A case study on the application of a genetic algorithm for optimization of engine parameters

R Verma and **P A Lakshminarayanan***
Engine Research and Development, Ashok Leyland, Hosur, Tamilnadu, India

The manuscript was received on 5 December 2004 and was accepted after revision for publication on 21 December 2005.

DOI: 10.1243/09544070D09204

Abstract: Optimization of modern engines is becoming involved because of both the stringent emission norms and the newly found ability to control various parameters in every zone of the map of engines using electronic fuel injection systems, in both steady state and transient conditions. This paper describes a genetic algorithm and its application to engine optimization when a fitness criterion can be described quantitatively. A specific case study on the selection of a simple speed-dependent injection timer for a diesel engine is described. A population of timers is randomly created where the survival depends on the fitness criterion. From the fit and asexual parents that are randomly selected, offspring are produced to replace the least-fit portion of the population belonging to the previous generation. A small percentage of the population is allowed to mutate. The fitness of the population improves every generation and, in the advanced generation, the fittest timer seems to be the most optimum.

Keywords: genetic algorithm, diesel engines, optimization, injection timing, injection timer, electronic fuel injection system, population, generation, mutation, elitist reproduction strategy, design of experiments

1 INTRODUCTION

The diesel injection timing plays an important role in deciding the combustion characteristics, power, and fuel consumption of the engine. For a given static injection timing, the delay due to the time taken by the pressure waves to reach the injector from the high-pressure fuel injection pump would retard the fuel injection at higher speeds, relative to the piston position. The retarded injection timing leads to increased smoke formation, decreased power, and decreased engine efficiency. If the injection of fuel is more advanced, it results in higher cylinder temperature and pressure, and thus in more nitrogen oxides (NO_x).

An injection timer is used in diesel engines to advance the injection timing as the speed increases and hence to compensate for the delay. The injection timer is characterized by the timer path curve, which describes how the injection time is advanced with the speed. The path is usually optimized manually by using the experimental test data. To finalize a particular timer path, this process is repeated many times. The selected timer may not be the optimum as the number of tests is limited. With more and more modern diesel engines developed using common-rail or other electronic fuel injection systems, optimization of injection timing and injection pressure at every operating point is carried out to maintain the integrated emissions at a optimum level and to minimize the specific fuel consumption (SFC). While this offers a great opportunity to develop environmentally friendly and fuel-economic engines, the optimization work has increased manifold. With fast computers that could directly interact with the fuel injection systems, online optimization of the engine is carried out using the design-of-experiments (DOE) method [**1**]. For complex problems, the approach is difficult. On the other hand, the genetic algorithm (GA) method presents an interesting alternative to optimize an engine. GAs have not been explored for this purpose so far in the published literature.

In this paper, the development of an optimization scheme for the timer path using a GA is presented. In this method, the input is test data such as the SFC, power p, and emission for constant fuel injection

* *Corresponding author: Engine Research and Development, Ashok Leyland, Plant-1, Hosur, 635 126, Tamilnadu, India. email: pl_loganayagi@rediffmail.com*

timings, and the output is the optimized timer path that corresponds to the minimum SFC, maximum power, and emissions limited to a given range. The timer travel path is coded as a binary string. A modified GA is used to find the optimum timer path from a random solution space.

2 GENETIC ALGORITHM

Many optimization problems are very complex and hard to solve by conventional optimization techniques. Since the 1960s, there has been increasing interest in imitating living beings to solve hard optimization problems. Simulating the natural evolutionary process results in stochastic optimization techniques called evolutionary algorithms, which can often outperform conventional optimization methods when applied to difficult real-world problems [**2–5**]. There are currently three main avenues of this research, namely GAs, evolutionary programming, and evolution strategies. Of these, the GA is perhaps the most widely known type of evolutionary algorithm today. Details of GAs can be found in the book by Goldberg [**6**] and the paper by Hajela [**7**].

The GA is motivated by the hypothesized natural process of evolution in biological populations, where genetic information stored in strings of chromosomes evolves over generations to adapt favourably to a static or changing environment. The algorithm is based on the elitist reproduction strategy, where members of a population deemed the fittest are selected for reproduction and are given the opportunity to strengthen the chromosome structure of progeny generation. This approach is facilitated by defining a fitness function or a measure indicating the goodness of a member of the population in the given generation during the evolution process.

To represent the designs of mechanical equipments, the design variable is converted to strings of binary numbers appearing like chromosomes. Thus, the string is the equivalent of the design variable and thereby mapped into a fixed-length string of 0s and 1s. A number of such strings, i.e. individuals, constitute a population of designs with each design having a corresponding fitness value. The fitness of chromosomes is evaluated during each generation [**8**]. This fitness value could be the objective function $F(x)$ for a function maximization problem. Thus, the GA can be used to solve optimization problems of the form

$$\text{maximize } F(x) \text{ subject to } x_{\min} \leqslant x \leqslant x_{\max}$$
$$\text{where } x \text{ is the design variable for GA}$$

The individuals of the very first population are created in the domain of x lying between the minimum and maximum values. The sampling space is characterized by the size N_p of the population and the number N_o of offspring produced at each generation. The following genetic operators are applied to each successive population to improve results.

2.1 Crossover operation

In the two-point crossover approach, two genderless mating parents are selected at random by involving the random number generator, to identify the sites on the strings. The strings of 0s and 1s enclosed between the chosen sites are swapped between the mating strings. The number of crossover operations performed depends on crossover rate p_c.

2.2 Mutation operation

A few members from the population pool are taken depending on the mutation rate p_m. By switching 0 and 1 at randomly selected substrings on the chosen string, mutation is simulated. The chromosome generated by the crossover or mutation operation is called offspring.

2.3 Selection of the next generation

The procedure creates a new population from the present sample space. The space may consist of all parents and offspring or some other combination of parents and offspring [**9–13**]. The mechanism for sampling can be stochastic, deterministic, or mixed. In stochastic sampling, the number of copies of a chromosome in the next generation is based on its survival probability of fitness [**14**]. Holland's [**9**] proportionate selection or roulette wheel selection is one example of this type of sampling. Deterministic sampling selects the best N_p chromosomes from the sampling space. Truncation selection and block selection belong to this type of selection technique [**15**]. In another technique, N_o least-fit and old chromosomes are replaced by N_o offspring [**16**, **17**]. Mixed sampling contains both random and deterministic features. Tournament selection and remainder stochastic sampling [**18**, **19**] are examples of mixed sampling.

The process of crossover and mutation followed by reproduction in one generation produces the next generation of the GA. After several generations, the GA is stopped and the individual string with the highest fitness value is taken as the optimum. Since the GA is a probabilistic search method, it is very good at finding the global maximum. Furthermore, GAs need only function values and not gradient information, which makes them easy to use for real systems where accurate gradient information is difficult to obtain, and local minima may occur. However, they are computationally expensive.

3 CASE STUDY ON THE SELECTION OF THE BEST MECHANICAL TIMER

A case study of selecting the best timer for the four-cylinder truck engine, the details of which are given in Table 1, is described here. The problem is to select the optimum timer characteristic in such a way that the fuel consumption is low with a constraint on NO_x to satisfy production tolerances and the legislated norm.

3.1 Formulation of the problem

The timer path is the injection timing against the engine speed n, as shown in Fig. 1. The objective of this problem is to obtain an injection timer path that corresponds to the combination of best SFC, best power p, and emissions within a particular range. The input data are the SFC against n data, p against n data, and the emission test data at different fuel injection timings ω.

A typical injection-timing curve for a mechanical timer is shown in Fig. 1 representing a probable individual. Here, the injection timing at an engine speed of 1200 r/min is 12° before top dead centre (BTDC). This remains constant until the speed becomes equal to 1600 r/min. From 1600 to 2300 r/min, the injection timing increases linearly with increasing engine speed, after which the injection timing attains a value of 17° BTDC.

3.2 Representation

The above timer has to be represented as a binary string individual. The characterizing graph can be defined by using four points x_1, x_2, x_3, and x_4, which are called decision variables. These points are shown on the timer figure. As the first step, the decision variables are encoded as a binary string. The length of the string depends on the required precision and the domain of the decision variable (Table 2).

The precision of variable x_1 is two places after the decimal point. This requirement implies that the range of domain of variable x_1 should be divided into at least $(x_{1\,\max} - x_{1\,\min}) \times 100$ size ranges, where $x_{1\,\max}$ is the higher limit and $x_{1\,\min}$ is the lower limit of the range of domain for x_1. The required number m_1 of

Table 1 Engine details

Bore (mm)	104
Stroke (mm)	113
Rated speed (r/min)	2800
Maximum torque speed (r/min)	1800
Rated power (hp)	63
Number of cylinders	4

Table 2 Precision and range of domain of the individual in Fig. 1

Design variable	Required precision (places after the decimal point)	Range of domain
x_1, x_3	2	12–18°
x_2, x_4	0	1200–2800 r/min

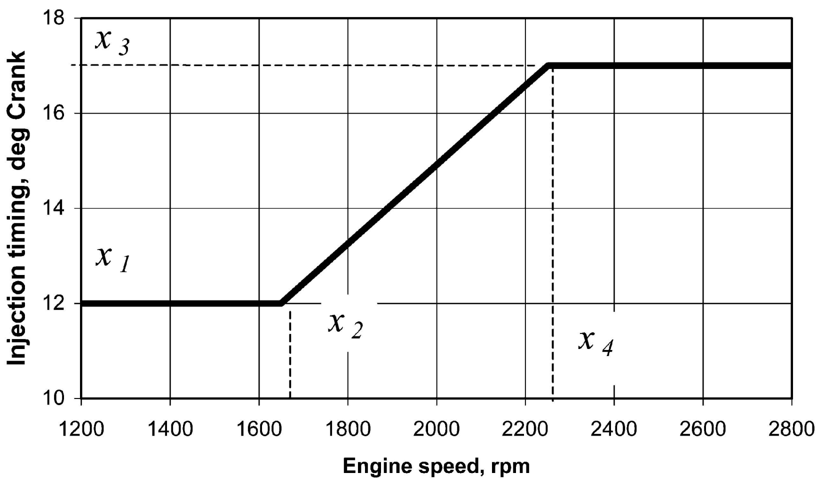

Fig. 1 Characteristic of an individual timer

bits is calculated as

$$2^{m_1-1} < (x_{1\,\max} - x_{1\,\min}) \times 100 \leqslant 2^{m_1-1}$$

m_2, m_3, and m_4 are found in similar ways.

Thus,

$$2^{10-1} < (18-12) \times 100 \leqslant 2^{10} - 1, \qquad m_1, m_3 = 10$$

$$2^{11-1} < 1600 \leqslant 2^{11} - 1, \qquad m_2, m_4 = 11$$

$$m_1 + m_2 + m_3 + m_4 = 42$$

Thus, the chromosome v_j representing the jth individual of the present population can be represented with a total length of 42 bits (see also Table 3) as

$$v_j = 011101100101000110101000101001010010100001$$

The mapping from the binary string to a real number for decision variable x_1 is

$$[x_1]_j = \left[x_{1-\min} + \text{decimal (substring}_1) \times \frac{x_{1-\max} - x_{1-\min}}{2^{m_1} - 1} \right]_j$$

$$j = 1, \ldots, N_\mathrm{p}$$

x_2 is calculated in a similar way. For decision variables x_3 and x_4 the condition that $x_3 > x_1$ and $x_4 > x_2$ should hold. Thus, the mapping is

$$[x_3]_j = \left[x_1 + \text{decimal (substring}_3) \times \frac{x_{3-\max} - x_1}{2^{m_3} - 1} \right]_j$$

$$[x_4]_j = \left[x_2 + \text{decimal (substring}_4) \times \frac{x_{4-\max} - x_2}{2^{m_4} - 1} \right]_j$$

$$j = 1, \ldots, N_\mathrm{p}$$

Here the decimal (substring$_1$) represents the decimal value of substring$_1$ for the decision variable x_1. Decimal values of x_2, x_3, and x_4 can be represented in similar ways.

Table 3 shows the values of x_1, x_2, x_3, and x_4 for v_j. This string is one particular solution from the solution space and corresponds to a particular injection timer. The procedure is discussed in the algorithm that follows.

Table 3 Example of construction of an individual or a chromosome

x	Substring	Decimal (substring)	Value of the decision variable x
x_1	0111011001	473	14.77
x_2	01000110101	565	1642
x_3	0001010010	82	15.25
x_4	10010100001	1185	2568

3.3 Objective function for describing fitness

The next step towards solving this problem is the definition of an objective function. An objective function is the parameter against which a particular individual or solution is compared with any other individual or solution. The objective function for this problem is defined as follows. The lower the value of this objective function for a solution, the better the solution will be. This objective function is calculated for all the individuals in the generation, N_p, as

$$[f_0]_j = \left[\frac{\sum_{i=1,\ldots,17} \mathrm{SFC}_i p_i w_i}{\sum_{i=1,\ldots,17} p_i w_i} \right]_j$$

$$j = 1, \ldots, N_\mathrm{p} \tag{1}$$

where

p_i = power
w_i = weight
i = selected point in the operating zone of the engine which is considered for the calculation of f_0
j = individual selected for fitness evaluation

3.4 Constraints

The constraint for this problem is to maintain the value of [NO_x] limited within a particular range. In practice, the upper limit is fixed by the legislated emission norm for NO_x. The limit should be just low enough to give production, with a sufficient engineering margin. Too low a lower limit jeopardizes the SFC and emission of smoke. To solve the two problems, the designer could end up with a costlier fuel injection system. There are many strategies to handle constraints with GAs such as rejecting, repairing, modifying genetic operators, and penalizing [**20**, **21**]. The strategy used for this problem is the penalising strategy, which has an advantage over the others in terms of searching both feasible and infeasible solution spaces. Glover and Greenberg [**22**] have suggested that constraint management techniques allowing movement through infeasible regions of search space tend to produce a better final solution than approaches limiting search trajectories to only feasible regions of search space. The penalty technique transforms the constrained problem to an unconstrained problem by penalizing infeasible solutions. This is achieved by adding a penalty term to the infeasible solutions for any violation of the constraints. The constraint for this problem is defined as

$$\text{if } [\mathrm{NO}_x] < 7 \text{ or } [\mathrm{NO}_x] > 7.5, \text{ penalty} = 0.0001$$
$$\text{else}$$
$$\text{penalty} = 1 \tag{2}$$

Here, the legislated norm for NO_x is 8 g/kW h, the engineering margin is 0.5, and the tolerance for production is 0.5 g/kW h.

A modified GA is used to minimize the objective function. For this purpose, the minimization problem is redefined as a maximization problem by defining a fitness function as

$$\text{fitness} = \frac{1}{f_0}\text{penalty} \tag{3}$$

A higher value of this fitness function for a solution will mean a better solution and vice versa.

3.5 Algorithm

After the solution representation, fitness function, and constraints for the problem are finalized, the algorithm is defined as follows.

3.6 Initial population

An initial population is created as a set of 100 individuals of randomly selected 42 bits. Thus, $N_p = 100$. These individuals are represented as v_j, $j = 1, \ldots, N_p$. The corresponding timers are t_j, $j = 1, \ldots, N_p$.

3.7 Evaluation steps

The process of evaluating the fitness of a chromosome consists of the following steps.

Step 1: convert the chromosome's genotype to its phenotype. The genotype is the encoded solution and the phenotype is the decoded solution. Here, this means converting binary string into relative real values (x_1, x_2, x_3, x_4).

Step 2: evaluate the objective function. 17 points are selected on the injection timing curve path for evaluation. The engine speeds at these points are

$$n_{ij} = [1200, 1300, 1400, \ldots, 2800]$$
$$i = 1, 2, \ldots, 17$$
$$j = 1, \ldots, N_p$$

where

$$n = \text{speed (r/min)}$$

The injection time at these speeds

$$(\omega_{ij}; i = 1, 2, \ldots, 17; j = 1, 2, \ldots, N_p)$$

is taken from the timer path curve. The values of SFC and power are taken from the performance maps available for different injection timings. Interpolation and extrapolation are carried out for the points that lie in between or outside the map range. 70 per cent more weight is given to the values that fall in the maximum used range of engine speeds (1800–2800 r/min). The weighted average as shown in equation (1) is taken as a measure of power and SFC. Thus

$$w_i = 0.3, \qquad i = 1, 2, \ldots, 6$$
$$w_i = 0.7, \qquad i = 6, 7, \ldots, 17$$

To estimate the value of $[NO_x]$, emission data at different injection timings are used. These tests are carried out according to the 13-mode emission test cycle. In this step, five emission readings are taken at an engine speed of 1800 r/min and corresponding ω, five emission readings are taken at an engine speed of 2800 r/min and corresponding ω, and three emission readings are taken at idle speed and corresponding ω.

Injection timing is found for speeds equal to the engine idle speed (minimum injection timing), 1800 r/min, and 2800 r/min from the timer path as per the individual in the population. The corresponding value of emission is found from the emission test data. Interpolation or extrapolation is carried out for points not within the test data. The value of the objective function f_0 and the penalty function for all the chromosomes in the population is evaluated using equations (1) and (2).

Step 3: evaluating fitness$_j$: $j = 1, 2, \ldots, N_p$. The fitness function is evaluated for all the chromosomes in the population using equation (3). The best generation individual is found.

Step 4: crossover. Two random individuals are selected and a crossover operation is performed on them to produce an offspring. The crossover used here is the one-cut-point method, which randomly finds the cut point and exchanges the right parts of two parents to generate an offspring. The crossover probability p_c is kept at 0.6.

Step 5: mutation. This step carries out the mutation operation. The mutation probability p_m is kept very low at 0.01.

Step 6: selection. In this step, the population is sorted in order of decreasing fitness. Deterministic sampling is used, in which the least-fit individuals of the population are replaced by the offspring from the crossover and mutation operation. For this, the population is ranked according to the fitness criterion and the least-fit N_0 individuals of the population are replaced with the offspring.

3.8 Next generation

The population after the previous steps consists of some parents and offspring. This generation is taken as the next generation. The fitness of the population is evaluated again and the individuals are again sorted out in decreasing order of their fitness as in steps 2 to 6. These steps are repeated until the termination criterion is met.

3.9 Termination criterion

The program termination criterion is given as a particular number of generation runs or as a particular value of fitness or some other desired parameter. The termination criterion for this case was the completion of a run consisting of 2500 generations. The individual with the best fitness in 2500 generations is taken as the solution.

4 RESULTS

In this study, a maximum of 2500 generations of the GA are used. The population size, crossover

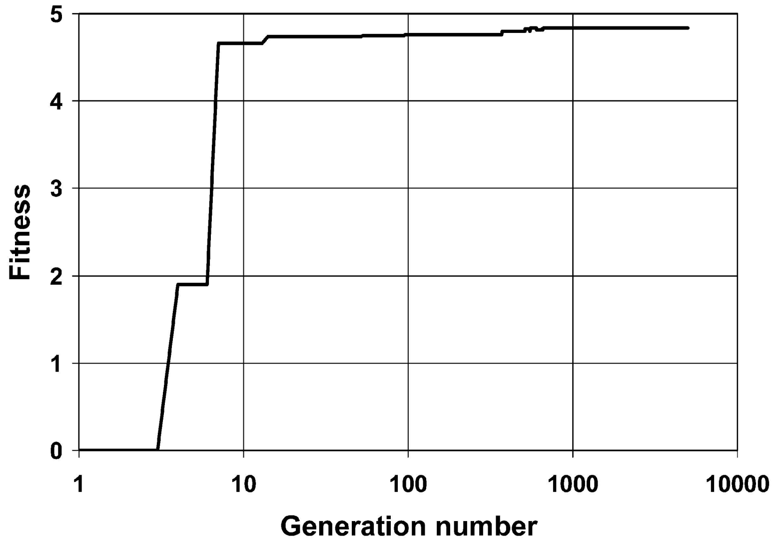

Fig. 2 Improvement in fitness of the population with advancing generations

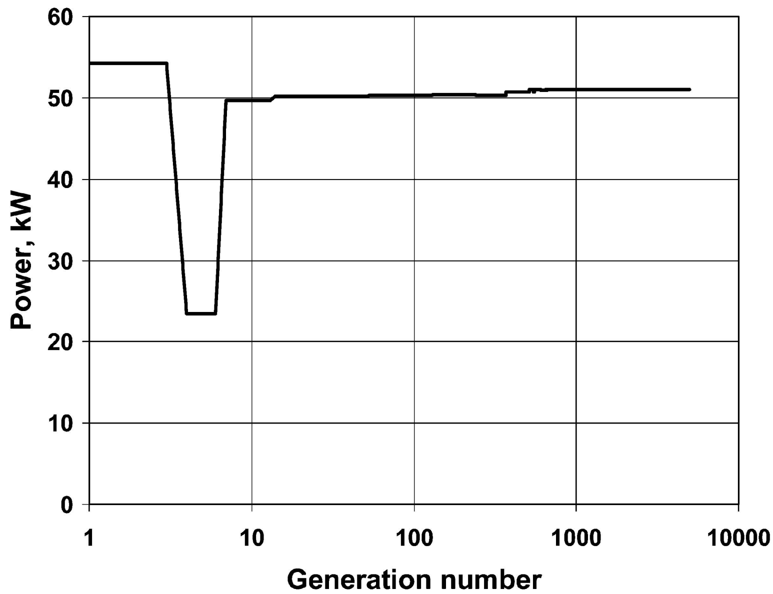

Fig. 3 Improvement in the engine power of the population with advancing generations

probability, and mutation probability are chosen as 50, 0.6, and 0.01 respectively. The code implementing the algorithm in this study takes about 3–5 min to run on MATLAB on a Pentium 4 personal computer with the full 2500 generations of the GA. The convergences occur in 1500–2000 generations. The initial solutions are found to be the worst. The best fitness in the first three generations is found to vary less because the [NO_x] values are greater than 7.5.

Figures 2 to 5 depict various aspects of the running of the GA. To demonstrate the stability of solutions, the graphs show results in excess of 2500 generations obtained by resetting the constraints in the study. The timescale is chosen to be logarithmic to emphasize the important episodes in the early generations of the method. The [NO_x] value is between 7 and 7.5 after the first three generations. The method seems to converge quickly in about eight generations. In

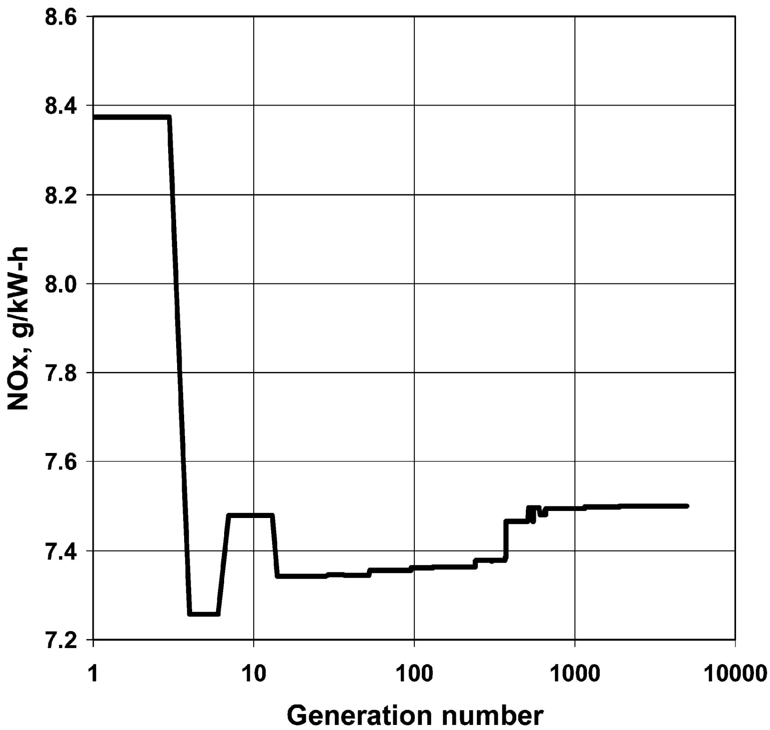

Fig. 4 Decrease in [NO_x] resulting from the population with advancing generations

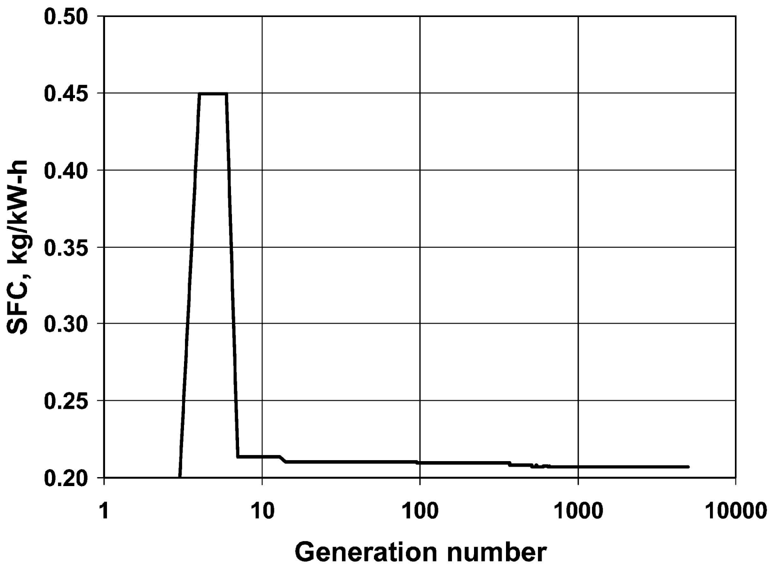

Fig. 5 Improvement in the SFC of the population with advancing generations

other words, the later generations that are fit already appear to be dull without any major achievement in the selection process. However, the mutation process produces fitter offspring in the population. The low probability of mutation assumed in the work is the reason for their rare occurrences as seen in individuals at about 12 and 300 generations. The best individual is found in the 1899th generation. Figures 3 to 5 show the improvement in the weighted average of the engine power, [NO_x], and SFC with advancing generation as the fitness of the population increases. Convergence earlier than 1899 is possible by increasing the probability of mutation to higher than 0.01. The weighted averages of the SFC, power and [NO_x] values corresponding to the best timer are 206.8 g/kW h, 51.1 kW, and 7.5 g/kW h respectvely. The timer selected by tedious experimental methods is very close to the timer achieved by the optimization using the GA method.

5 SUMMARY AND CONCLUSIONS

Similar to working with the DOE method, the difficulty with the GA strategy is to design it specifically for an application. However, since the focus is on timing optimization in the new computerized engines, a GA can be applied by developing suitable algorithms. The case of selecting the timer is an example of how a GA can be applied to complicated optimization of an engine. The only inputs to the system are the performance characteristics at various injection timings and the emission test results for [NO_x]. The program can interpolate or extrapolate to obtain the parameter values outside the input space. This paper shows that the technique can be extended to many other cases of engine parameters if the fitness criterion can be described.

REFERENCES

1 CAMEO software of AVL. http://tec.avl.com/wo/webobsession.servlet.go/encoded/YXBwPWtiYXNlJnBhZ2U9Y29udGVudC1tYW5hZ2VtZW50L3ZpZXcmaWQ9NDAwMDEzMjg0.html.
2 **Back, T.** *Evolutionary algorithms in theory and practice*, 1996 (Oxford University Press, Oxford).
3 **Schwefel, H.** *Numerical optimisation of computer models*, 1981 (John Wiley, Chichester, West Sussex).
4 **Back, T.** and **Schwefel, H.** *Evolutionary computation: an overview*, pp. 20–29.
5 **Michalewicz, Z.** *Evolutionary computation: practical issues*, pp. 30–39.
6 **Goldberg, D.** *Genetic algorithms in search, optimisation and machine learning*, 1989 (Addison-Wesley, Reading, Massachusetts).
7 **Hajela, P.** Non-gradient methods in multidisciplinary design optimisation – status and potential. *J. Aircr.*, 1999, **36**(1), 255–274.
8 **Fogel, D.** and **Ghozeli, A.** *Using fitness distributions to design more efficient evolutionary computations*, pp. 11–19.
9 **Holland, J.** *Adaptation in natural and artificial systems*, 1975 (University of Michigan Press, Ann Arbor, Michigan).
10 **De Jong, K.** *An analysis of the behavior of a class of genetic adaptive systems*, PhD Thesis, University of Michigan, Ann Arbor, Michigan, 1975.
11 **Grefenstette, J.** and **Barker, J.** How genetic algorithms work: a critical look at implicit parallelism. In Proceedings of the Third International Conference on *Genetic Algorithms* (Ed. J. Schaffer), 1989, pp. 20–27 (Morgan Kaufmann, San Mateo, California).
12 **Back, T.** Selective pressure in evolutionary algorithms: a characterisation of selection mechanisms. In Proceedings of the First IEEE Conference on *Evolutionary Computation* (Ed. D. Fogel), 1994, pp. 57–62 (IEEE Press, Orlando, Florida).
13 **Back, T.** and **Hoffmeister, F.** Extended selection mechanisms in genetic algorithms. In Proceedings of the Fourth International Conference on *Genetic Algorithms* (Eds Belew and Booker), 1991, pp. 92–99 (Morgan Kaufmann, San Mateo, California).
14 **Baker, J.** Adaptive selection methods for genetic algorithms. In Proceedings of the Second International Conference on *Genetic Algorithms* (Ed. Grefenstette), 1987, pp. 100–111 (Lawrence Erlbaum Associates, Hillsdale, New Jersey).
15 **Thierens, D.** and **Goldberg, D.** Convergence models of genetic algorithm selection schemes. In *Parallel Problem Solving from Nature* (PPSN III) (Eds Davidor *et al.*), 1994, pp. 119–129 (Springer, Berlin).
16 **Whitley, D.** GENITOR: a different genetic algorithm. In Proceedings of the Rocky Mountain Conference on *Artificial Intelligence*, Denver, Colorado, USA, 1989.
17 **Syswerda, G.** Uniform crossover in genetic algorithms. In Proceedings of the Third International Conference on *Genetic Algorithms* (Ed. J. Schaffer), 1989, pp. 2–9 (Morgan Kaufmann Publishers, San Mateo, California).
18 **Goldberg, D., Korb, B.,** and **Deb, K.** Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems*, 1989, **3**, 493–530.
19 **Brindle, A.** *Genetic algorithms for function optimisation*, PhD Thesis, University of Alberta, Edmonton, Alberta, 1981.
20 **Michalewicz, Z.** A survey of constraint handling techniques in evolutionary computation methods. In *Evolutionary Programming IV* (Ed. McDonnell *et al.*), 1995, pp. 135–155 (MIT Press, Cambridge, Massachusetts).

21 **Michalewicz, Z., Dasgupta, D., Le Riche, R. G.,** and **Schoenauer,** Evolutionary algorithms for industrial engineering problems. *Int. J. Computers Ind. Engng,* 1996, **30**(4).

22 **Glover, F.** and **Greenberg, H.** New approaches for heuristic search: a bilateral linkage with artificial intelligence. *Eur. J. Opl Res.*, 1989, **39**, 119–130.

APPENDIX

Notation

f_0	objective function
m	number of bits for a design variable
n	engine speed
N_o	number of offspring produced in each generation
N_p	size of the population
NO_x	nitrogen oxides
p	engine power
p_c	rate of crossover
p_m	rate of mutation
SFC	specific fuel consumption
t	injection timer
w	weightage factor
x	design variable
v	chromosome or an individual
ω	injection time

Subscripts

i	selected points for injection timer evaluation in the engine map operating zone
j	individual number in the population, ranges from 1 to the size of the population