APPLICATION OF GENETIC ALGORITHMS FOR ROBUST PARAMETER OPTIMIZATION

N. Belavendram

Department of Mechanical and Materials Engineering
Faculty of Engineering and Science, Universiti Tunku Abdul Rahman
Jalan Genting Kelang, 53300 Setapak, Kuala Lumpur, Malaysia.
Email: nicolob@utar.edu.my
Phone: +603-4107 9802, Fax: +603 4107 9803

ABSTRACT

Parameter optimization can be achieved by many methods such as Monte-Carlo, full, and fractional factorial designs. Genetic algorithms (GA) are fairly recent in this respect but afford a novel method of parameter optimization. In GA, there is an initial pool of individuals each with its own specific phenotypic trait expressed as a 'genetic chromosome'. Different genes enable individuals with different fitness levels to reproduce according to natural reproductive gene theory. This reproduction is established in terms of selection, crossover and mutation of reproducing genes. The resulting child generation of individuals has a better fitness level akin to natural selection, namely evolution. Populations evolve towards the fittest individuals. Such a mechanism has a parallel application in parameter optimization. Factors in a parameter design can be expressed as a genetic analogue in a pool of sub-optimal random solutions. Allowing this pool of sub-optimal solutions to evolve over several generations produces fitter generations converging to a pre-defined engineering optimum. In this paper, a genetic algorithm is used to study a seven factor non-linear equation for a Wheatstone bridge as the equation to be optimized. A comparison of the full factorial design against a GA method shows that the GA method is about 1200 times faster in finding a comparable solution.

Keywords: parameter, optimization, genetic, algorithms, simulation

INTRODUCTION

Parameter optimization is an important field of engineering that seeks to find some optimum (i.e., minimum or maximum) point in a design space (Phadke, 1989). Classical methods for such optimization include the Monte Carlo and the full or fractional factorial methods. More recently several new methods have emerged including the genetic algorithms, the particle swarm optimization and ant colony optimization. In this paper, the author compares the more ¬intensive full factorial method against the genetic algorithm method using the Wheatstone bridge equation as a function to be optimized. Consider the Wheatstone bridge circuit as shown in Figure 1. The bridge circuit is used to identify the value of the unknown resistor y. Mathematically,

$$y = \frac{BD}{C} - \frac{G(AD + AC + BD)(BC + BD + BF + CF)}{C^2 E}$$

(1)

where A, B, C, D and F are resistors, E is the applied voltage and G is the galvanometer (Taguchi, 1987). Given nominal values of variables, A, … , G, there are infinitely many combinations of values which will give a target value τ of y. The objective of parameter design is to find not only the given target value of y, say $\tau = 2\ \Omega$, but to do so in a region of space where small variations in the variables (e.g., resistors, voltage, galvanometer readings) and other noise variations (e.g., humidity, temperature) do not affect the measured value y.
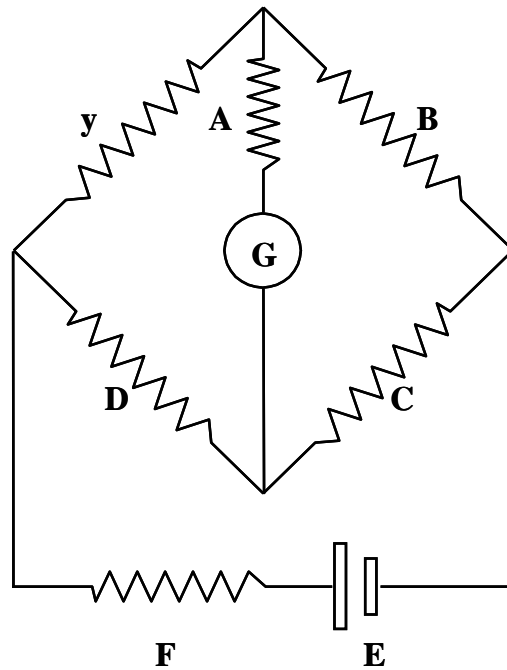

Figure 1. Wheatstone bridge circuit.

Let ϕ_i for i = 1, … , 7 represent, respectively, the variables A, …, G. Then it is necessary to find values of ϕ_i such that $y = \tau$. Hence, the measure of y is a target performance measure, or TPM (Logothetis, 1989). Further, the variance associated with y for ϕ_i can be estimated by the change $\Delta y = \dfrac{\delta y}{\delta \phi_i} \Delta \phi_i$. Since the variance $Var(\phi_1 + \phi_2 + \ldots + \phi_k) = Var(\phi_1) + Var(\phi_2) + \ldots + Var(\phi_k)$, the variance σ^2 of all the variables can be estimated by Eq. 2, (Logothetis, 1992).

$$\sigma^2 = \sum_{i=1}^{7} \frac{\delta y}{\delta \phi_i} \Delta \phi_i \qquad (2)$$

However, choosing to evaluate the variance directly is not a good method since variance is not a scale invariant quantity. Therefore, the quantity y/σ, i.e. the inverse coefficient of variation can be used. This however, distinguishes $-y$ from $+y$ which is not pertinent to the present case. Hence, one can use the function $\eta^* = (y/\sigma)^2$ as a noise performance measure or NPM. This measure can be further improved by using a logarithmic function. Suppose, ϕ is confounded with some unknown noise effect ω,

and $y = \omega\phi$, then taking logarithms, $\log y = \log\omega + \log\phi$. The effect of ω is then reduced to a constant which can be ignored in differences of $\log y$. Therefore, one can use the noise performance measure or NPM as shown in Eq. 3 to improve additivity (Logothetis, 1989).

$$\eta = 10\log\left(y/\sigma\right)^2 \tag{3}$$

Referring to

Figure **2**, the optimization problem is now Step 1, to use NPM to minimize the variance of a factor; and Step 2, to use TPM to minimize the bias, i.e. $\left(\bar{y} - \tau\right) \approx 0$ and achieve the target value as best as possible. Such a method of optimization was first proposed by Taguchi as the two step optimization and referred to as robust design (McEwan et. al., 1992).
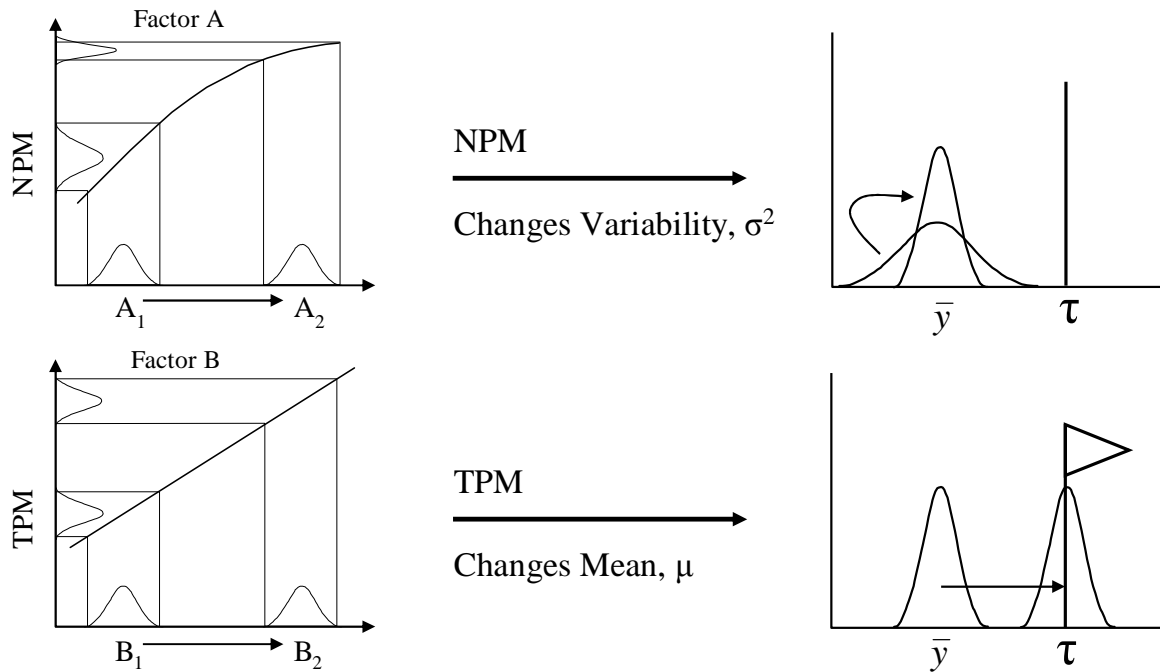

Figure 2. Effect of NPM and TPM on response.

OPTIMIZATION TECHNIQUES

Full Factorial Method

In a full factorial method, one can evaluate TPM = y. Eq. 1 can be partial differentiated with respect to each variable, e.g. for factor A,

$$\frac{\partial y}{\partial A} = \frac{-G(D+C)(BC+BD+BF+CF)}{C^2 E} \tag{4}$$

If the tolerance about the nominal value of A is known or set as ΔA then, $\Delta y = \dfrac{\delta y}{\delta A}\Delta A$. The variance of y for a given combination of variables ϕ_i can then be found from $\sigma^2 = \sum_{i=1}^{7}\dfrac{\delta y}{\delta \phi_i}\Delta \phi_i$ and $\mathrm{NPM} = \eta = 10\log\left(\dfrac{y}{\sigma}\right)^2$.

Table 1. Parameter values for factors.

Factors	A	B	C	D	E	F	G
Min value	20	1.99	2	2	1.2	2	-0.0002
Max value	500	2.01	50	50	30.0	50	+0.0002
Tolerance	0.003	0.003	0.003	0.003	0.050	0.003	0.0002
Unit	ohm	ohm	ohm	ohm	volt	ohm	ampere
Optimum	1	3	2	2	7	1	1

Using $j = 1, \ldots, k$ levels for each factor, the factor values are set according to $\phi_{ij} = \phi_{i:\min} + (j-1)\left(\dfrac{\phi_{i:\max} - \phi_{i:\min}}{k}\right)$.

A full factorial is initiated with all factor levels at level 1 and then incrementing factor G's level by 1 until k. Then factor F's level 1 is incremented by 1 for k levels of G. The process is repeated for factors E, D, C, B and so on until all k levels of A are completed. The TPM and NPM values were calculated for $8^7 = 2{,}097{,}152$ combinations of factor levels using a simple Excel program. Partial results are shown in Figure 3.

	A	B	C	D	E	F	G	Tolerances							Statistics		
#	1	2	3	4	5	6	7	ΔA	ΔB	ΔC	ΔD	ΔE	ΔF	ΔG	y	σ^2	η
1	1	1	1	1	1	1	1	0.00	0.01	0.01	0.01	0.00	0.00	0.06	2.05253	0.0035	30.75
2	1	1	1	1	1	1	2	0.00	0.01	0.01	0.01	0.00	0.00	0.06	2.03580	0.0035	30.68
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
2097152	8	8	8	8	8	8	8	0.00	0.01	0.03	0.01	0.02	0.00	0.39	1.61312	0.1559	12.23

Figure 3. Partial results of full factorial experiment.

Every time a TPM value is within $2 \pm 0.00005\ \Omega$, the TPM and NPM values are noted and recorded by the program. The factor combination with the highest NPM (i.e. 1, 3, 2, 2, 7, 1, 1) is then taken as the optimum factor level combination. The corresponding optimum values are $y = 1.999\,961\ \Omega$ and $\eta = 45.447\ \mathrm{dB}$. Noise factors built around the nominal $\pm 1 \times$tolerance gives a typical graph as shown in Figure 4. The reduction in variation is dramatic (110 times) while also achieving the target value of $2.0\ \Omega$.
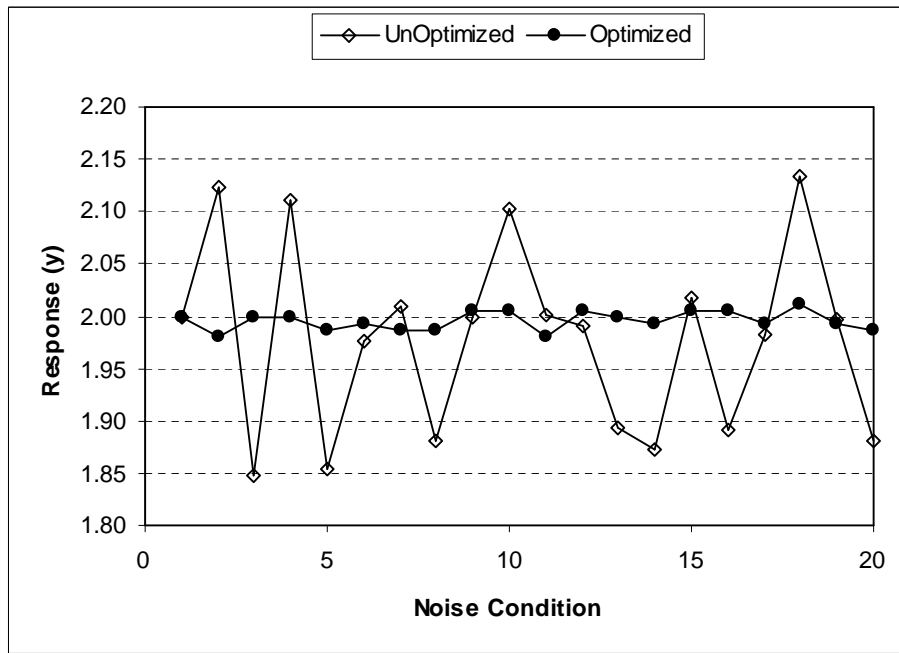
Figure 4. Comparison of unoptimized–optimized responses from full factorial.

Genetic Algorithms

The GA method of optimization begins with the expression of engineering factors in binary code although other codes are also possible (Coley, 1999). Consistent with the full factorial, the range of factor values j to be tried is set up with $k = 8$ values according to the binary levels 000, 001, 010, 011, 100, 101, 110, 111. This corresponds to a gene length of 3 and chromosome length of 21 binary digits as shown in
Figure 5.

For a combination, $F(A_{000}; B_{010}; \ldots, G_{111})$ corresponding to $F(A_1, B_2, \ldots, G_8)$, the gene representation is translated to the factor value and substituted into Eq. (1) to calculate y. Using the same factor levels, the variance is calculated by substitution of the factor values into $\sigma^2 = \sum_{i=1}^{7} \frac{\delta y}{\delta \phi_i} \Delta \phi_i$ (Phadke, 1989).

The NPM is calculated as $\eta = 10 \log \left(\frac{y}{\sigma} \right)^2$ and lastly, the fitness function F is calculated as shown Figure 5.

#	Factor A			Factor B				Factor G			Tolerances							Statistics			
	1	2	3	4	5	6	...	19	20	21	ΔA	ΔB	ΔC	ΔD	ΔE	ΔF	ΔG	y	σ^2	η	F
1	0	1	1	0	1	0	...	0	1	0	0.00	0.01	0.01	0.01	0.00	0.00	0.03	2.00128	0.0008	36.82	0.010
2	0	1	1	0	0	0	...	0	1	0	0.00	0.01	0.01	0.01	0.00	0.00	0.03	1.99874	0.0012	35.20	0.012
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10	0	1	1	0	1	0	...	0	1	0	0.00	0.01	0.01	0.01	0.00	0.00	0.16	2.02031	0.0258	22.00	0.467
Opt	0	1	1	0	1	1	...	0	1	0	0.00	0.01	0.01	0.01	0.00	0.00	0.03	1.99874	0.0012	35.20	0.012

Figure 5. Example of calculations with genetic algorithm.

Fitness function

For parameter optimization, the selection of the fitness function poses some difficulty since we require to achieve a specified value of τ and η. Attaining a specified target value with the smallest variability implies that both the mean and variance need to be considered. Consistent with the full factorial, if we set a target of $y = 2.0\ \Omega$ and $\eta = 45$ dB, it is possible to optimize $f = (y-2)(\eta-45)$ but the resulting surface response is a saddle as shown in
Figure **6**A. Clearly, it is not possible to find a suitable minimum on a saddle.

However, the function $F = (y-2)^2(\eta-45)^2$ produces a surface response as shown in
Figure 6B. Although the region of interest is fairly flat (which is a good thing to have) it is possible to determine the minima. In this paper, this fitness function F is used.
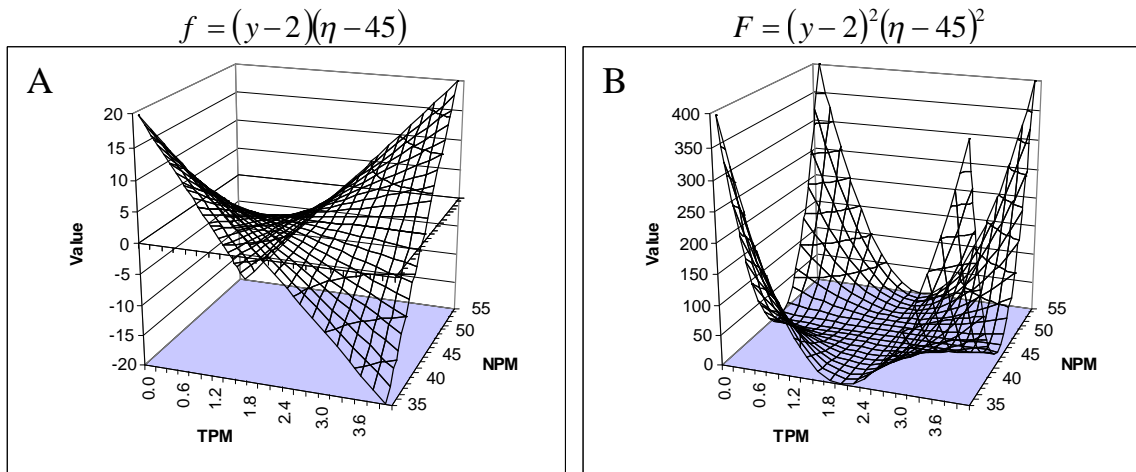
Figure 6. Response surfaces of fitness functions.

Reproduction

In the GA method, a random sample of 10 individuals was set up and the fitness function was evaluated corresponding to Generation 0. Using this parent generation, individuals for the next generation (Generation 1) were selected (by roulette wheel, tournament, etc.), crossed over (with probability p and position x along the chromosome) and mutated (with probability q and position z along the chromosome) to form the next generation as shown in
Figure **7**, (Haupt and Werner, 2007; Haupt and Haupt, 2004). The fitness function for each chromosome is evaluated and used in subsequent selection processes. This reproduction is continued until a specific criterion (e.g. number of generations, tolerance of target value, etc.) is reached. At this point, the optimum gene pool is noted. Since a GA can produce different solutions at different times it is suggested to run several searches.
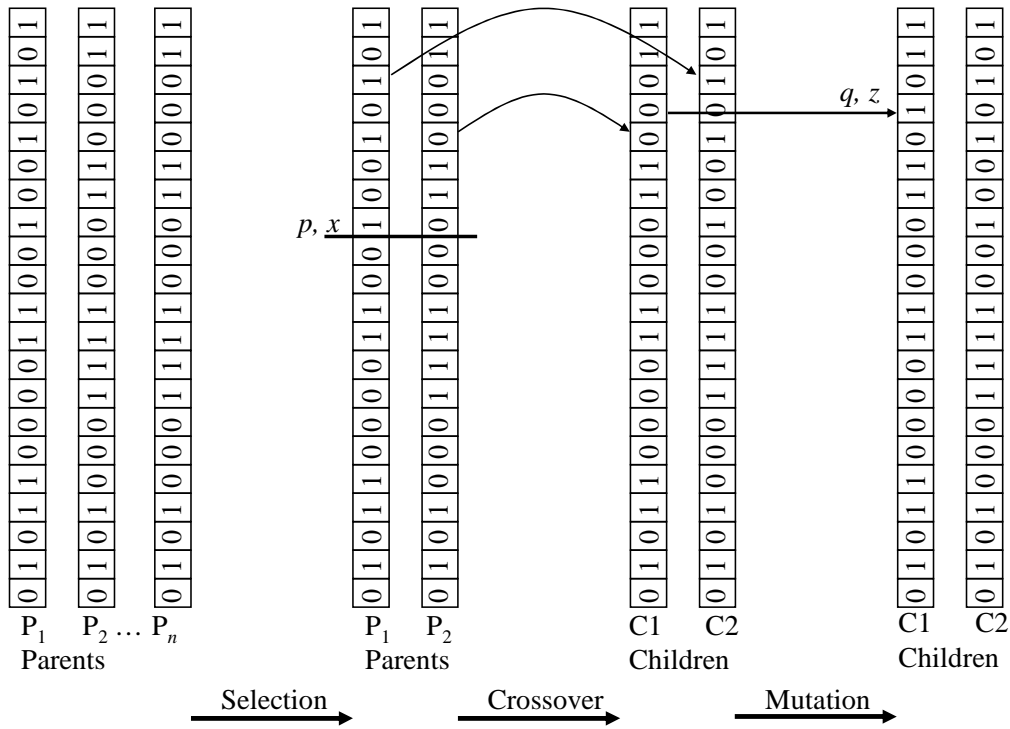
Figure 7. Genetic algorithm method

RESULTS AND DISCUSSION

Tracing the fitness function of a random population over 20 generations (
Figure 8) showed that the population converges rapidly to a solution in an amazing time of less than 3.0 seconds.
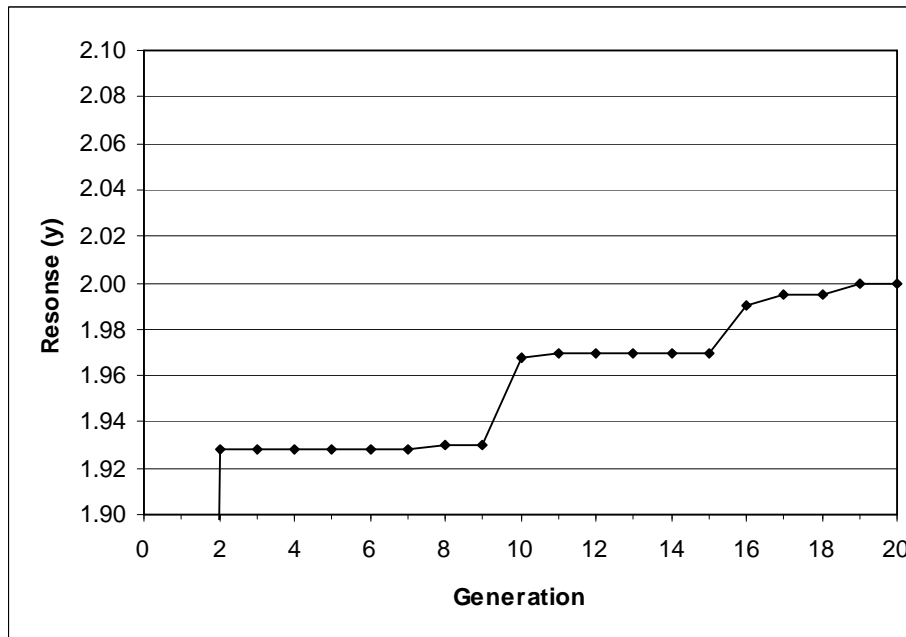

Figure 8. Twenty generations of GA reproduction.

Figure 9 compares the optimum solution to the unoptimized solution for the genetic algorithm with a typical variance reduction in the breakthrough region of 150 times. As in the full factorial, the reduction in variation is dramatically obvious while achieving the target value of 2.0 Ω.
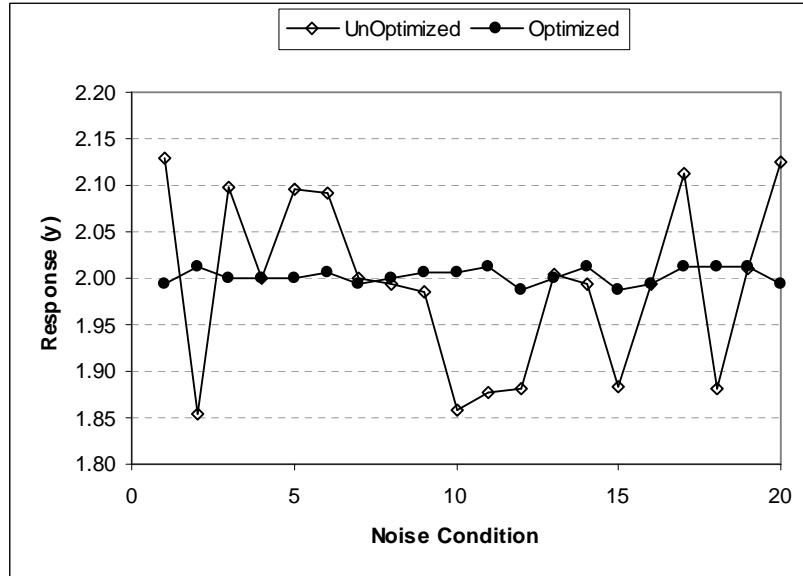

Figure 9. Comparison of unoptimized–optimized responses from genetic algorithms.

Comparison of Full Factorial and Genetic Algorithm

It now remains to compare the full factorial method with the GA method. To do this, optimum conditions from each method were exposed to similar noise conditions (i.e. nominal ± 1×tolerance). The result of such a comparison is shown in Figure 10. It is clearly seen that the results are comparable both in terms of target value and variance.
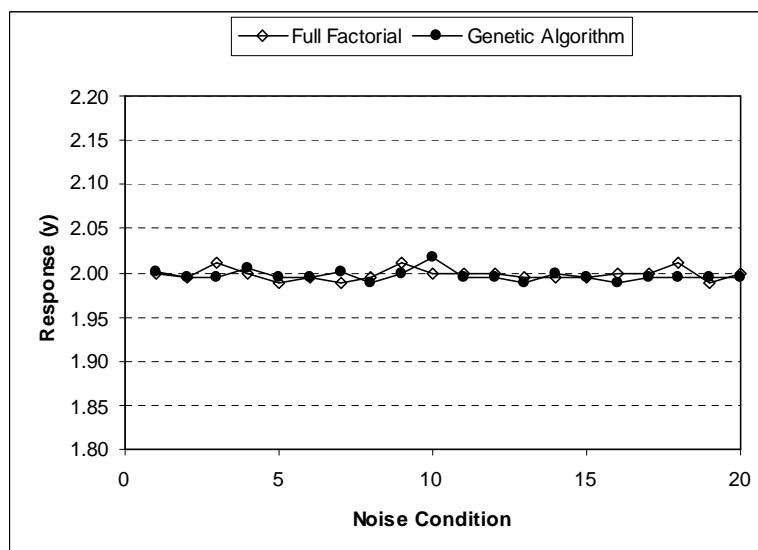

Figure 10. Comparison of full factorial and genetic algorithm.

Approaching the parameter optimization as a full factorial experiment for seven variables, each at eight levels, results in 2,097,152 permutations. This took 3657 seconds (61 minutes or approximately 1 hour) on a Pentium IV, 2.20 GHz processor with 1 GB of RAM. The genetic algorithm method took only 3 seconds to arrive at a comparable solution on the same computer. That is, the time taken was reduced by a factor of 1219. Although full factorial methods provide highly repeatable results they are very time consuming. If an eighth factor was introduced into Eq. 1, the computation time will increase linearly by eight times, i.e. to eight hours. However, genetic algorithms provide a radically different approach to finding the solution. Even if a linear increase was expected, this would only increase the time to 24 seconds. Another drawback of the full factorial is the grid size of factor-level values. If a smaller grid size is used the experimentation time would increase exponentially. Thus, large multi-level factors would become impractical. If however, the grid size is decreased to fewer levels, it is possible that a global optimum is missed. Such a problem is avoided in genetic algorithms by the use of the selection, crossover and mutation processes that provides a superior method of escaping local optima.

CONCLUSION

From the foregoing research, genetic algorithms provides a comparable solution to the Full Factorial design. Genetic algorithms achieved the optimization in 3 seconds while the full factorial took 3657 seconds. Genetic algorithms starts with a population of guesses distributed randomly throughout the search space rather than a single point. Genetic algorithms then uses selection, crossover and mutation to generate new solutions that are closer to the fitness function. Newer solutions evolve within the population towards a convergence to the global optimum.

FUTURE WORK

This paper studied the optimization of a function with a point target. For the Wheatstone bridge circuit, the target was set to 2.0 Ω. But would this 'optimum condition' also hold for a target value of 4.0 Ω? Consider another example in the optimization of a car braking system for an average speed of say 80 km/hr. Such optimization is unlikely to be ideal for all speeds. Thus, braking may be abrupt at low speeds and dragging at high speeds. To optimize such a system, a range of speeds (signal levels) must be optimized. The author is currently developing genetic algorithms for use in such applications where the optimization is the gradient of a regression line.

ACKNOWLEDGEMENTS

REFERENCES

Belavendram, N. (1995) *Quality by Design, Taguchi Techniques for Industrial Experimentation*. Hertfordshire, UK: Prentice-Hall.

Coley, D. (1999) *An Introduction to Genetic Algorithms for Scientists and Engineers.* London: World Scientific.

Haupt, R. L. and Haupt, S.E. (2004) *Practical Genetic Algorithms.* 2 ed. Hoboken, NJ: Wiley.

Haupt, R. L. and Werner, D. (2007) *Genetic Algorithms in Electromagnetics*. Hoboken, NJ: Wiley.

Logothetis, N. (1992) *Managing for Total Quality; From Deming to Taguchi and SPC*. Hertfordshire, UK: Prentice-Hall.

Logothetis, N. and Wynn, H. P. (1989) *Quality Through Design—Experimental Design, Off-Line Quality Control and Taguchi's Contributions*. Oxford.

McEwan, W., Belavendram, N. and Abou-Ali, M. (1992) Improving Quality Through Robustisation, *Journal of the Institute of Quality Assurance*, 18(2): 56–61.

Phadke, M. S. (1989) *Quality Engineering Using Robust Design.* Hertfordshire, UK: Prentice-Hall.

Taguchi, G. (1987) *System of Experimental Design*. New York: Kraus.