# A DECISION SUPPORT METHODOLOGY FOR PROCESS IN THE LOOP OPTIMISATION

**Dan Gladwin [(a)], Paul Stewart [(b)], Jill Stewart [(c)], Rui Chen [(d)], Edward Winward [(e)]**

[(a)(b)(c)] Department of Electronic and Electrical Engineering, University of Sheffield, UK
[(d)(e)] Department of Aeronautical and Automotive Engineering, University of Loughborough, UK

[(a)]d.gladwin@shef.ac.uk, [(b)]p.stewart@shef.ac.uk, [(c)]jill.stewart@shef.ac.uk, [(d)]r.chen@lboro.ac.uk,
[(e)]r.e.t.b.winward@lboro.ac.uk

**ABSTRACT**
Experimental optimisation with hardware-in-the-loop is a common procedure in engineering, particularly in cases where accurate modelling is not possible. A common methodology to support experimental search is to use one of the many gradient descent methods. However, even sophisticated and proven methodologies such as Simulated Annealing (SA) can be significantly challenged in the presence of significant noise. This paper introduces a decision support methodology based upon Response Surfaces (RS), which supplements experimental management based on variable neighbourhood search, and is shown to be highly effective in directing experiments in the presence of significant signal to noise (S-N) ratio and complex combinatorial functions. The methodology is developed on a 3-dimensional surface with multiple local-minima and large basin of attraction, and high S-N ratio. Finally, the method is applied to a real-life automotive experimental application.

Keywords: Experimental decision support, variable neighbourhood search, gradient descent, Simulated Annealing

## 1. INTRODUCTION

Experimental search with hardware or process in-the-loop is a common procedure in, for example, the engineering and pharmaceutical industries. These procedures are often applied to combinatorial problems during (in the engineering case for example) the development of new hardware systems or control (Stewart, Fleming and MacKenzie 2003, Stewart, Stone and Fleming 2004). The systems under test, can be described as a set of dependent variables which vary according to some functions of independent variables. In this case, there doesn't exist a complete specification of the function relating the variables. This implies that there is no accurate *a-priori* knowledge of the fundamental cause and effect present in the system. Thus, for an example linear function, the values in the co-efficients matrix would be unknown.

Commonly it is required to identify the sets of independent variables which maximise or minimise the dependent variables (Myers and Montgomery 1995). To obtain the necessary information to have a confident estimate of the parameters, it is possible to vary the independent parameters over successive trials (Designed Experiments) and measure the corresponding dependent variables. In order to fully examine this relationship, a large number of trials is often required to identify the location of the desired response. However, real-world problems are difficult to solve by this methodology for a number of reasons (Michalewicz and Fogel 2000):

- The number of possible solutions in the experimental space is so large as to preclude an exhaustive search for the best (or acceptable) and answer (Michalewicz and Fogel 2000).
- The evaluation function that describes the solutions is extremely noisy and/or complex.
- The cost of conducting an experiment at large numbers of points in the search space may be prohibitive in terms of time taken and/or resources used.

These constraints motivate the use of gradient descent methods in order to provide the decision support to direct the search and minimise the number of experiments conducted. Other meta-heuristics such as Genetic Algorithms (Narayana Naik, Gopalakrishnan and Ganguli, 2008) are applicable to this class of problem, but are relatively difficult to implement and tune due to the number of parameters associated with this technique as compared to gradient descent methods. These methods are based upon the statistics of the neighbourhood around a given point, thus relying on local information at each step. However, basic gradient descent methods only provide locally optimum solutions whose values depend on selection of the starting point (Mladenovic and Hansen, 1997). There have been many meta-heuristic methods developed to increase the efficiency of the experimental search, such as simulated annealing (Burke and Kendal 1999), tabu search (Glover and Hanfi 2002), genetic algorithms (Stewart, Stone and Fleming 2004) and variable neighbourhood search (Toksari and Guner 2007).

As the nature of the experimental surface is an unknown, it is important to utilise methodologies which require the minimum number of parameters to be 'tuned' in order to conduct effective search. With this *caveat* in mind, simple gradient descent, simulated annealing (SA) and variable neighbourhood search (VNS) will be considered in this paper, since in most of their varieties, implementation is simple, and basic tuning rules are readily available.

It has been noted in the literature that the performance of meta-heuristics such as simulated annealing are to some extent compromised when directing search over significantly noisy surfaces (Michalewicz Fogel 2000). This paper describes the implementation of a *weighted stochastic decision support operator* based on RS which guides experimental process to predicted areas of interest in the search space. Basic Gradient Descent, SA and VNS are supplemented by this methodology, and performance is compared to the basic form of the meta-heuristics. The supplemented meta-heuristics are shown to have significantly improved performance when searching noisy environments.

### 1.1. Scope of this paper

What this paper is about:

- Experimentation on real engineering problems
- Inherently noisy data
- Decision support to reduce experimentation
- Applying a decision support operator to common search methodologies

What this paper is not about:

- Performance comparison of heuristics and meta-heuristics
- Meta-heuristic tuning methodologies
- 'Toy' problems and surfaces

This paper concerns itself with the problem of finding a result in an unseen, noisy search space, where every individual evaluation of a point in the search space is expensive in terms of time and/or resources.

## 2. SEARCH METHODOLOGIES

A comprehensive description of gradient descent based methods can be found in (Michalewicz and Fogel, 2000). In this section of the paper, the implementations of the algorithms are described. The class of problems addressed in this paper are in general of a *minimisation* type (Baldick 2006). That is, it is desired to minimise a function $f(x)$ over choices of $x$ which lie in the feasible set $S$ such that;

$$f^* = \min_{x \in s} f(x) \tag{1}$$

thus; $\exists x^* \in S$, such that: $f^* = f(x^*)$ and $x \in S \Rightarrow f(x^*) \leq f(x)$. Set $S$ is the constraint set, $f^*$ is the minimum of the problem, while $x^*$ is the minimiser. We will refer to minimisation problems in this paper, however a conversion to maximisation is a trivial task.

### 2.1. Gradient Descent

A sequence of intermediate values are successively generated by the algorithm. We begin with an initial random guess and successively improve it. In general, none of the iterates exactly solve the problem, therefore we include a termination criteria which when satisfied will cause the algorithm to terminate with a suitable approximation to the exact solution. This is particularly applicable to real-world, noisy surfaces. Iterative hill descent can be described with the general form of recursion;

$$x^{v+1} = x^v + \alpha^v \Delta x^v, v = 0,1,2... \tag{2}$$

where;

- $x^0$ is the initial guess
- $v$ is the iteration counter
- $x^v$ is the value of the iterate at the $vth$ iteration
- $\alpha^v \in \Re_+$ is the step size
- $\Delta x^v \in \Re^n$ is the step direction
- $\alpha^v \Delta x^v$ is the update to add to the current iterate $x^v$ to obtain new iterate $x^{v+1}$

In the case of minimisation, the step direction is chosen as to reduce the objective $f^*$. If we let $\hat{x} \in \Re$, then the vector $\Delta x \in \Re$ is called a descent direction for $f$ at $\hat{x}$ if $\exists \overline{\alpha} \Re_{++}$ such that;

$$0 < \alpha \leq \overline{\alpha} \Rightarrow f(\hat{x} + \alpha \Delta x) < f(\hat{x}), \tag{3}$$

and $\Delta x$ is a descent direction for $f$ at $\hat{x}$ if the objective is smaller than $f(\hat{x})$ at points along the line segment $\hat{x} + \alpha \Delta x$ for $\alpha > 0$ and $\alpha \leq \overline{\alpha}$. There are some *caveats* associated with gradient descent methods:

- The methods usually terminate at solutions which are only locally optimal
- No information is apparent as to how the discovered local optima deviates from the global minima or other local minima
- The optimum obtained depends on the original configuration
- In general it is not possible to calculate an upper bound for computation time

Gradient Descent thus exploits the best opportunities for improvements, but neglects to explore a large search space. In contrast, random search where points are sampled from $S$ with equal probability explores thoroughly, but forgoes local exploitation. Thus most gradient descent methods execute a random 'jump' at local minima, to balance exploration with exploitation

### 2.2. Variable Neighbourhood Search

The VNS algorithm implemented in this paper systematically exploits the idea of neighbourhood change in the descent to minima (Burke and Kendall 2005), and attempts to balance local exploitation with global exploration. It is simply an implementation of the basic gradient descent method described in the previous section, however, in this case the step length $\alpha^v$ is variable rather than fixed. A number of variations have been reported, with both lengthening step length (Mladenovic and Hansen 1997) and reducing step length (Toksari and Guner 2006). In this case, reducing step length is implemented, by a static schedule (Kirkpatrick 1983), using a step decementation function given by;

$$C_{k+1} = \alpha \cdot c_k, k = 0,1,2... \tag{4}$$

The initial step length is usually chosen to be significant with respect to the search variable ranges, where $\alpha$ is chosen to be a positive constant > 1. The final value is fixed, generally related to the smallest feasible measurement or control increment of the variables. As with gradient descent, a random 'jump' is implemented to escape local minima.

### 2.3. Simulated Annealing

The implementation of Simulated Annealing used in this paper is based again on gradient descent, accepting improvements in cost in traversing the search space, however depending on a control parameter $c$ it will accept deteriorations to a limited extent to escape local minima. Initially, at large values of $c$, large deteriorations will be accepted, as $c$ decreases, smaller deteriorations are accepted, and finally, as the value of $c$ approaches 0, no deteriorations are accepted. The probability of accepting deteriorations is achieved by comparing the value of $\exp((f(i) - f(j))/c)$ with a random number generated from a uniform distribution on the interval $[0,1]$. In this case, the rate of decrease of the parameter $c$ is achieved by implementing the VNS decrementation function.

## 3. WEIGHTED STOCHASTIC DECISION SUPPORT OPERATOR (WSDS)

Local search methods such as gradient descent execute a random jump at local optima or other pre-defined termination metric based upon the implementation. SA-type methodologies typically execute a random jump at the termination of the cooling schedule if the global minimum or some upper bound of acceptable performance hasn't been reached. Obviously with unknown experimental functions, the exact value of the global maximum will not be known, however it is common for the designer/experimenter to have an "acceptable" performance metric in mind when starting the experimental procedure which will act as a termination criteria.

Tabu Search has been shown to be a particularly effective meta-heuristic by directing the experimental search 'jumps' away from areas which have been found to be unproductive. However, this doesn't take advantage of the previously gathered data with respect to the possibility of 'predicting' promising areas of search. The RS methodology has been shown to be an effective tool in approximating complex and noisy functions for real time control (Stewart and Fleming 2002, 2004), and thus would appear to be a useful tool to direct experimentation based upon past results.

The response surface methodology is a technique which was initially developed to optimise process control and experimentation by the application of designed experiments in order to characterise the relationship between the system variables and outputs (Myers and Montgomery 1995). The relationship between the response variable of interest $y$, and the predictor variables $(\xi_1, \xi_2, ..., \xi_k)$ comprise a description of the system of the form

$$y = g(\xi_1, \xi_2, ..., \xi_k) + \varepsilon \tag{5}$$

where $\varepsilon$ represents the model error, and includes measurement error, and other variability such as background noise. The error will be assumed to have a normal distribution with zero mean and variance $\sigma^2$. In general, the experimenter approximates the system function $g$ with an empirical model of the form

$$y = f(\xi_1, \xi_2, ..., \xi_k) + \varepsilon \tag{6}$$

where $f$ is a polynomial of arbitrary order (generally first or second order in the process control industry). This is the empirical or response surface model. The variables are known as *natural variables* since they are expressed in physical units of measurement. The natural variables are transformed into *coded variables* $x_1, x_2, ...x_k$ which are dimensionless, zero mean, and the same standard deviation. The response function now becomes

$$\eta = f(x_1, x_2, ...x_k) \tag{7}$$

The successful application of RS relies on the identification of a suitable approximation for $f$. This will often be a first order model of the form

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k \tag{8}$$

or a second order model of the form

$$\eta = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k} \beta_{jj} x^2{}_j + \sum_{i<j} \sum \beta_{ij} x_i x_j \tag{9}$$

It may also be necessary to employ an approximating function greater than an order of two, based on the standard Taylor series expansion. The set of parameters can be estimated by regression analysis based upon the experimental data. The method of least squares is typically used to estimate the regression coefficients. With $n < k$ on the response variable available, giving $y_1, y_2, ... y_n$, each observed response will have an observation on each regression variable, with $x_{ij}$ denoting the $ith$ observation of variable $x_j$. Assuming that the error term $\varepsilon$ has $\mathrm{E}(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma^2$ and the $\varepsilon_i$ are uncorrelated random variables. The model can now be expressed in terms of the observations

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik} + \varepsilon_i$$
$$i = 1,2,...n \tag{10}$$

The $\beta$ coefficients in equation (10) are chosen such that the sum of the squares of the errors $\varepsilon_i$ are minimised via the least squares function

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij} \right)^2 \tag{11}$$

The model can be more usefully expressed in matrix form as

$$y = \mathrm{X}\beta + \varepsilon \tag{12}$$

where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix}, \mathrm{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & ... & x_{1k} \\ 1 & x_{21} & x_{22} & ... & x_{1k} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 1 & x_{n1} & x_{n2} & ... & x_{nk} \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ . \\ . \\ . \\ \beta_n \end{bmatrix}, \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ . \\ . \\ . \\ \varepsilon_n \end{bmatrix} \tag{13}$$

It is now necessary to find a vector of least squares estimators $b$ which minimises the expression

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \varepsilon' \varepsilon = (y - \mathrm{X}\beta)' (y - \mathrm{X}\beta) \tag{14}$$

and yields the least squares estimator of $\beta$ which is

$$b = (\mathrm{X}'\mathrm{X})^{-1} \mathrm{X}' y \tag{15}$$

and finally, the fitted regression model is

$$\hat{y} = \mathrm{X}b, e = y - \hat{y} \tag{16}$$

where $e$ is the vector of residual errors of the model. The Response Surface Method can thus be implemented upon either simulated or actual experimental results to derive a polynomial expression describing the relationship between the causal inputs and resulting outputs of the dynamic systems under consideration.

As data from the experimental results are gathered under the direction of the meta-heuristics, it is possible to generate a surface approximation for the system under consideration.

### 3.1. WSDS Method

The WSDS method in its basic form is encapsulated in the following pseudo-code.

1. UNTIL
2. run meta-heuristic to global minimum (or acceptable value)
3. END
4. ELSE
5. add *new path data* to *old path data*
6. fit *Normalised Response Surface* to *old path data*
7. generate WSDS surface
8. perform weighted jump
9. END

At present, the method uses a fixed 2nd order response surface, however in future work, it is anticipated that this will be developed into an adaptive system. For the development of this methodology, a realistic noisy surface with multiple local minima, plateaus and one global minimum was considered for algorithm development. The standard Matlab ‘Peaks’ surface (figure 1) is corrupted with significant measurement and process noise, for example (figure 2).

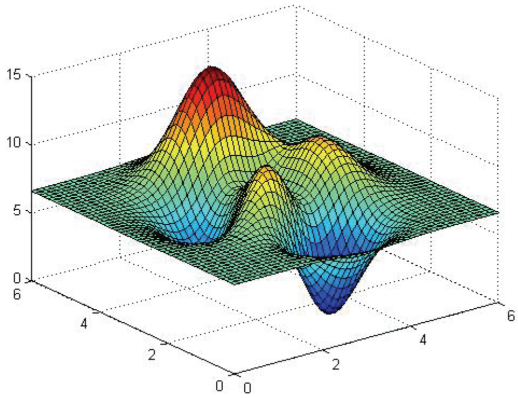

Figure 1: Noise-free Algorithm Development Fitness Landscape - Peaks 0

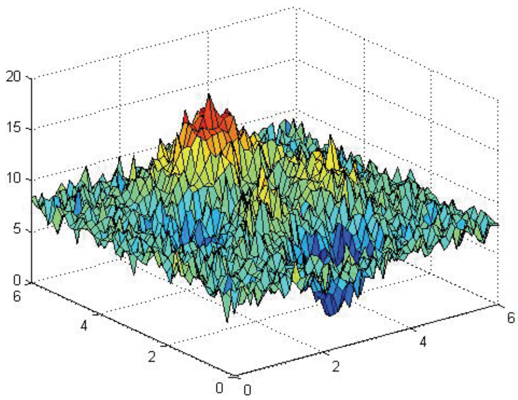

Figure 2: Noisy Algorithm Development Fitness Landscape - Peaks 3

The development surfaces are designated *Peaks 0 - Peaks 3* with increasing levels of noise imposed on the clean *Peaks 0* surface according to;

- Peaks 1 - mean 0.1189, variance 0.0836
- Peaks 2 - mean 0.2842, variance 0.3705
- Peaks 3 - mean 1.7277, variance 0.7648

Using the VNS method as an example, a typical unsuccessful search is shown in figure(3); although unsuccessful, the path does reveal some information about the system, and hence prior to the next “jump” of the meta-heuristic, the most recent path information is added to the path database to generate an up-to date WSDS surface. A fifth order example resulting from the search shown in figure(3), is shown in figure(4).
The contours represent the predicted probability of future successful searches based upon past data.

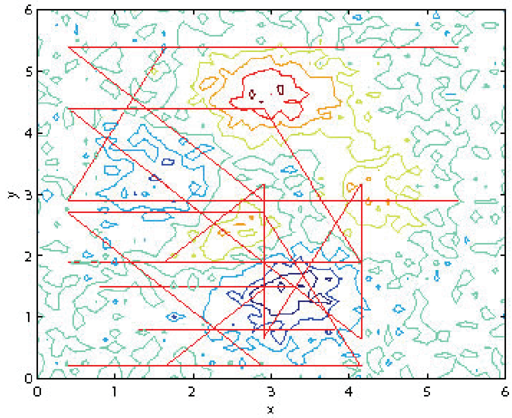


Figure 3: Example VNS Decision Support Path

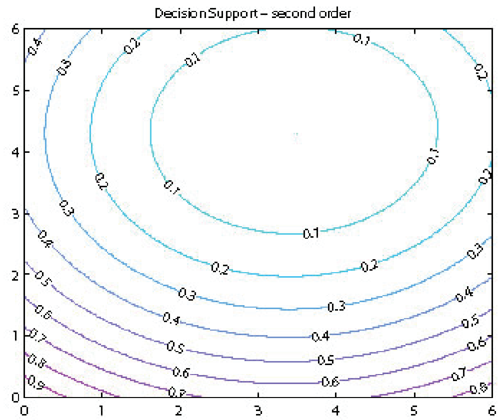


Figure 4: Example Second Order WSDS

The method has successfully identified the lower region of the experimental space as being the most productive in terms of identifying the required acceptable minimum. The parameters for the three search methods are given in table 1.

Table 1: Search Parameters

| Method | Stop-point | Stepsize | Schedule |
|---|---|---|---|
| GD | min+0.2 | 10 | none |
| VNS | min+0.2 | $300 \rightarrow 10$ | 0.5 |
| SA | min+0.2 | 10 | $1 \rightarrow 0.01(0.01)$ |

## 4. EXPERIMENTAL RESULTS AND CONCLUSION

Three methodologies (simple gradient descent, variable neighbourhood search, simulated annealing) were applied to a surface of experimental data taken from an automotive application. A technical description of the experimental data is beyond the scope of this paper, however it relates peak cylinder pressures in a single cylinder IC engine with two control parameters which relate to mean and controlled piston trajectories on an experimental hybrid vehicle drivetrain (figure 5). The objective was to find the control parameters which are associated with a predicted peak cylinder pressure.

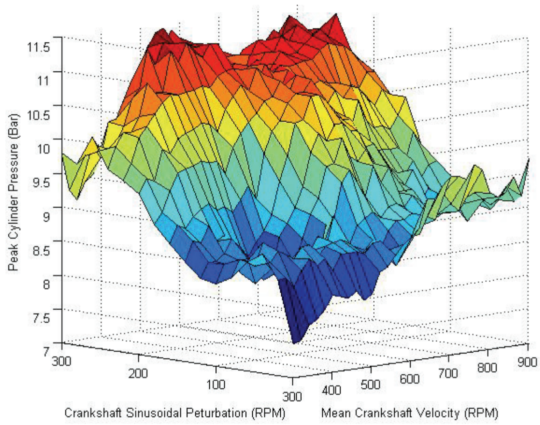

Figure 5: Experimental Engine Response Surface

A comparison of the number of experiments required to find the global maximum by the three search methods is given in table 2, comparing the performance of the methods both with and without the decision support operator. As the methodologies are stochastic by nature, the figures are the averaged results from 50 runs for each application.

Table 2: Support Vs No Support Performance Metrics

| Metric | Support? | GD | VNS | SA |
|---|---|---|---|---|
| Comp | no | 9.6098e+3 | 2.6262e+3 | 7.257e+3 |
| | yes | 757.9 | 1.8033e+3 | 879.82 |
| Std | no | 8.3786e+3 | 2.5261e+3 | 8.1887e+3 |
| | yes | 715.9 | 1.5786e+3 | 564.2 |
| Max | no | 28930 | 1207 | 46090 |
| | yes | 2945 | 2440 | 2776 |

The decision support method gives an approximate order of magnitude improvement in performance when compared to the base methodologies. Further work will include a study of the effectiveness of higher order surfaces, and the inclusion of this study into adaptive schemes which will further increase the performance of this support method.

## REFERENCES


Baldick R., 2002. Applied optimisation, formulation. *Applied Mathematics*, Vol. 119, No. 1, pp. 3-36.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by Simulated Annealing. *Science*, Vol. 220, pp. 671-680.

Michalewicz, Z. and Fogel, D.B., 2006. *How to solve and algorithms for engineering systems.* Cambridge: Cambridge University Press.

Burke, E., Kendall, G., 1990. Comparison of meta-heuristic algorithms for clustering rectangles. *Journal of Computers and Industrial Engineering*, Vol. 37, No. 1, pp. 383-386.

Burke, E., and Kendall, G., 2005. *Search Methodologies: Introductory tutorials in optimisation and decision support techniques*, New York: Springer.

Glover, F., Hanfi, S., 2002. Tabu search and finite convergence. *Journal of Discrete it: modern heuristics,* Berlin: Springer-Verlag.

Mladenovic, M., Hansen, P., 1997. Variable neighbourhood search. *Journal of Computers and Operations Research,* Vol. 24, No. 11, pp. 1097-1100.

Myers, R.H., and Montgomery, D.C., 1995. *Response surface methodology: process and product optimization using designed experiments.* USA: John Wiley and Sons Inc.

Narayana Naik, G., Gopalakrishnan, S., Ganguli, R., 2008. Design optimization of composites using genetic algorithms and failure mechanism based failure criterion. *Composite Structures*, Vol. 83, No. 4, pp. 354-367.

Stewart, P., and Fleming, P.J., 2002. The Response Surface Methodology for rapid prototyping of real-time control systems. *2002 IEEE American Control Conference*, pp. 3343-3348, May 8-10, Anchorage (Alaska, USA).

Stewart, P., Fleming, P.J., MacKenzie, S.A., 2003. Real Time Simulation and Control Systems Design by the Response Surface Methodology and Designed Experiments. *International Journal of Systems Science,* Vol. 34, No. 14-15, pp. 837-850.

Stewart, P., Stone, D.A., Fleming, P.J., 2004. Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop. *IFAC Journal of Engineering Applications of Artificial Intelligence*, Vol.70, no.3, pp. 275-284.

Stewart, P., Fleming, P.J., 2004. Drive by wire control of automotive driveline oscillations by response surface methodology. *IEEE Transactions on Control Systems Technology*, Vol.12(5), pp. 737-741.

Toksari, M.D., Guner, E., 2007. Solving the unconstrained optimisation problem by a variable neighbourhood search. *Journal of Mathematical Analysis and Applications*, No. 328, pp. 1178-1187.



## ACKNOWLEDGMENTS

Technical support, and the hardware used in this research was provided by the EPSRC Project Zero-Constraint Free Piston Energy Converter, Grant Number: GR/S97507/01.