c, p_m and C_L
generation i=0
generate initial population
create chromosomes ch_{ij}
for population j=1 to *pop*
- For bit k=1 to C_L
- Generate $RN \in$ U[0,1] for n(no. of critical subsystems)
- Generate $RN \in$ U[0,3] for the remaining bits (Improvement)
- Next k

next j
for population N=1 to *pop*
- Perform cross over operation
- for population j=1 to *pop*
    - consider ch_{ij} and ch_{ij+1}
    - Generate $RN \in$ U[0,1]
    - If ($RN \le p_c$)
      { determine the cross over site in ch_{ij}
        Perform the cross over
      }
- next j
- / To perform the mutation operation
    - for bit k=1 to C_L
    - generate $RN \in$ U[0,1]
    - If ($RN \le p_m$)
      { choose the mutation site in ch_{ij}
        flip the bit
      }
    - next k
- Evaluate the ch_{ij} for reliability constraint
- Run simulation to determine Fitness value F(x)

next N
- Report best ch_{ij} *and compute* Z_b

for generation i=1 to (*gen-1)*
- Sort ch_{ij} based on the F(x)
- Reproduce ch_{ij} using Simulated Annealing
- for population j=1 to *pop*
    - Perform cross over operation
    - perform the mutation operation
    - Evaluate the ch_{ij} for reliability constraint
    - Run simulation to determine Fitness value F(x)
- next j
- Report best ch_{ij} and compute Z_b
- Update Z_b and corresponding ch_{ij}^b
- next i

// **Simulated Annealing algorithm**
begin
  for cr = 1 to pop
  initialize T and select 2 chromosome
  for i=1 to 20
    Run simulation to det F(x) and get Fmax
   if Fmax – F(x) ≤ 0
      then select x and set fmax to f(x)
      else if exp[-(Fmax- F(x))/T] > RN[0, 1]
      then select x
        else select x corresponding to fmax
     next i
  lower T
end

3.3 HGA Implementation
The maintenance optimization problem is totally analyzed for 6 periods. HGA is applied to choose the best combination in a simulation run.

3.3.1 Chromosome generation
In this problem, a fixed length binary string of 21 bits is considered. Among the first 7 bits of the genotype the bit represent maintenance activity/ replacement activity. Among the next 14 bits the bits in odd places represents the percentage increase in MTBF and in even places represents the reduction in MTTR of the corresponding components.

Chromosome:

(Decision bits)	(Improvement bits)
1 1 0 0 1 1 0	1 3 0 0 0 0 2 1 3 2 0 0 1 3

The process starts with random selection of two chromosomes. A temperature value of 700 is fixed. Now, the chromosomes are decoded and optimal chromosome is chosen as best chromosome. Next another chromosome is randomly generated and compared with the best chromosome. If the old best chromosome is still the best, t process is continued. But if the newly generated chromosome is found as best chromosome, metropolis algorithm is adopted. This process is repeated for 20 times. Now the temperature value is reduced by a small factor of 3%. Likewise the above process is repeated till the temperature is reduced to 30. The final chromosome is one best chromosome. Like this 40 chromosomes are generated. Now the 40 chromosomes are ready for crossover and mutation. A random choice of a pair of chromosomes is made from the 40, crossover probability is checked and then the crossover is done. The assumed crossover probability is 0.8. Now the set is ready for mutation. A random choice of chromosomes is made and is checked with mutation probability of 0.05. The mutation locations are chosen in a random fashion and the bits in the chromosome are mutated. Then the C_{ij} value is evaluated by extracting information from 40 chromosomes. Using the data from the database for the first period and the chromosome which gives the minimum cost for that period is selected. The above all procedures are to be carried out for 3 years. The finally arrived 40 chromosomes are used for the calculation of the value of the objective function. Finally, the total cost and the combination of activity for various components in various periods are computed.

3.4 HGA Validation

The formulated maintenance model is validated by using the data given in Table 1.

α=0.8; C_{sd} = \$33.32/min; k = 0.1; m = 0.1; r = 0.08

The parameters are tuned by means of carrying out the sensitivity analysis. Crossover probability=0.8; Mutation probability = 0.05, Initial Temperature= 700, final Temperature = 30, Temperature reduction factor=3%.

4. CONCLUSION

The implementation of Hybrid Genetic Algorithm for maintenance planning provides better results compared to Simple GA results. The obtained combination of replacement and maintenance activities for all the subsystems gives the minimum cost for the given number of periods. Thus the heuristic approach provides optimum solution by improving the availability of the machines. This maintenance optimization model may be applied to any type of similar process industries and effective maintenance planning can be carried out. This work can be extended by taking into account the impact of risk factor of the system. Table 2 shows the final result of the simulation study.

5. REFERENCES

[1] Akeel Al-Attar, A Hybrid GA-Heuristic Search Strategy, 1994 issue of AI Expert USA. (Sep. 1994), 1-5.

[2] Andal Jayalakshmi, G , Sathiyamoorthy, S, Rajaram, R , A Hybrid Genetic Algorithm- a new approach to solve travelling salesman problem, 1-17.

[3] D E Goldberg. .Genetic Algorithms in Search, Optimization and Machine Learning Addison-Wesley, New York, 1989.

[4] Husband and Baskar, 1982, Optimizing maintenance/production systems, Maintenance management international, 75-81.

[5] Masaya Yoshikawa, Hironori Yamauchi, and Hidekazu Terai, Hybrid Architecture of Genetic Algorithm and Simulated Annealing, Engineering letters, 16:3, EL_16_3_11, 20 August 2008, 1-7.

[6] Michael Andresena , Heidemarie Bräsela, 2008, Simulated annealing and genetic algorithms for minimizing mean flow time in an open shop, Otto-von-Guericke-Universität, Fakultät für Mathematik, PSF 4120, 39016 Magdeburg, Germany. Press

[7] Pakhira M.K , A Hybrid Genetic Algorithm using Probabilistic Selection, IE (I) Journal, l84, (May 2003), 23-30.

[8] Rajesh Krishnan, Carla C. Purdy, Comparison of Simulated Annealing and Genetic Algorithm approaches in optimizing the output of Biological pathways, 1-8.

[9] Suzannah Yin Wa Wong, 2000, Hybrid simulated annealing/genetic algorithm approach to short-term hydro-thermal scheduling with multiple thermal plants. Department of Computer and Mathemetics, School of Continuing Studies, Chinese University of Hong Kong, 67 Chatham Road South, 13/F, Kowloon, Hong Kong, People's Republic of China.

[10] Tarek M. Mahmoud, A Genetic and Simulated Annealing Based Algorithms for solving the Flow Assignment problem in Computer networks, International Journal of Electronics, Circuits and Systems, 1, 2, 128-134.

[11] Xiangkun MAa, Pingjing YAO, Xing LUO and Wilfried ROETZEL, 2007, Synthesis of multi-stream heat exchanger network for multi-period operation with genetic/simulated annealing algorithms, Institute of Thermodynamics, University of the Federal Armed Forces Hamberg, Hamberg D-22039, Germany.

Table 1. Input values for the critical components

Critical components	$M1_i$ (\$)	$R1_i$ (\$)	Sc (\$)	β_i	η_i	TTR_i (min)
Booster fan	100	360	460	2.14	22687	90
Conveyor roller assembly	80	440	540	1.83	18647	40
Air slide	120	300	400	1.98	18263	120
Elevator	120	440	540	1.59	18170	150
Separator system	160	400	500	1.45	28459	60
Impact crusher	2400	700	800	1.14	32477	200
Raw-mill gearbox	3000	760	860	2.29	11788	80

Table 2. Simulation Results

PERIOD	SUBSYSTEM 1			SUBSYSTEM 2			SUBSYSTEM 3			SUBSYSTEM 4			SUBSYSTEM 5			SUBSYSTEM 6			SUBSYSTEM 7			DOWN TIME	C	REPAIR COST	TC
n	ACTIVITY	P	Q	ACTIVITY	P	Q	ACTIVITY	P	Q	ACTIVITY	P	Q	ACTIVITY	P	Q	ACTIVITY	P	Q	ACTIVITY	P	Q	HOURS	\$	\$	\$
1	M	0.05	0	M	0	0.1	M	0	0	M	0	0.05	M	0	0.05	M	0	0.05	M	0	0.05	711.5	22878.26665	4998	27876.2667
2	M	0	0	M	0	0	M	0	0	M	0	0.1	M	0	0	M	0	0.05	M	0	0.1	707	21667.11108	3998.4	25665.5111
3	M	0	0.1	M	0	0.1	M	0.1	0	M	0	0	M	0	0.1	M	0.05	0	M	0	0	721	21052.59255	2665.6	23718.1925
4	M	0	0.1	M	0	0.05	M	0.05	0	M	0.15	0	M	0	0	M	0	0	M	0.05	0.1	721	20078.87809	2665.6	22744.4781
5	M	0.05	0.05	M	0.1	0.1	M	0	0.1	M	0	0.15	M	0	0.1	M	0	0	M	0.05	0	691	18353.81397	3998.4	22352.214
6																								4998	4998
																								TOTAL COST PER CYCLE TIME	**127354.66**

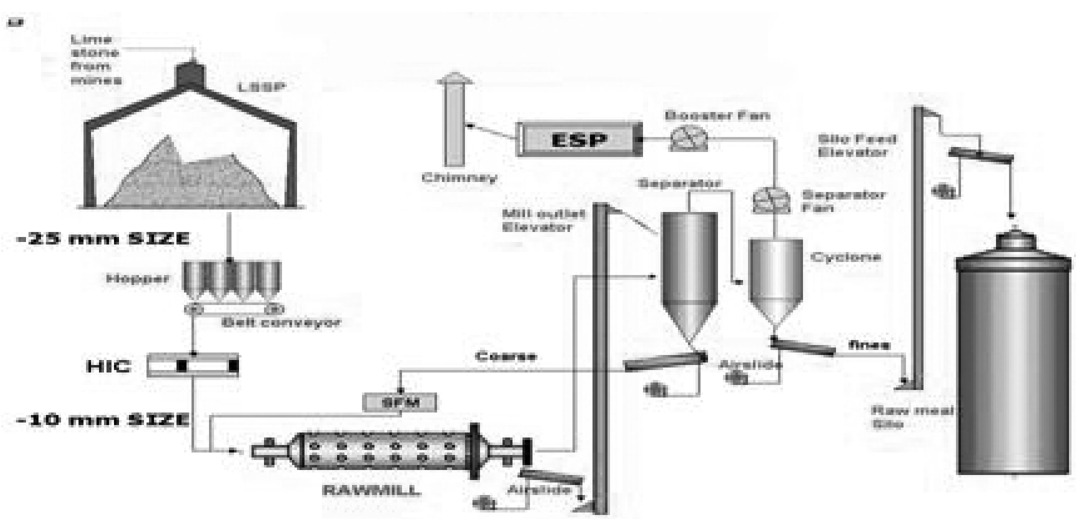

Figure 1. Raw-mill system of the cement industry