References

1. Cerny, V. (1985). Thermodynamic approaches to the travelling salesman problems. J. Optim. Theory Appl. Vol.45. pp: 41- 51

2. Cox, D. R. (1962). Renewal Theory, John Wiley

3. Fagihih, N. And Sohrabi, B. (1999). Genetic algorithm in optimal preventive part replacement for minimum downtime maintenance planning, Journal of Interdisciplinary Mathematics. Vol. 1. pp: 57- 80.

4. Fogel, I. J; Owens, A. J. and Walsh, M. J. (1966). Artificial intelligence through simulated evolution. John Wiley. New York.

5. Glover, F. (1990). tabu search: part 2. ORSA Journal on computing, Vol. 2. pp: 4- 32.

6. Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan press. Ann Arbor. MI.

7. Jardine, A. K. S. (1973). Maintenance, Replacement and Reliability, Pitman publishers.

8. Kirkpatrick, S.; Gelatt, D. D and Vecchi, M. P. (1983). Optimisation by Simulated Annealing. Science. Vol. 22. pp: 671- 680.

9. Sohrabi, B. (2003). Simulated annealing in optimal preventive part replacement for minimum downtime maintenance planning, Management Knowledge. Vol. 16. No. 82. pp: 140- 152.

10. Sohrabi, B (2000). Solving large-scale problem using heuristics algorithm, Ph. D. thesis, Lancaster University, U. K.

the ability to generate high quality feasibale solutions, whenever such a solution exist.

- In terms of quality of solution both algorithm (GA and SA) obtained same result but in terms of computation time simulated annealing prformance was better than genetic algorithm.
- Simulated annealing is easy to understand and also easier than genetic algorith to code.
- Simulated annealing can easily handel change in the objective function. But in enetic algorith me need to have fitness function, which some times makes problem.
- Simulated annealing can be simply stated and that lend themselves more readily to analysis

4. Conclusion

In this paper we focused in the performance of genetic algorithm and simulated annealing in optimal part replacement. Some evaluation criteria were explained. Although there are not too many application of SA and GA in optimal preventive part replacement for minimum downtime maintenance planning. But it was decided to recommend using simulated annealing approach in this topic.

3. Evaluation criteria for approximate algorithms

Some output are evidenced in table 3 and figure 1,which shows that the result produced by both algorithms are same. Some evaluation criteria are explained in order to compare the performance of algorithms. In other word, which algorithm is more suitable for optimal preventive part replacement?

a) Quality of solution and computation time

Solution quality and computation time of an algorithm are important criteria to assess the effectiveness of an algorithm. A reasonable running time is a very important element of algorithm evaluation and implementation. Therefore, a very desirable algorithm would be one that is equipped with a set of adjustable parameters, that would allow the user to meet changes in emphasis between cost and performance through controlling the trade-off between the quality of the solution and the amount of computational effort.

b) Code difficulty and ease of implementation

It is difficult to measure the intricacy and simplicity of coding of a particular algorithm.Algorithm principles must be simple, not cumbersome; generally applicable, not problem specific. This generality would enable an easy implementation of the algorithm to new domain areas with little a priori knowledge of the problem structure.

c) Flexibility

Since heuristic algorithms are typically involved in the solution of real world problems, it is important that they should be flexible. In particular, they should easily handle changes in the model, constraints and objective function.

d) Simplicity and analysability

There is a significant appeal of algorithms that can be simply stated and that lend themselves more readily to analysis. Extremely complex algorithms are less likely to be analyzed in terms of flexibility and quality of solution than a simple algorithm.

e) Robustness

This class of algorithms have a number of desirable characteristics including: the ability to perfrm parametric analysis; a good characterisation that would enable a user to prove that the solution is within a certain percentage error(devation) from the optimal solution;

Here follows a description of a genetic algorithm where k_1 is the number of generations and k_2 is the number of selections per generation. At each of iteration t = 1,..., k_1, apply k_2 times steps 1 through 3, then apply step4

Step 1, ***selection.*** Randomly select two parents from X^t, with a probability inversely related to the value of $D(T_r) = (T_r + \tau_r)/[\tau_r + g(T_r)\tau_f]$ the fitness function.

Step 2, ***one-point crossover.*** Create two offspring from the two parents by swapping a bit string located after a randomly selected cut point.

Step 3, ***mutation.*** Apply a random mutation to each offspring by flipping bits with small probability.

Step 4, ***new population.*** Obtain X^{t+1} from X^t by removing the $2k_2$ worst solutions in X^t and replace them with the $2k_2$ offspring just created.

Some output samples are evidence in figure 1, which show that generations finally settle at fixed values, rendering optimal solutions. The outputs read as xmax=5 and max=160.24, which indicates $T_r = 5$ (weeks) and minimum value of $D(T_r)$ as 1/160.24 = 0.0062. Therefore, the same results as observed from table 2, are obtained by the application of genetic algorithm.

Table 4. Output sample for generations 49 and 50

Population Report

Generation 49						Generations 50		
# string	x	fitness	#	parents	xsite	string	x	fitness
0) 0101	5.00	160.24	| 0)	(1 , 8)	0	0101	5.00	160.24
1) 0101	5.00	160.24	| 1)	(1 , 8)	0	0101	5.00	160.24
2) 0101	5.00	160.24	| 2)	(8 , 4)	1	0101	5.00	160.24
3) 0101	5.00	160.24	| 3)	(8 , 4)	1	0101	5.00	160.24
4) 0101	5.00	160.24	| 4)	(8 , 6)	3	0101	5.00	160.24
5) 0101	5.00	160.24	| 5)	(8 , 6)	3	0101	5.00	160.24
6) 0101	5.00	160.24	| 6)	(5 , 2)	1	0101	5.00	160.24
7) 0001	1.00	42.90	| 7)	(5 , 2)	1	0101	5.00	160.24
8) 0101	5.00	160.24	| 8)	(0 , 6)	0	0101	5.00	160.24
9) 0101	5.00	160.24	| 9)	(0 , 6)	0	0101	5.00	160.24

Note: Generation 50& Accumulated Statistics: xmax=5 max=160.24 min=160.24 avg= 160.24 sum=1602.43 nmutation=64 ncross=167

Else if random (0,1) $< \exp(-\sigma / T)$ then $i = j$;

$m = m + 1$;

Until $m = N(t)$;

$t = t + 1$;

$T = T(t)$;

Until stopping criterion true (number of iterations)

After some initial experiments (results not recorded) testing the effect of different values, a series of experimental runs was carried out where some parameters was given two possible values in the region of what was expected to be a suitable value. The table (3) show the different values for the parameters.

T : initial temperature

N: number of iterations

r: the temperature update rule

Sc: stopping criteria

$D(T_r)$: Minimum downtime

No.its: total of number of the iterations

CPU: total time in second

Table 3. Experiment with different values of the different parameters (T=200)

T	N	r	Sc	$D(T_r)$	No.its	CPU
200	100	0.99	50	0.0062	11057	421
200	100	0.99	20	0.0062	9687	367
200	100	0.90	50	0.0068	6939	187
200	100	0.90	20	0.0068	4569	134
200	50	0.99	50	0.0062	11057	421
200	50	0.99	20	0.0062	9687	367
200	50	0.90	50	0.0071	3339	157
200	50	0.90	20	0.0071	2969	123

The result produced by the computer program (table3) was same with the result that obtainable by ordinary methods.

2-2. Genetic algorithm procedure:

Genetic algorithms are rooted in the work of Fogel et al. [4] and Holland [10]. They work on a population of solutions. At each step, a new population is derived from the preceding one by combining some of its best elements and discarding the worst.

Table 1. Value of replacement function

N	0	1	2	3	4	5	6	7	8	9
g(n)	0	0.001	0.006	0.023	0.067	0.159	0.310	0.504	0.698	0.868

Table 2. Minimum downtime

T_r	1	2	3	4	5	6	7	8	9
$D(T_r)$	0.0232	0.0119	0.0082	0.0067	0.0062	0.0064	0.0068	0.0071	0.0072

2-1. Simulated annealing procedure

The simulated annealing algorithm has its origins in statistical mechanics. The interest in simulated annealing began with the work of Kirkpatrick [8], and Cerny [1]. They proposed a simulated annealing algorithm, which is based on the analogy between the annealing process of solids and the problem of solving optimization problems.

Having above defined all simulated annealing terminology and its analogy with statistical mechanics, simulated annealing can be seen as a generalized iterative improvement algorithm [10]. The simulated annealing algorithm steps are summarized as follows:

Select an initial state $i = D(T_1)$

$g(0) = 0,\ n = 1;$

$$g(n) = \sum_{i=0}^{n-1} \{[1 + g(n-i-1)] \int_{iT}^{(i+1)T} p(t)dt\},$$

$$D(T_r) = [\tau_r + g(T_r)\tau_f]/(T_r + \tau_r)$$

Select an initial temperature $T > 0;$

Set temperature change counter $t = 0;$

Repeat

Set repetition counter $m = 0;$

Repeat

Generate state j, a neighbour of i;

$n = n + 1;$

$$g(n) = \sum_{i=0}^{n-1} \{[1 + g(n-i-1)] \int_{iT}^{(i+1)T} p(t)dt\},$$

$$j = D(T_r) = [\tau_r + g(T_r)\tau_f]/(T_r + \tau_r)$$

Calculate $\sigma = f(j) - f(i);$

If $\sigma < 0$ then $i = j;$

$$g(n) = \sum_{i=0}^{n-1} \{[1 + g(n-i-1)] \int_{iT}^{(i+1)T} p(t)dt\} \qquad (1)$$

Known as the replacement function, stats at $g(0) = 0$, due to the fact that $g(nt) = 0$, for $n = 0$, i.e., there is no renewal at the time origin $(t = 0)$. There on, it renders the number of renewals or replacements as $g(n)$, at discrete time intervals nT, for any number n, used as a multiple of an arbitrary constant time interval, T. the equipment is, however, supposed to fail according to some probability density function p(t). In optimal part replacement policies, the replacement function, given by equation (1), is used to formulate the required strategy. This is usually accomplished by minimizing or maximizing a desired objective function. As a major tendency, the optimality of part preventive replacements and also required due to the random failures that may occur. Hence, the function to be minimized can be written as [2]:

$$D(T_r) = [\tau_r + g(T_r)\tau_f]/(T_r + \tau_r) \qquad (2)$$

In the above equation, the total downtime $D(T_r)$ is minimized with respect to the replacement time T_r. The parameters τ_r and τ_f are the times required for a preventive replacement and a replacement forced due to failure, respectively. The value for $g(T_r)$ can also be computed from equation (1), for $T_r = nT$. In practice, the values of $g(T_r)$ are obtained, iteratively, from equation (1) and substituted in equation (2), to find the value of T_r which minimizes $D(T_r)$, as a numerical solution [2][3]. As a numerical example, consider an equipment with a failure characteristic following the Gaussian probability distribution function with parameters, $\mu = 7$ and $\delta = 2$. (weeks, day), as the mean and standard deviation, respectively. Suppose, also, that, $\tau_r = 0.0238$ and $\tau_f = 0.0476$ weeks. Hence, iterating equation (1), with $T = 1$(week) and $p(t)$ as a Gaussian probability density function with the given parameters, the values shown in table (1) can be obtained. Then substituting these values for $g(T_r)$, together with the given values of τ_r and τ_f, in equation (2), table (2) can be computed. As it can be seen in table (2), the minimum value of $D(T_r)$ occurs at $T_r = 5$ (weeks); that is, the optimum preventive replacement period, for minimum downtime is 5 weeks.

1. Introduction

In most of these problems, the optimal solution is computationally difficult to obtain. Hence, it is important to have approximate algorithms (heuristics) which can provide near optimal solutions for large - sized problems in a reasonable amount of computational time.

Of late, a lot of research attention has been focused on the development of intelligent and effective heuristic approaches that solve large problems of practical size. These approaches have evolved through interactions and analogies derived from biological, physical, computer and decision making sciences. New approaches to the approximate solution of difficult combinatorial problems include simulated annealing, tabu search, genetic algorithms and neural networks. The First derives from physical science - more specifically, from statistical mechanics. The second stems from the general tenets of intelligent problem solving. The last two are inspired by principles derived from biological sciences. See Glover, [5] for a further expose.

In this paper we just focus on the performance of simulated annealing (SA) and genetic algorithm (GA) in optimal preventive part replacement for minimum downtime maintenance planning. However, in the maintenance planning, optimal part replacement policies, mainly, employ statistical renewal theory, leading to minimization of a function such as downtime [7][2]. Hence, simulated annealing and genetic algorithms appear to be suitable tools for optimal maintenance planning. We review shortly in section 2 the results of studies carried out by the author ([3][9]) that consisted of the using simulated annealing and genetic algorithm in preventive part replacement. In section 3, some evaluation criteria will be explained in order to show which algorithm is more suitable to use in optimal preventive part replacement for minimum downtime maintenance planning. Finally section 4 presents the overall conclusion.

2. Applying simulated annealing and genetic algorithm in optimal part replacement

In the maintenance planning, part replacement strategies, mainly, rely on the renewal function, which in its numerical forms, appears as [7], [2]: