Parameter Screening and Optimisation for ILP using Designed Experiments

Ashwin Srinivasan[1],[2] and Ganesh Ramakrishnan[3]

[1] IBM India Research Laboratory, 4-C, Vasant Kunj Institutional Area
New Delhi 110070, India
[2] Dept. of Computer Science and Engineering & Centre for Health Informatics,
University of New South Wales, Sydney.
ashwin.srinivasan@wolfson.oxon.org
[3] Dept. of Computer Science and Engineering
Indian Institute of Technology, Bombay.
ganramkr@cse.iitb.ac.in

Abstract. Reports of experiments conducted with an Inductive Logic Programming system rarely describe how specific values of parameters of the system are arrived at when constructing models. Usually, no attempt is made to identify sensitive parameters, and those that are used are often given "factory-supplied" default values, or values obtained from some non-systematic exploratory analysis. The immediate consequence of this is, of course, that it is not clear if better models could have been obtained if some form of parameter selection and optimisation had been performed. Questions follow inevitably on the experiments themselves: specifically, are all algorithms being treated fairly, and is the exploratory phase sufficiently well-defined to allow the experiments to be replicated? In this paper, we investigate the use of parameter selection and optimisation techniques grouped under the study of experimental design. Screening and response surface methods determine, in turn, sensitive parameters and good values for these parameters. Screening is done here by constructing a stepwise regression model relating the utility of an ILP system's hypothesis to its input parameters, using systematic combinations of values of input parameters (technically speaking, we use a two-level fractional factorial design of the input parameters). The parameters used by the regression model are taken to be the sensitive parameters for the system for that application. We then seek an assignment of values to these sensitive parameters that maximise the utility of the ILP model. This is done using the technique of constructing a local "response surface". The parameters are then changed following the path of steepest ascent until a locally optimal value is reached. This combined use of parameter selection and response surface-driven optimisation has a long history of application in industrial engineering, and its role in ILP is demonstrated using well-known benchmarks. The results suggest that computational overheads from this preliminary phase are not substantial, and that much can be gained, both on improving system performance and on enabling controlled experimentation, by adopting well-established procedures such as the ones proposed here.

Keywords: ILP, parameter screening and optimisation, experimental design

1 Introduction

We are concerned in this paper with Inductive Logic Programming (ILP) primarily as a tool for constructing models. Specifications of the appropriate use of a tool, its testing, and analysis of benefits and drawbacks over others of a similar nature are matters for the engineer concerned with its routine day-to-day use. Much of the literature on the applications of ILP have, to date, been once-off demonstrations of either the model construction abilities of a specific system, or of the ability of ILP systems to represent and use complex domain-specific relationships [5, 10]. It is not surprising, therefore, that there has been little reported on practical issues that arise with the actual use of an ILP system.

Assuming some reasonable solution has been found to difficult practical problems like the appropriateness of the representation, choice of relevant "background knowledge", poor user-interfaces, and efficiency[4], we are concerned here with a substantially simpler issue. Like all model-building methods, an ILP system's performance is affected by values assigned to input parameters (the term is used here in the sense understood by the computer scientist, and not the statistician). For example, the model constructed by an ILP system may be affected by the maximal length of clauses, the minimum precision allowed for any clause in the theory, the maximum number of new variables that could appear in any clause, and so. The ILP practitioner is immediately confronted with two questions: (a) Which of these parameters are relevant for the particular application at hand?; and (b) What should their values be in order to get a good model? In an industrial setting, an engineer confronted with similar questions about a complex system—a chemical plant, for example—would try to perform some form of sensitivity analysis to determine an answer to (a), and follow it with an attempt to identify optimal values for the parameters identified. As it stands, experimental applications of ILP usually have not used any such systematic approach. Typically, parameters are given "factory-supplied" default values, or values obtained from a limited investigation of performance across a few pre-specified values. The immediate consequence of this is that it is not clear if better models could have been obtained if some form of parameter selection and optimisation had been performed. A measure of the unsatisfactory state of affairs is obtained by considering whether it would be acceptable for a chemical engineer to take a similar approach when attempting to identify optimal operating conditions to maximise the yield of his plant.

The work in [27] addressed the second question—that of optimal values for the input parameters—somewhat indirectly by first constructing an "operating characteristic curve" that describes the performance of the ILP system across a range of values for the relevant variables. While no specific method is proposed

[4] In [28], experience gained from applications of ILP to problems in biochemistry were used to extract some guiding principles of relevance to these problems for any ILP application.

for identifying either the parameters or their values, the characteristic curve provides a way of selecting amongst models obtained by varying parameter values, provided model goodness is restricted to a specific class (that of cost functions that are linear in the error-rates). The work of Bengio [2] is more closely related, in that it presents a methodology to optimize several parameters (Bengio calls them hyperparameters, to avoid confusion with the statistical term), based on the computation of the gradient of a model selection criterion expressed in terms of the hyperparameters. The main restriction is that this criterion must be a known, continuous and differentiable function of the hyperparameters (almost) everywhere. In almost all ILP settings, the training criterion cannot be even expressed in closed form, let alone being a differentiable and continuous function of the hyperparameters. That is, what can be done at best is to treat the ILP system is a black box and its variation as a function of the hyperparameters can be measured only empirically in terms of the response of the system to changes in the values of the hyperparameters.

Here take up the questions of screening and optimisation of parameters directly with the only restrictions being that parameter and goodness values are quantitative in nature. The methods we use have origins in optimising industrial processes [4] and been developed under the broad area concerned with the design and analysis of experiments. This area is concerned principally with discovering something about a black-box system by designing deliberate changes to the system's input variables, and analysing changes in its output response. The representation of a system is usually as shown in Fig. 1(a) (from [20]). The process being modelled transforms some input into an output that is characterised a measurable response y. The system has some controllable factors, and some uncontrollable ones and the goals of an experiment could be to answer questions like: which of the controllable factors are most influential on y; and what levels should these factors be for y to reach an optimal value. The relevance of the setting to the ILP problem we are considering here will be evident in Section 2.

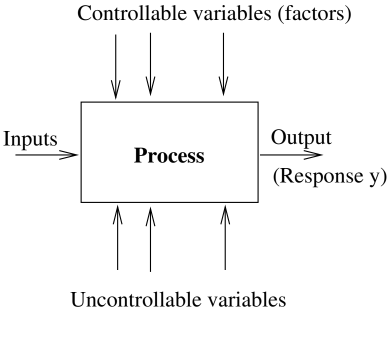


Fig. 1. Model of a system used in experimental design (from [20]). The process can be a combination of systems, each modelled by some input-output behaviour.

There are a wide variety of techniques developed within the area of experimental design: we will be concentrating here on some of the simplest, based around the use of regression models. Specifically, using designed variations of input variables, we will use a stepwise linear regression strategy to identify variables most relevant to the ILP system's output response. This resulting linear model, or response surface, is then used to change progressively the values of the relevant variables until a locally optimal value of the output is reached. We demonstrate this approach empirically on some ILP benchmarks.

The rest of this paper is organised as follows. Sections 2 and 3 provide background details on the kinds of ILP systems we consider here and some relevant details on techniques in experimental design. Section 4 describes, first, two empirical studies. The studies demonstrate how, for a given set of inputs, parameter screening and selection using designed experiments yields a better model than simply using default values, or performing an exhaustive combination of predetermined values for parameters. They also demonstrate how, if inputs are changed, then both the set of relevant parameters and their values can change. These experiments are then followed up with others that use some well-known benchmark datasets. The results confirm the findings from the primary investigation; and demonstrate how the procedures proposed here can be used for controlled comparisons of ILP systems. Section 5 concludes the paper. The paper is accompanied by two appendices that provide standard material from the literature concerned with the construction of linear models, and with specific aspects of the optimisation method used here.

2 An Engineer's View of ILP Systems

Inductive Logic Programming (ILP) has been largely characterised by two classes of programs. The first, predictive ILP, has been concerned with constructing discriminative models (sets of rules; or first-order variants of classification or regression trees) for distinguishing accurately amongst two sets of examples ("positive" and "negative"), or more generally, amongst examples classified into one of several classes. The second category of ILP programs, descriptive ILP, has been concerned with generative models of relationships that hold amongst the background knowledge and examples. This latter category includes programs that identifies logical constraints in a database [9] and more recently, programs that capture complex probabilistic relationships amongst objects (the area of statistical relational learning: [12]).

In this paper, we take an engineer's view of ILP. In this, an ILP implementation is a system that, given some inputs—in usual ILP terminology, background knowledge and examples—and settings for parameters, some of which are under the control of the engineer, produces an output model by performing some form of optimisation (see Fig. 2). For example, many ILP systems that explore the space of alternatives imposed by the inverse entailment setting proposed in [21] could be seen as performing a form of discrete optimisation, using some approximation to a branch-and-bound search procedure. The task of the system engineer is then to tune the parameters under his or her control to enable the system to return the best performance.[5] In [27], for example, it is demonstrated how widely varying performance can be obtained by varying a single parameter (the minimum accuracy of clauses found in a search).

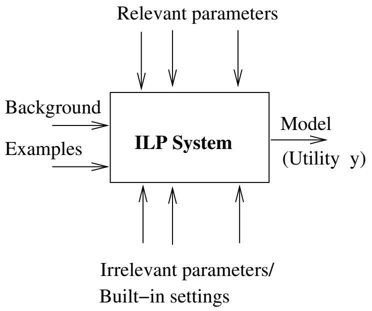


Fig. 2. An system engineer's view of an ILP system. We are assuming here that "Background" includes syntactic and semantic constraints on acceptable models. "Built-in settings" are the result of decisions made in the design of the ILP system. An example is the optimisation function used by the system.

The immediate difficulty is, of course, that it is usually impractical to examine the system's performance by enumerating every possible combination of values for the controllable parameters. With ILP systems there are two further difficulties. First, it may often not be known beforehand which parameters are actually relevant to system for the problem being solved. The system Aleph [26] provides perhaps the most clear instance of this: see Fig. 3. Second, models constructed, and hence system performance, can vary even if all inputs and parameters have fixed values: for example, the system may use a search strategy that employ some random choices ([31] provides an example of such a strategy).

Within ILP, no significant attention has been paid to this problem. Reports in the literature rarely contain any discussion of sensitive parameters of the system or their values.[6] The problem of selection and tuning of parameters to optimise system performance has, however, been studied extensively in areas of industrial

[5] This is different to improving the optimisation procedure performed by the system itself. Rather, it is concerned with enabling the existing optimisation procedure find better results, usually by changing the space of alternatives in some principled manner. It is beyond the engineer's remit to alter either the sytem's inputs or its optimisation criterion as a means of improving system performance.

[6] Of 100 experimental studies reported in papers presented between 1998 and 2008 to the principal conference in the area, none attempt any form of screening for relevant parameters. 17 describe settings for some pre-selected parameters—usually one—from performance estimates obtained during an enumerative search over some small set of possible values (that is, effectively using the wrapper approach of [18]). 38 reports, however, mention values assigned to *some* parameters, without elucidating

```
1. The following parameters can affect the size of the search space:
      i, clauselength, nodes, minpos, minacc,
      noise, explore, best, openlist, splitvars.
2. The following parameters affect the type of search:
      search, evalfn, refine, samplesize.
3. The following parameters have an effect on the speed of execution:
      caching, lazy_negs, proof_strategy, depth,
      lazy_on_cost, lazy_on_contradiction, searchtime, prooftime.
4. The following parameters alter the way things are presented to the user:
      print, record, portray_hypothesis, portray_search,
      portray_literals, verbosity,
5. The following parameters are concerned with testing theories:
      test_pos, test_neg, train_pos, train_neg.
```

Fig. 3. A categorisation of some of the parameters of the ILP system Aleph (reproduced from [26]). Not all of these are relevant to every problem being solved.

engineering, using results obtained in the design and analysis of experiments. It is our intention in this paper to apply the techniques developed in these areas to ILP.

3 Design and Analysis of Experiments

The area broadly concerned with the design of experiments (DOE) deals with devising deliberate variations in the values of input variables, or *factors*, and analysing the resulting variations in a set of one or more output, or *response*, variables. The objectives of conducting such experiments are usually: (a) Understand how variation or uncertainty in input values affects the output value. The goal here is the construction of robust systems in which a system's output is affected minimally by external sources of variation; and (b) Maximise (or minimise) a system's performance. In turn, these raise questions like the following: which factors are important for the response variables; what values should be given to these factors to maximise (or minimise) the values of the response variables; what values should be give to the factors in order that variability in the response is minimal, and so on.

In this paper, we will restrict ourselves to a single response variable and the analysis of experimental designs by multiple regression. It follows, therefore, that we are restricted in turn to quantitative factors only. Further, by "experimental design" we will mean nothing more than a selection of points from the factor-space, in order that a statistically sound relationship between the factors and the response variable can be obtained. Each factor-level combination will constitute

how these values were reached (on occassions, these were just the default values provided by the system).

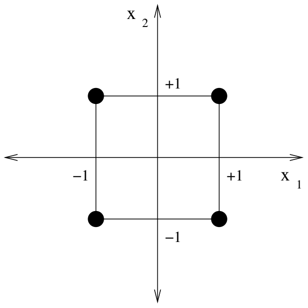
an experiment, and a design will therefore require us to specify the experiments and, if necessary, the number of replications of each experiment.

3.1 Screening using Factorial Designs

We first consider designs appropriate for screening. By this, we mean deciding which of a set of potentially relevant factors are really important, statistically speaking. The usual approach adopted is what is termed a 2-level factorial design. In this, each factor is taken to have just two levels (encoded as "-1" and "+1", say)[7], and the effect observed on the response variable of changing the levels of each factor. It is evident that with k factors, this will result in 2^k experiments, each of which may need to be repeated in case there is some source of random variation in the response variable. For example, with two factors, conducting a 2^2 full factorial design will result in a table such as the one shown in Fig. 4

Expt.	**Factor**	**Factor**	**Response**
	x_1	x_2	y
E1	-1	-1	...
E2	-1	+1	...
E3	+1	-1	...
E4	+1	+1	...

(a)



(b)

Fig. 4. (a) A 2-level full factorial design for two factors; and (b) a graphical representation of the design.

[7] One way to achieve the coded value x of a factor X is as follows. Let X^- and X^+ be the minimum and maximum values of X (these are pre-specified). Then $x = \frac{X-(X^+ + X^-)/2}{(X^+ - X^-)/2}$.

We are then able to construct a regression model relating the response variable to the factors:

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1 x_2$$

The model describes the effect of each factor $x_{1,2}$ and interactive effect $x_1 x_2$ of the two factors on y.[8] It is usual also to add "centre points" to the design in the form of experiments that obtain values for y for $x_1 = 0$ and $x_2 = 0$. The results of these experiments will not contribute to estimation of the coefficients $b_{1,2,3}$ (since the x_i are all 0s), but allows us to obtain a better estimate for the value of b_0. Further, it is also the case that with a 2-level full factorial design only linear effects can be estimated (that is, the effect of terms like x_i^2 cannot be obtained: in general, a n^{th} order polynomial will require $n+1$ levels for each factor). In this paper, we will use the coefficients of the regression model to guide the screening of parameters: that is, parameters with coefficients significantly different from 0 will be taken to be relevant (more on this in Appendix A).

Clearly, the number of experiments required in a full factorial design constitute a substantial computational burden, especially as the number of factors increase. Consider, however, the role these experiments play in the regression model. Some are necessary for estimating the effects of each factor (that is, the coefficients of $x_1, x_2, x_3, \ldots$: usually called the "main effects"), others for estimating the coefficients for two-way interactions (the coefficients of $x_1 x_2$, $x_1 x_3$, $\ldots$) , others for three-way interactions ($x_1 x_2 x_3$, $\ldots$) and so on. However, in a screening stage, all that we wish to do is to identify the main effects. This can usually be done with fewer than the 2^k experiments needed for a full factorial design with k factors. The result is a 2-level "fractional" factorial design. Figure 5 below illustrates a 2-level fractional factorial design for 3 factors that uses half the number of experiments to estimate the main effects (from [29]).

The experiments in the fractional design have been selected so that $x_1 x_2 x_3 = +1$. Closer examination of the table on the right will make it clear that the following equalities also hold for this table: $x_1 = x_2 x_3$; $x_2 = x_1 x_3$; and $x_3 = x_1 x_2$. That is, main effects and interaction terms are confounded with each other. This has some direct implications when constructing regression models using the fractional table. In effect, instead of the full regression model:

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_1 x_2 + b_5 x_1 x_3 + b_6 x_2 x_3$$

we are reduced to obtaining the following model:

$$y = b_0 + b'_1(x_1 + x_2 x_3) + b'_2(x_2 + x_1 x_3) + b'_3(x_3 + x_1 x_2)$$

In fact, a regression program will be unable, for example, to distinguish the regression model above from this one:

$$y = b_0 + b''_1 x_1 + b''_2 x_2 + b''_3 x_3$$

[8] Interaction effects happen if the effect of a factor, say X_1 on the response depends on the level of another factor X_2.

Expt.	x_1	x_2	x_3	y
E1	-1	-1	-1	...
E2	-1	-1	+1	...
E3	-1	+1	-1	...
E4	-1	+1	+1	...
E5	+1	-1	-1	...
E6	+1	-1	+1	...
E7	+1	+1	-1	...
E8	+1	+1	+1	...

Expt.	x_1	x_2	x_3	y
E2	-1	-1	+1	...
E3	-1	+1	-1	...
E5	+1	-1	-1	...
E8	+1	+1	+1	...

Fig. 5. A full 2-level factorial design for 3 factors (left) and a "half fraction" design (right).

or even this:

$$y = b_0 + b_1'''x_1 + b_2'''x_2 + b_3'''x_1x_2$$

The b_i'' and b_i''' will differ from the b_i' by a factor of 2, but this will not change the model's fit of the data, since the corresponding independent variables in the regression equation would be halved (x_1 instead of $x_1 + x_2x_3$ and so on). Thus, the price for fractional experiments is therefore, that we will in general, be unable to distinguish the effects of all the terms in the full regression model. However, if it is our intention—as it is in the screening stage—only to estimate the main effects (such models are also called "first-order" models), then we can ignore interactions (see Fig. 6). Main effects can be estimated with a table that is a fraction required by the full factorial design: for example, the half fraction in Fig. 5 is sufficient to obtain a regression equation with just the main effects x_1, x_2 and x_3.[9]

More details on fractional designs are provided in Appendix A. We use the techniques and results described there to direct the screening of factors by focussing on a linear model that contains the main effects only:

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_kx_k$$

Depending on the number of factors, this can be done with a fractional designs of "Resolution III" or above (see Appendix A). Standard tests of significance can be performed on each of the coefficients $b_1, b_2, \ldots, b_k$ to screen factors for relevance (the null and alternative hypotheses in each case are $H_0 : b_i = 0$ and $H_1 : b_i \neq 0$). In fact, this test is the basis for inclusion or exclusion of factors by stepwise regression procedures (see Appendix A). Using such a procedure would

[9] This is apparent from the fact that n distinct data points are needed to fit a regression model with n terms. Thus, when fitting a model with just x_1, x_2, and x_3, we need 4 data points.

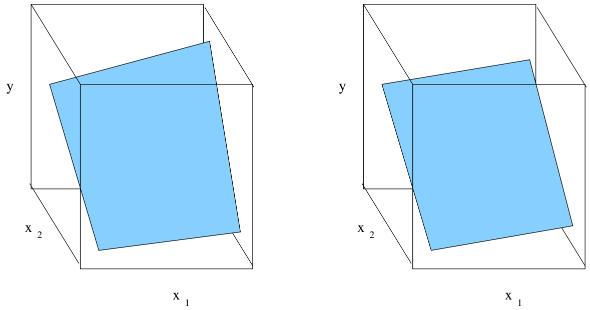


Fig. 6. A surface with a "twist" arising from interactions between the factors (left) and a planar approximation that ignores this twist (right). For the purpose of estimating the main effects, the surface on the right is adequate, as it shows that x_2 has a much bigger effect than x_1 on the response y (we are assuming here that x_1 and x_2 represent coded values on the same scale).

naturally return a model with only the relevant factors (the use of stepwise regression is also the preferred method for sensitivity analysis suggested at the end of the extensive survey in [14]).

3.2 Optimisation Using the Response Surface

Suppose screening in the manner just described yields a set of k relevant factors from a original set of n factors (which we will denote here as $x_1, x_2, \ldots, x_k$ for convenience). We are now in the position of describing the functional relationship between the expected value of the response variable and the relevant factors, by the "response surface":

$$E(y) = f(x_1, x_2, \ldots, x_k)$$

Usually, f is taken to be some low-order polynomial, either a first-order model involving only the main effects $x_1, x_2, \ldots$ (recall that if stepwise regression procedure is used at the screening stage, then this is the model that would be obtained):

$$y = b_0 + \sum_{i=1}^{k} b_i x_i$$

or a second-order model involving quadratic terms like $x_1^2, x_2^2, \ldots$ and linear interaction terms like $x_1x_2, x_1x_3, \ldots$:

$$y = b_0 + \sum_{i=1}^{k} b_i x_i + \sum_{i=1}^{k} b_{ii} x_i^2 + \sum_{i=1}^{k} \sum_{j>i} b_{ij} x_i x_j$$

Clearly, if first-order models are adequate (this can be checked by an analysis of how well the model fits the data: see Appendix A) then much of the effort expended in the screening stage can be re-used (for example, we can use the model constructed by stepwise regression as the response surface model). A second-order model, on the other hand, will require experiments involving additional levels for each factor, and some effort has been invested in the literature on determining these levels. Since first-order models are all that are used in this paper, we do not pursue this further here, and refer the reader to a standard text like [20] for more details.

The principal approach adopted in optimising using the response surface is a sequential one. First, a local approximation to the true response surface is constructed, using a first-order model. Next, factors are varied along a path that improves the response the most (more on this in a moment). Experiments are conducted along this direction and the corresponding responses obtained until no further increase in the response is observed. At this point, a new first-order response surface is constructed, and the process repeated until it is evident that a first-order model is inadequate (or no more increases are possible). If the fit of the first-order model is poor, a more detailed model is then obtained—usually a second-order model, using additional levels for factors—and its stationary point obtained. The basic idea is illustrated in Fig. 7 (from [20]).

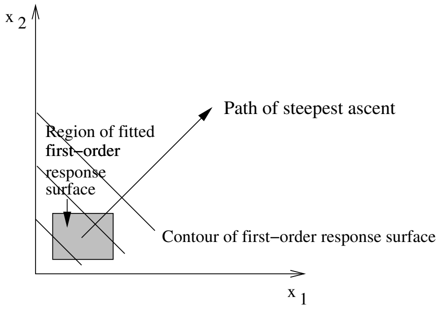


Fig. 7. Sequential optimisation of the response surface using the path of steepest ascent. A first-order response surface is obtained in the shaded region. The factors are then changed to move along a direction that gives the maximum increase in the response variable.

Now, we can view the response y to be given by a scalar field f that at each point $x_1, x_2, \ldots, x_k$ gives the response $f(x_1, x_2, \ldots, x_k)$. Then, from standard vector calculus, the gradient of f at the point gives the direction in which the response will change most quickly (that is, the direction of steepest ascent: see Appendix A). This gradient, usually denoted ∇f, is given by $\left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_k}\right)$. The sequential optimisation of the response surface just described involves calculating the gradient of the first-order model at the centre, or origin, of the experimental design ($x_1 = x_2 = \cdots = 0$). For a model of the form $f(x_1, \ldots, x_k) = b_0 + b_1x_1 + \cdots + b_kx_k$, ∇f is simply $(b_1, \ldots, b_k)$. For convenience, let us take b_1 to have the largest absolute value. Then, along the direction of ∇f, a unit change in x_1 will result in a change of b_2/b_1 units of x_2, b_3/b_1 units of x_3 and so on. Sequential response optimisation proceeds by starting at the origin and increasing the x_i along ∇f until increases in the response y is observed. Each such increase results in a new experiment to be performed (see Fig.8, for an example with 3 factors).

Expt.	**Factor**	**Factor**	**Factor**	**Response**
	x_1	x_2	x_3	y
E9	0	0	0	...
E10	δ	$\frac{b_2}{b_1}\delta$	$\frac{b_3}{b_1}\delta$	...
E11	2δ	$\frac{2b_2}{b_1}\delta$	$\frac{2b_3}{b_1}\delta$	...
E12	3δ	$\frac{3b_2}{b_1}\delta$	$\frac{3b_3}{b_1}\delta$	...
...	...	...	...	...

Fig. 8. Sequential experiments that obtain new values for y by moving in the direction of the gradient to $b_0 + b_1x_1 + b_2x_2 + b_3x_3$. Experiments E1–E8 are as in Fig. 5.

3.3 Screening and Optimisation for ILP

We are now in a position to put together the material in the previous sections to state more fully a procedure for screening and optimisation of parameters for an ILP system:

SO: Screen quantitative parameters using a two-level fractional factorial design, and optimise values using the response surface.

*ScreenFrac***.** Screen for relevant parameters using the following steps:

$S1$. Decide on a set of n quantitative parameters of the ILP system that are of potential relevance. These are the factors x_i in the sense just described. Take some quantitative summary of the model constructed by the system—for example, its estimated predictive accuracy—as the response variable y (we will assume here that we wish to maximise the response).

$S2$. Decide on on two levels (“low” and “high” values) for each of the factors. These are then coded as ±1.

$S3$. Devise a two-level fractional factorial design of Resolution III or higher, and obtain values of y for each experiment (or replicates of values of y, if so required).

$S4$. Construct a first-order regression model to estimate the role of the main effects x_i on y. Retain only those factors that are important, by examining the magnitude and significance of the coefficients of the x_i in the regression model (alternatively, only those factors found by a stepwise regression procedure are retained: see Appendix A).

*OptimiseRSM***.** Optimise values of relevant parameters using the following steps:

$O1$. Construct a first-order response surface using the relevant factors only (this is not needed if stepwise regression was used at the screening stage). If no adequate model is obtained, then return the combination of factor-values that gave the best response at the screening stage. Otherwise go to Step O2.

$O2$. Progressively obtain new values for y by changing the relevant parameters along the gradient to the response surface. Stop when no increases in y are observed.[10]

$O3$. If needed, construct a new first-order response surface. If this surface is adequate, then return to Step O2. Otherwise, go to Step O4.

$O4$. If needed, construct a second-order response surface. Return the optimum values of the relevant factors using the second-order surface, or from the last set of values from Step $O2$.[11]

We contrast OptimiseRSM with the multi-level full factorial design below, which has been used on a few occasions within the ILP literature:

*OptimiseFact***.** Optimise values of relevant parameters using the following steps:

$O1'$. Decide on on multiple levels for each of the relevant factors.

$O2'$. Devise a full factorial design by combing each of the levels of the factors against those of the others. For each such combination, obtain values of y for each experiment (or replicates of values of y, if so required).

$O3'$. Select the combination of values that yielded the highest value of y (including those obtained at the screening stage).

This procedure, a multi-level full factorial design, is the basis of the wrapper-based optimisation method of [18], recast in the terminology of experimental design. A simplified analysis gives us some feel of the complexity of **SO**. **SO** conducts some fraction of 2^n experiments in the *ScreenFrac* stage, followed by those conducted in OptimiseRSM. Suppose we always conduct a 2^{n-p}-fractional design at the screening stage, and that this stage results in no more than r variables being selected as relevant. Further, let each round of sequential optimisation consist of s experiments in Step O2. Let there be m such rounds of sequential optimisation, each followed by a new first-order model in Step O3 (since there are r variables, building this model will require an additional $r + 1$

[10] In practice, this is taken to mean that no increases have been observed for some number of consecutive experimental runs: the so-called “k-in-a-row” stopping rule.

[11] We note that the use of gradient ascent in this manner is only capable of finding local maxima in y values. A question is raised about what is to be done if the local maximum found in this manner is *lower* than a response value known already–for example, from an experiment from the screening stage. A modification would be return the combination of factor-values that give the best y value obtained over all experiments. This would be at variance with standard response-surface optimisation, and we do not consider it here.

experiments). Finally a second-order model is constructed (Step O4), using a central composite design. Then the total number of experiments conducted by **SO** is: 2^{n-p} (screening) + ms (sequential optimisation) + $(m-1)(r+1)$ (new first-order models) + $2r+1$ (second-order model). In the case that only one round of sequential experimentation is performed (that is, $m = 1$) and no additional first- or second-order models are constructed, the number of experiments is simply $2^{n-p} + s$. It is evident that a procedure **SO′** that employs *ScreenFrac* followed by *OptimiseFact* would always perform $2^{n-p} + l^r$ experiments (assuming, for simplicity, that all relevant factors are taken to have l levels during the optimisation stage). This is no more than $2^{n-p} + l^n$.

4 Empirical Evaluation

4.1 Aims

Our aim here is to demonstrate the utility of the screening and optimisation procedure **SO** that we have described in Section 3.3 (that is, **SO** is *ScreenFrac* followed by *OptimiseRSM*). We assess this utility by comparing the ILP system when it employs **SO** against the performance of the system when it uses one of following alternatives: **Default**, in which no screening or optimisation is performed and default values provided for all parameters are used; and **SO′**, in which screening is performed as in **SO**, but a multi-level full factorial design is used for optimisation (that is, **SO′** is *ScreenFrac* followed by *OptimiseFact*). Specifically, we intend to investigate the following conjectures:

$C1$. Using **SO** is better than using **Default**; and
$C2$. Using **SO** is better than using **SO′**.

In both cases, "better" is short-form for stating that an ILP system that uses **SO** has better predictive performance; or in the case of ties, requires fewer experiments than the alternative.

4.2 Materials

Domains The investigation is conducted first on the well-studied ILP biochemical problems concerned with identifying mutagenic and carcinogenic chemicals. Although we will extend it later to other datasets used in the literature, we have selected to focus on these problems first since they constitute perhaps the most commonly used inputs for demonstrating the performance of ILP systems. The data have been described extensively elsewhere (for example, see [16] for mutagenesis; and [17] for carcinogenesis) and we refer the reader to these reports for details. For each application, the input to an ILP can vary depending on the background information used. We investigate the conjectures $C1$ and $C2$ with minimal and maximal amount of background knowledge contained in these benchmarks. That is:

Mutagenesis. We consider background information in the sets M0 and M0–M4, descriptions of which are reproduced below from [27]:

M0. Molecular description at the atomic level. This includes the atom and bond structure, the partial charges on atoms, and arithmetic constraints (equalities and inequalities). There are 5 predicates in this group;

M1. Structural properties identified by experts as being related to mutagenic activity. These are: the presence of three or more benzene rings, and membership in a class of compounds called acenthrylenes. There are 2 predicates in this group;

M2. Chemical properties identified by experts as being related to mutagenic activity, along with arithmetic constraints (equalities and inequalities) The chemical properties are: the energy level of the lowest unoccupied molecular orbital ("LUMO") in the compound, an artificial property related to this energy level (see [8]), and the hydrophobicity of the compound. There are 6 predicates in this group;

M3. Generic planar groups. These include generic structures like benzene rings, methyl groups, *etc.*, and predicates to determine connectivity amongst such groups. There are 14 predicates in this group; and

M4. Three-dimensional structure. These include the positions of individual atoms, and constraints on distances between atom-pairs. There are 2 predicates in this group.

Carcinogenesis. We consider background information in the sets C0 and C0–C3, descriptions of which reproduced below, once again from [27]:

C0. Molecular description at the atomic level. This is similar to M0 above and is comprised of 5 predicates;

C1. Toxicity properties identified by experts as being related to carcinogenic activity, and arithmetic constraints. These are an interpretation of the descriptions in [1], and are contained within the definitions of 5 predicates;

C2. Short-term assays for genetic risks. These include the *Salmonella* assay, in-vivo tests for the induction of micro-nuclei in rat and mouse bone marrow *etc.* The test results are simply "positive" or "negative" depending on the response and are encoded by a single predicate definition; and

C3. Generic planar groups. These are similar to M3 above, extended to 30 predicate definitions.

We will henceforth refer to background knowledge with the definitions in M0 (respectively, C0) as B_{min} and with the definitions in M0–M4 (respectively, C0–C3) as B_{max}.

Algorithms and Machines Experimental design and regression models for screening and the response surface are constructed by the procedures made available by the authors of [29]. The ILP system used in all experiments will be Aleph [26]. The programs are executed on a IBM Thinkpad (T43p), equipped with an Intel 2 GHz Pentium processor with 1 gigabyte of random access memory.

4.3 Method

Our method is straightforward:

> For each problem (Mutagenesis and Carcinogenesis) and each level of background knowledge (B_{min} and B_{max}):
>
> 1. Construct a model with the ILP system using default values for all parameters of the ILP system. Call this model **ILP+Default**.
> 2. Select a set of n quantitative parameters of the ILP system as being potentially relevant. Use the procedure $ScreenFrac$ described in Section 3.3 to screen this set using a fractional factorial design of Resolution III or higher. Let this result in a set of relevant variables R.
> 3. Use the procedure $OptimiseRSM$ in Section 3.3 to obtain values for variables in R. All other parameters of the ILP system are left at their default values. Construct a model using the ILP system with this set of values. Call this model **ILP+SO**.
> 4. Decide on l levels for each variable in R and use the procedure $OptimiseFact$ in Section 3.3 to obtain values for the variables in R. All other parameters of the ILP system are left at their default values. Construct a model using the ILP system with this set of values. Call this model **ILP+SO′**.
> 5. Compare the performance of the ILP system when it produces as output each of **ILP+Default**, **ILP+SO**, and **ILP+SO′** (see the details below).

The following details are relevant:

1. Since the tasks considered here are binary classification tasks, the performance of the ILP system in all experiments will be taken to be the classification accuracy of the model produced by the system. By this we mean the usual measure computed from a 2 × 2 cross-tabulation of actual and predicted classes of instances. The entries in the 2 × 2 table are estimated here using 10-fold cross-validation.
2. We have no general prescription for the selection of the initial set of n parameters (Step 2). We postpone a discussion of this limitation to Section 4.4. For our experiments we have selected four parameters: C, the maximum number of literals in any acceptable clause constructed by the ILP system; $Nodes$, the maximum number of nodes explored in any single search conducted by the ILP system; $Minacc$, the minimum accuracy required of any acceptable clause; and $Minpos$, the minimum number of positive examples to be entailed by any acceptable clause. C and $Nodes$ are directly concerned with the search space explored by the ILP system. $Minacc$ and $Minpos$ are concerned with the quality of results returned (they are equivalent to "precision" and "support" used in the data mining literature). We propose to examine a two-level fractional factorial design, using the levels shown below (the column "Default" refers to the default values for the factors assigned by the Aleph system, and ±1 refers to the coded values of the factors):

Factor	**Levels**		
	Default	Low (-1)	High ($+1$)
C	4	4	8
$Nodes$	5000	5000	10000
$Minacc$	$+1$	0.75	0.90
$Minpos$	1	5	10

3. We use a Resolution IV design, that comprises of a randomised presentation of the following 8 experiments (recall the full factorial design will require $2^4 = 16$ experiments):

Expt.	C	$Nodes$	$Minacc$	$Minpos$	$Accuracy$
E1	-1	-1	-1	-1	...
E2	-1	-1	$+1$	$+1$	...
E3	-1	$+1$	-1	$+1$	...
E4	-1	$+1$	$+1$	-1	...
E5	$+1$	-1	-1	$+1$	...
E6	$+1$	-1	$+1$	-1	...
E7	$+1$	$+1$	-1	-1	...
E8	$+1$	$+1$	$+1$	$+1$	...

   This design was obtained using the software tools for experimental design provided with [29]. The "Accuracy" column is obtained for each task, and for each of the two sets of background knowledge in order to screen the four variables for relevance. Additional experiments will be needed in Step 3 to obtain values of the relevant parameters using the response surface. We restrict ourself to constructing just one first-order regression model for screening, using the stepwise regression procedure provided by the authors of [29]. This model is taken to approximate the local response surface: we then proceed to change levels of factors along the normal to this surface in the manner described in Fig. 8. Experiments are stopped once a maximal value for the response variable is followed by three consecutive runs that yield responses that are no higher.
4. In the event that all four parameters chosen are relevant, the step of obtaining parameter values using a multi-level full factorial design (Step 5) would require conducting l^4 experiments. We will take $l = 5$, which means that, in the worst case, no more than 625 experiments will be conducted to obtain model **ILP+SO′**. Inspired by the choices made for a so-called "Central Composite" (or CC) design [20], we will take the (coded) levels to be 0, ±1, and $\pm\sqrt{2}$.
5. Comparisons of models will be done on the basis of their classification accuracy (estimated as explained in (1) above). In the event of ties, then the model requiring fewer experiments will be preferred. That is, a model is

represented by the pair (A, E) (denoting estimated accuracy and number of experiments required to identify the model). Comparisons are then based on the usual definition of a lexicographic ordering on such tuples.

6. Further, since it is of particular relevance to ILP practitioners, we also test for statistical differences between the accuracies of **ILP+SO** and **ILP+ Default** using results on additional datasets used in the ILP literature. This is done using the Wilcoxon signed-rank test [25]. This is a non-parametric test of the null hypothesis that there is no significant difference between the median performance of the two procedures. The test works by ranking the absolute value of the differences observed in performance of the pair of algorithms. Ties are discarded and the ranks are then given signs depending on whether the performance of the first algorithm is higher or lower than that of the second. If the null hypothesis holds, the sum of the signed ranks should be approximately 0. The probabilities of observing the actual signed rank sum can be obtained by an exact calculation (if the number of entries is less than 10), or by using a normal approximation. We note that the comparing a pair of algorithms using the Wilcoxon test is equivalent to determining if the area under the ROC curves of the algorithms differ significantly [13].

4.4 Results and Discussion

We present first the results concerned with screening for relevant factors. Figure 9 show responses from the ILP system for experiments conducted for screening using the fractional design described under "Methods". The sequence of experiments following this stage for optimising relevant parameter values using: (a) the response surface; and (b) a multi-level full factorial design are in Figs. 10 and 11. Finally, a comparison of the three procedures **ILP+Default**, **ILP+SO**, and **ILP+SO′** is in Fig. 12. It is this last tabulation that is of direct relevance to the experimental aims of this paper, and we note the following: (1) Although no experimentation is needed for the use of default values, the model obtained with **ILP+Default** usually has the lowest predictive accuracies (the exception is Carcinogenesis, with B_{min})[12]; (2) The classification accuracy of **ILP+SO** is never lower than that of any of the other methods; (3) When the classification accuracies of **ILP+SO** and **ILP+SO′** are comparable, the number of experiments needed by the former is lower; and (4) When the number of experiments for **ILP+SO** is more than those for **ILP+SO′** the classification accuracies of the former are higher.

Taken together, these observations provide reasonable empirical evidence for the conjectures made at the outset of this section, namely:

$C1$**.** Using **SO** is better than using **Default**; and
$C2$**.** Using **SO** is better than using **SO′**.

[12] We recall that the regression model obtained for Carcinogenesis (B_{min}) was not very good. This affects the performances of both **ILP+SO** and **ILP+SO′**.

Expt.	C	$Nodes$	$Minacc$	$Minpos$	Acc
E1	-1	-1	-1	-1	0.798
E2	-1	-1	$+1$	$+1$	0.612
E3	-1	$+1$	-1	$+1$	0.771
E4	-1	$+1$	$+1$	-1	0.723
E5	$+1$	-1	-1	$+1$	0.771
E6	$+1$	-1	$+1$	-1	0.761
E7	$+1$	$+1$	-1	-1	0.803
E8	$+1$	$+1$	$+1$	$+1$	0.612

(a) Mutagenesis (B_{min})
$Acc = 0.731 - 0.054\ Minacc - 0.040\ Minpos$

Expt.	C	$Nodes$	$Minacc$	$Minpos$	Acc
E1	-1	-1	-1	-1	0.883
E2	-1	-1	$+1$	$+1$	0.845
E3	-1	$+1$	-1	$+1$	0.883
E4	-1	$+1$	$+1$	-1	0.867
E5	$+1$	-1	-1	$+1$	0.883
E6	$+1$	-1	$+1$	-1	0.872
E7	$+1$	$+1$	-1	-1	0.883
E8	$+1$	$+1$	$+1$	$+1$	0.862

(b) Mutagenesis (B_{max})
$Acc = 0.872 - 0.011\ Minacc$

Expt.	C	$Nodes$	$Minacc$	$Minpos$	Acc
E1	-1	-1	-1	-1	0.481
E2	-1	-1	$+1$	$+1$	0.454
E3	-1	$+1$	-1	$+1$	0.439
E4	-1	$+1$	$+1$	-1	0.460
E5	$+1$	-1	-1	$+1$	0.424
E6	$+1$	-1	$+1$	-1	0.490
E7	$+1$	$+1$	-1	-1	0.445
E8	$+1$	$+1$	$+1$	$+1$	0.460

(a) Carcinogenesis (B_{min})
$Acc = 0.456 - 0.013\ Minpos$ (*)

Expt.	C	$Nodes$	$Minacc$	$Minpos$	Acc
E1	-1	-1	-1	-1	0.591
E2	-1	-1	$+1$	$+1$	0.475
E3	-1	$+1$	-1	$+1$	0.564
E4	-1	$+1$	$+1$	-1	0.525
E5	$+1$	-1	-1	$+1$	0.561
E6	$+1$	-1	$+1$	-1	0.490
E7	$+1$	$+1$	-1	-1	0.582
E8	$+1$	$+1$	$+1$	$+1$	0.513

(b) Carcinogenesis (B_{max})
$Acc = 0.537 - 0.037\ Minacc$

Fig. 9. Screening results (procedure *ScreenFrac* in Section 3.3). *Acc* refers to the estimated accuracy of the model. The regression model is built using the "Autofit" option provided with [29]. This essentially implements the stepwise regression procedure described in Appendix A. The regression equation in (c) is marked with a "*" to denote that significance levels had to be relaxed from the usual levels of F_{in} and F_{out} (these are discussed in Appendix A: the critical values are changed from 0.05 to 0.1) to obtain a model. This suggests that this model should be treated with caution.

Expt.	Coded Values		Natural Values		Acc
	$Minacc$	$Minpos$	$Minacc$	$Minpos$	
E9	0	0	0.83	8	0.761
E10	−0.50	−0.37	0.79	7	0.808
E11	−1	−0.74	0.75	6	0.798
E12	**−1.50**	**−1.11**	**0.71**	**5**	**0.814**
E13	−2.00	−1.48	0.67	4	0.681
E14	−2.50	−1.85	0.63	3	0.665
E15	−3.00	−2.22	0.60	2	0.665

(a) Mutagenesis (B_{min})

Expt.	Coded Value	Natural Value	Acc
	$Minacc$	$Minacc$	
E9	**0.00**	**0.83**	**0.883**
E10	−0.50	0.79	0.883
E11	−1.0	0.75	0.872
E12	−1.50	0.71	0.872

(b) Mutagenesis (B_{max})

Expt.	Coded Value	Natural Value	Acc
	$Minpos$	$Minpos$	
E9	0	8	0.457
E10	−0.50	7	0.462
E11	−1	6	0.462
E12	−1.50	5	0.490
E13	−2.00	3	0.487
E14	**−2.50**	**2**	**0.510**
E15	−3.00	1	0.510

(a) Carcinogenesis (B_{min})

Expt.	Coded Value	Natural Value	Acc
	$Minacc$	$Minacc$	
E9	0	0.83	0.558
E10	−0.50	0.79	0.581
E11	−1	0.75	0.579
E12	−1.50	0.71	0.567
E13	**−2.00**	**0.67**	**0.605**
E14	−2.50	0.63	0.605
E15	−2.50	0.63	0.596
E16	−3.00	0.59	0.591

(b) Carcinogenesis (B_{max})

Fig. 10. Optimisation using the response surface (procedure $OptimiseRSM$ in Section 3.3). In each case, the response surface used is the first-order regression model found by stepwise regression at the screening stage (shown in Fig. 9). Parameters are varied along the path of steepest ascent. Experiments are stopped once a maximal value for the response variable is followed by three consecutive runs that yield responses that are no higher.

Expt.	Coded Values		Natural Values		*Acc*
	Minacc	*Minpos*	*Minacc*	*Minpos*	
E9	**−1.41**	**−1.41**	**0.72**	**4**	**0.814**
E10	−1.41	−1	0.72	5	0.803
E11	−1.41	0	0.72	8	0.803
E12	−1.41	+1	0.72	10	0.771
E13	−1.41	+1.41	0.72	12	0.782
E14	−1	−1.41	0.75	4	0.798
E15	−1	−1	0.75	5	0.798
E16	−1	0	0.75	8	0.787
E17	−1	+1	0.75	10	0.771
E18	−1	+1.41	0.75	12	0.782
E19	0	−1.41	0.82	4	0.782
E20	0	−1	0.82	5	0.782
E21	0	0	0.82	8	0.755
E22	0	+1	0.82	10	0.771
E23	0	+1.41	0.82	12	0.771
E24	+1	−1.41	0.90	4	0.745
E25	+1	−1	0.90	5	0.723
E26	+1	0	0.90	8	0.628
E27	+1	+1	0.90	10	0.612
E28	+1	+1.41	0.90	12	0.612
E29	+1.41	−1.41	0.93	4	0.745
E30	+1.41	−1	0.93	5	0.729
E31	+1.41	0	0.93	8	0.649
E32	+1.41	+1	0.93	10	0.670
E33	+1.41	+1.41	0.93	12	0.670

(a) Mutagenesis (B_{min})

Expt.	Coded Value	Natural Value	*Acc*
	Minacc	*Minacc*	
E9	−1.41	0.72	0.872
E10	−1	0.75	0.872
E11	**0**	**0.82**	**0.883**
E12	+1	0.90	0.878
E13	+1.41	0.93	0.872

(b) Mutagenesis (B_{max})

Expt.	Coded Value	Natural Value	*Acc*
	Minpos	*Minpos*	
E9	**−1.41**	**4**	**0.490**
E10	−1	5	0.463
E11	0	8	0.457
E12	+1	10	0.457
E13	+1.41	12	0.460

(a) Carcinogenesis (B_{min})

Expt.	Coded Value	Natural Value	*Acc*
	Minacc	*Minacc*	
E9	−1.41	0.72	0.555
E10	**−1**	**0.75**	**0.579**
E11	0	0.83	0.558
E12	+1	0.90	0.546
E13	+1.41	0.93	0.510

(b) Carcinogenesis (B_{max})

Fig. 11. Optimisation by using a multi-level full factorial design (procedure *OptimiseFact* in Section 3.3). In each case, relevant factors are those obtained by screening (Fig. 9). A 5-level full factorial design is then used to find the best values for these factors.

Procedure	(Accuracy,Expts.)			
	Mutagenesis		Carcinogenesis	
	B_{min}	B_{max}	B_{min}	B_{max}
ILP+Default	$(0.755 \pm 0.031, 1)$	$(0.846 \pm 0.026, 1)$	$(0.510 \pm 0.028, 1)$	$(0.504 \pm 0.028, 1)$
ILP+SO	$(0.814 \pm 0.028, 15)$	$(0.883 \pm 0.023, 12)$	$(0.510 \pm 0.028, 15)$	$(0.605 \pm 0.027, 16)$
ILP+SO′	$(0.814 \pm 0.028, 33)$	$(0.883 \pm 0.023, 13)$	$(0.490 \pm 0.028, 13)$	$(0.579 \pm 0.027, 13)$

Fig. 12. Comparison of procedures, based on their estimated accuracies and the number of experiments needed to obtain this estimate. The accuracies are 10-fold cross-validation estimates, for which there is no unbiased estimator of variance [3]. The standard error reported is computed using the approximation in [6].

We now turn to some broader implications of these results, enumerated in order of seriousness to current ILP practice:

1. The results suggest that default levels for factors need not yield optimal models for all problems, or even when the same problem is given different inputs (here, different background knowledge). This means that using ILP systems just based on default values for parameters—the accepted practice at present—can give misleading estimates of the best response possible from the system. This is illustrated in Fig. 13, which shows estimated accuracies on other datasets reported in the literature that also use the Aleph system with default values for all parameters (these datasets have been used widely: for example [22, 19]). Taken with our previous results for the mutagenesis and carcinogenesis data (we will only use the B_{max} results, as these are the results used in the literature), we are now able to make some statements of statistical significance. Fig. 14 shows, across the 8 datasets, differences between the optimised and default models. The probability of obtaining these results, under the hypothesis that the optimised and default procedures have equivalent performance (correctly, that the median difference between their accuracies is 0) is 0.02. In fact, since our research hypothesis is evidently directional (that accuracy of optimised models is higher than that of "default models"), the one-tailed probability of 0.01 is more appropriate. Some readers would perhaps prefer only to rank those instances where the optimised model was substantially higher. If we take "substantially higher" to mean "2 standard errors or more", then the optimised model is substantially higher than the default model in 6 out of the 8 cases (the two mutagenesis datasets are eliminated). The corresponding Wilcoxon probabilites are now 0.05 (two-tailed) and 0.025 (one-tailed). The statistical evidence in favour of the optimised models therefore appears to be significant, perhaps even highly so.
2. The screening results suggest that as inputs change, so can the factors that are relevant (for example, when the background changes from B_{min} to B_{max} in Mutagenesis, *Minpos* ceases to be a relevant factor). Further evidence for this comes from the "DSSTox" dataset (see Fig. 15). This means that

a once-off choice of relevant factors across all possible inputs can lead to sub-optimal performances from the system for some inputs.

3. Screening, as proposed here, still requires identification of an initial set of variables as factors to be varied (here, these were C, $Nodes$, $Minacc$ and $Minpos$). While the set can have any number of elements (all quantitative of course, for the techniques here to be applicable), the choice of these elements remains in the hands of the practitioner using the ILP system. Some element of human expertise of this kind appears unavoidable (and indeed, is even desirable, to prevent pointless experimentation). Additional assistance in the form of including, with each ILP system, a set of potentially sensitive parameters, could be a great help.
4. Optimisation, as proposed here, requires the selection of an appropriate step-size and specification of a stopping criterion for a sequential search conducted along the gradient to the response surface. We have followed the prevalent practice in the field, namely, obtaining the step-size by a process of a binary search over the interval $[0, 1]$; and using a "k-in-a-row" stopping rule (that is, stopping the search if k steps yield no improvement in response). Other techniques exist, and are described in Appendix A.
5. Even if a set of relevant factors are available for a given input, a multi-level full factorial design can be an expensive method to determine appropriate levels. Once done, performance may still be sub-optimal. The results here suggests that experimental studies that use the popular wrapper approach [18] to exhaustively test combinations of different levels of relevant parameters may not yield the best results.

Data	**ILP+Default**	**ILP+SO**
Mut(42)	0.857 ± 0.054	0.857 ± 0.054
Alz (Amine)	0.714 ± 0.017	0.748 ± 0.017
Alz (Tox)	0.792 ± 0.014	0.883 ± 0.011
Alz (Acetyl)	0.527 ± 0.014	0.776 ± 0.011
Alz (Memory)	0.551 ± 0.020	0.671 ± 0.019
DSSTox	0.647 ± 0.020	0.731 ± 0.018

Fig. 13. Estimated accuracies for the Aleph system from some additional datasets used in the literature [22, 19]. The datasets are used in comparative experiments ("System X versus Aleph") that use default settings for all parameters of Aleph. Accuracy estimates for such models are in the column headed "ILP+Default" (although these exact values do not concern us here, we note that differences, if any, to accuracies reported in the literature can be attributed to differences in the cross-validation splits used). The column headed "ILP+SO" are accuracies obtained using Aleph with the **SO** procedure described in the paper. Standard errors are calculated as before. The DSSTox background information differ slightly in [22] and [19] and the models here use the variant from [22].

Data	**ILP+Default**	**ILP+SO**	Δ	Signed Rank
Carcin	0.504	0.605	0.101	+5
Mut (188)	0.846	0.883	0.037	+2
Mut(42)	0.857	0.857	0	–
Alz (Amine)	0.714	0.748	0.034	+1
Alz (Tox)	0.792	0.883	0.091	+4
Alz (Acetyl)	0.527	0.776	0.249	+7
Alz (Memory)	0.551	0.671	0.120	+6
DSSTox	0.647	0.731	0.084	+3

Fig. 14. Absolute differences in accuracy Δ between the procedures **ILP+SO** and **ILP+Default**, and their signed ranks (eliminating ties). The Wilcoxon probability of obtserving the signed ranks under the null hypothesis that median differences are 0, is 0.02 (0.01 for a directional test).

Data	**ILP+Default**	**ILP + SO**
DSSTox [22]	0.647 ± 0.020	0.731 ± 0.018
DSSTox [19]	0.631 ± 0.020	0.684 ± 0.019

Fig. 15. Estimated accuracies for the Aleph system for two variants of the "DSSTox" problem. The datasets in the two variants use slightly different background information, resulting in different accuracies for both default and optimised models. Screening identifies different parameters as being relevant in the two cases: C and $Minacc$ in DSSTox [22] and only C in DSSTox [19].

Finally, a controlled comparison of **Default**, **SO** and **SO′** has required us to enforce that the ILP system used is the same in all experiments. In practice, we are often interested in controlled comparisons of a different kind, namely, the performances of different ILP systems. The results here suggest equipping each ILP system with the procedure **SO** could enable a controlled comparison of best-case performances: a practice which has hitherto not been adopted by empirical ILP studies, but whose value is self-evident. Of course, screening and optimisation experiments would have to be conducted for each system in turn, since the factors relevant to one system (and its levels) would typically have no relation to those of any of the others. We illustrate this in Figs. 16–17. The former shows results of applying the procedure **SO** to a recently proposed ILP system (Toplog) on the datasets we have considered thus far. Parameter screening and optimisation proceeds for a different set of parameters to those used for Aleph: we have used the parameters $Max\_literals\_in\_hypothesis$ (equivalent to the parameter C in the Aleph experiments), $Max\_singletons\_in\_hypothesis$, $Example\_inflation$, and $Minpos$ (which has the same meaning as $Minpos$ in the Aleph experiments). The choice of these parameters was based on their use in data files provided with the Toplog program. It is evident from Fig. 16 that there is an improvement in performance after using **SO** (the overall sum of signed ranks is in favour of Toplog+**SO**) although the differences are not statistically significant. This statistical caveat notwithstanding, Fig. 17 shows the perils of not comparing like-with-like. Fig. 17(a) shows that having subject both Toplog and Aleph to the same procedure for screening and optimisation (that is, **SO**), we find no significant difference in their performance. On the other hand, Fig. 17(b) shows that performing screening and optimisation on one (Aleph), but not the other (Toplog), can lead to misleading results (that the performance of Aleph is significantly better than Toplog).

Data	**Toplog+Default**	**Toplog+SO**	Δ	Signed Rank
Carcin	0.641	0.659	0.018	+2
Mut (188)	0.840	0.878	0.038	+5
Mut(42)	0.881	0.881	0	–
Alz (Amine)	0.704	0.678	−0.026	−3
Alz (Tox)	0.672	0.699	0.027	+4
Alz (Acetyl)	0.640	0.635	0.005	−1
Alz (Memory)	0.526	0.648	0.122	+6
DSSTox	0.618	0.618	0	–

Fig. 16. Absolute differences in accuracy Δ between the procedures **Toplog+SO** and **Toplog+Default**, and their signed ranks (eliminating ties). Once again, we differences, if any, to accuracies reported in the literature can be attributed to differences in the cross-validation splits used. Although the sum of the signed ranks (+13) is in favour of Toplog+**SO**, the evidence is not statistically significant (that is $p > 0.05$)

5 Concluding Remarks

As an ILP system moves from being a prototype for demonstrating a proof-of-concept to being a tool for regular data analysis, it moves into the province of engineering. The requirements of a system in this latter world are significantly more stringent than in the former: robustness is needed, of course, as are mechanisms that facilitate ease of use, recovery from failures, and so on. It also becomes no longer adequate simply to demonstrate that a model *can* be constructed in some novel manner, requiring instead that the model constructed is as good as possible for a given set of inputs (by this we mean primarily the background knowledge and examples). Besides the obvious benefit to the modelling problem being addressed, it ensures that the performance of ILP systems can be assessed in a meaningful manner. Here, we have taken a system engineer's approach to this problem by identifying a set of critical parameters of the system, and then varying these to improve performance. The principal tools we have used are those developed under the umbrella of design and analysis of experiments. Our principal contribution here is to show how these tools can be used to develop better models with ILP systems. To the best of our knowledge, this is the first time any such formal framework has been employed for this purpose in ILP.

Data	**Toplog+SO**	**Aleph+SO**	Δ	Signed Rank
Carcin	0.659	0.605	−0.054	−4
Mut (188)	0.878	0.883	0.005	+5
Mut(42)	0.881	0.857	−0.024	−3
Alz (Amine)	0.678	0.748	0.070	+5
Alz (Tox)	0.699	0.883	0.184	+8
Alz (Acetyl)	0.635	0.776	0.141	+7
Alz (Memory)	0.648	0.671	0.023	+2
DSSTox	0.618	0.731	0.113	+6

(a)

Data	**Toplog+Default**	**Aleph+SO**	Δ	Signed Rank
Carcin	0.641	0.605	0.036	−2
Mut (188)	0.840	0.883	0.043	+3
Mut(42)	0.881	0.857	−0.024	−1
Alz (Amine)	0.704	0.748	0.040	+4
Alz (Tox)	0.672	0.883	0.211	+8
Alz (Acetyl)	0.640	0.776	0.136	+6
Alz (Memory)	0.526	0.671	0.145	+7
DSSTox	0.618	0.731	0.113	+5

(b)

Fig. 17. (a) Absolute differences in accuracy Δ between the procedures **Aleph+SO** and **Toplog+SO**, and their signed ranks (eliminating ties). Although the sum of the signed ranks is in favour of Aleph+**SO** (+22), the evidence is not statistically significant (that is $p > 0.05$). (b) Absolute differences in accuracy Δ between the procedures **Aleph+SO** and **Toplog+Default**, and their signed ranks (eliminating ties). The sum of the signed ranks is in favour of Aleph+**SO** (+30), is now statistically significant ($p = 0.05$ for a non-directional test, $p = 0.025$ for a directional test). This can result in the misleading conclusion that the Aleph system performs significantly better than Toplog on these datasets.

There are a number of ways in which the work here can be extended further. On the conceptual front, we have concentrated on the simplest forms of designed experiments (sometimes called "classical" DOE). Substantial effort has been expended in developing designs other than the fractional factorial designs used here. Response surface optimisation could also involve more complex models than the simple first-order models used here. Both options could yield better results than those obtained here. On the experimental front, our emphasis has been on a controlled study of fractional-factorial screening and response-surface optimisation, using well-studied ILP benchmarks. There are clearly many other datasets studied within ILP that could benefit from utilising the techniques proposed. Finally, it is evident from our results in Fig. 17 that there are wider implications of the results here to the work on the comparative study of ILP systems, and to the development of ILP systems as tools for data analysis. Indeed, nothing restricts the procedures here just to ILP, and the same comments apply to many other machine learning systems. Although outside the scope of this paper, these directions are clearly of some importance, and worth pursuing.

Acknowledgements

The author would like to thank Ravi Kothari was initiating his interest in the area of design and analysis of experiments.

References

1. J. Ashby and R.W. Tennant. Definitive relationships among chemical structure, carcinogenicity and mutagenicity for 301 chemicals tested by the U.S. NTP. *Mutation Research*, 257:229–306, 1991.
2. Y. Bengio. Gradient based optimisation of hyperparameters. *Neural Computation*, 12(8):1889–1900, 2000.
3. Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, 2004.
4. G.E.P. Box and K.B. Wilson. On the Experimental Attainment of Optimum Conditions. *Journal of the Royal Statistical Society, Series B (Methodological)*, 13(1):1–45, 1951.
5. I. Bratko and S.H. Muggleton. Applications of Inductive Logic Programming. *Communications of the ACM*, 38(11):65–70, 1995.
6. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, 1984.
7. R. Bronson and G. Naadimuthu, editors. *Schaum's Outline of Theory and Problems of Operations Research.* McGraw Hill, New York, 1982. (2nd Edition).
8. A.K. Debnath, R.L Lopez de Compadre, G. Debnath, A.J. Schusterman, and C. Hansch. Structure-Activity Relationship of Mutagenic Aromatic and Heteroaromatic Nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry*, 34(2):786 – 797, 1991.
9. L. DeRaedt and M. Bruynooghe. Interactive concept-learning and constructive induction by analogy. *Machine Learning*, 8(2):107–150, 1992.

10. S. Dzeroski. Relational Data Mining Applications: An Overview. In S. Dzeroski and N. Lavrac, editors, *Relational Data Mining*, pages 339–360. Springer, Berlin, 2001.
11. M.C. Fu. Optimization via Simulation. *Annals of Operations Research*, 53:199–248, 1994.
12. L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning.* MIT Press, Cambridge, 2007.
13. D. J. Hand. *Construction and assessment of classification rules.* Wiley, Chichester, 1997.
14. J.C. Helton, J.D. Johnson, C.J. Sallaberry, and C.B. Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10–11):1175–1209, 2006.
15. R. Jain. *The Art of Computer Systems Performance Analysis.* John Wiley, New York, 1991.
16. R.D. King, S.H. Muggleton, A. Srinivasan, and M.J.E. Sternberg. Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proc. of the National Academy of Sciences*, 93:438–442, 1996.
17. R.D. King and A. Srinivasan. Prediction of rodent carcinogenicity bioassays from molecular structure using inductive logic programming. *Environmental Health Perspectives*, 104(5):1031–1040, 1996.
18. R. Kohavi and G.H. John. Automatic Parameter Selection by Minimizing Estimated Error. In A. Prieditis and S. Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, San Francisco, CA, 1995. Morgan Kaufmann.
19. N. Landwehr, A. Passerini, L. De Raedt, and P. Frasconi. kfoil: Learning simple relational kernels. In *AAAI*, pages 389–396, 2006.
20. D.C. Montgomery. *Design and Analysis of Experiments (5th Ed.).* John Wiley, New York, 2005.
21. S. Muggleton. Inverse Entailment and Progol. *New Gen. Comput.*, 13:245–286, 1995.
22. S.H. Muggleton, J.C. Almeida Santos, and A. Tamaddoni-Nezhad. Toplog: Ilp using a logic program declarative bias. In *ICLP*, pages 687–692, 2008.
23. H.G. NNeddermeijer, G.J. van Oortmarssen, N. Piersma, and R.Dekker. A framework for response surface methodology for simulation optimization. In *Winter Simulation Conference*, pages 129–136, 2000.
24. M.H. Safizadeh and R. Signorile. Optimization of Simulation via Quasi-Newton Methods. *ORSA Journal on Computing*, 6(4):388–408, 1994.
25. S. Siegel. *Nonparametric Statistics for the Behavioural Sciences.* McGraw-Hill, New York, 1956.
26. A. Srinivasan. The Aleph Manual. Available at http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/, 1999.
27. A. Srinivasan. Extracting context-sensitive models in Inductive Logic Programming. *Machine Learning*, 44:301–324, 2001.
28. A. Srinivasan. Four Suggestions and a Rule Concerning the Application of ILP. In Nada Lavrac and Saso Dzeroski, editors, *Relational Data Mining*, pages 365–374. Springer-Verlag, Berlin, 2001.
29. D.D. Steppan, J. Werner, and R.P. Yeater. *Essential Regression and Experimental Design for Chemists and Engineers.* 1998. Available at: http://www.jowerner.homepage.t-online.de/download.htm.

30. R.E. Walpole and R.H. Myers. *Probability and Statistics for Engineers and Scientists.* Collier Macmillan, New York, 1978. 2nd Edition.
31. F. Zelezny, A. Srinivasan, and C.D. Page. Lattice-Search Runtime Distributions May Be Heavy-Tailed. In *Proceedings of the Twelfth International Conference on Inductive Logic Programming (ILP2002)*, LNAI, Berlin, 2002. Springer.

A A Note on Linear Regression Models

In this section we provide details of regression models that are of relevance to this paper. All these details can be obtained in any textbook on statistical modelling: we reproduce them here simply for completeness.

Given a response variable y and variables $x_1, x_2, \ldots, x_k$, a regression model expresses a relationship between y and the x_i as follows:

$$y = f(x_1, x_2, \ldots, x_k) + \epsilon$$

where f denotes a systematic functional relationship between y and the x_i, and ϵ denotes random variation in y that is unrelated to the x_i (usually called the *error*). Usually f is specified as some mathematical function (for example, a polynomial in the x_i) and ϵ by a probability density function (PDF). The PDF for ϵ is taken to have mean 0 and standard deviation σ: normally the distribution is also taken to be Gaussian. Thus, in a slightly lop-sided way, for a given set of values for the x_i, it is easier to think of a random value being chosen for ϵ and then constant $f(x_1, \ldots, x_k)$ being added to give the final value of y. From this is evident that y will have a PDF with mean given by $E(y) = E(f(x_1, \ldots, x_k) + \epsilon)$ $= f(x_1, \ldots, x_k) + E(\epsilon) = f(x_1, \ldots, x_k))$; and standard deviation σ. Thus, the regression function effectively specifies the expected, or mean value, of y, given the x_i. "Linear regression" refers to the case when the functional relationship is a linear equation of the form:

$$f(x_1, \ldots, x_k) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

Here, "linear" refers to being linear in the coefficients β_i. So, the following is also a case of linear regression:

$$f(x_1, \ldots, x_k) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + \beta_{k+1} x_1^2 + \cdots + \beta_{2k} x_k^2 + \beta_{2k+1} x_1 x_2 + \cdots$$

To differentiate between these kinds of equation, we denote the former kind which only contain terms $x_1, x_2, \ldots$ as first-order function; and equations of the latter kind which contain quadratic and interaction terms as a second-order function.

In general, assuming we knew the form of f (for example, that it was a first-order function, with errors following a Gaussian distribution with zero mean and variance σ^2), and which of the x_i were functionally related to y, we still need to be able to obtain values of the β_i from a set of observations, or data points, giving values for the relevant x_i and the corresponding values of y. Actually, the best we are able to do is obtain estimates of β_i, which we will denote here as b_i, along with some statistical statement on these estimates. The result is a regression model:

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots$$

Thus, with each data point k, we have an associated "residual" given by difference between the value y_k for that data point, and the value $\hat{y}_k$ obtained from the regression model. The usual approach for obtaining the estimates b_i is the method of least squares, that attempts to minimise the sum of squares of the residuals. The details can be found in any standard statistical textbook (for example, [30]).

We now turn to the first of our assumptions, namely, that of the form of the function. The validity of this assumption can be tested by examining how well the model fits the observed data; and, if used for prediction, estimating how well it will predict response values on new data. The degree of model fit is obtained by examining the residuals and calculating first the statistical significance of model. This tests the null hypothesis $H_0 : b_0 = b_1 = \cdots = b_k = 0$ (that is, there is no linear relationship between y and any of the x_i). Specifically, the quantity:

$$F = \frac{SSR/k}{SSE/(N-1-k)}$$

is calculated, where where SSE refers to the sum of squared residuals $(\sum_{k=1}^{N} (y_k - \hat{y}_k))^2$, N being the number of data points); and SSR is the sum of squares of deviations of the model's response from the mean response $(\sum_{k=1}^{N} (\hat{y} - \overline{y}))^2)$. F is known to follow the F-distribution with $k, N-1-k$ degrees of freedom [30]. So, the hypothesis H_0 can be rejected at some level of significance α, if the F-value obtained is greater than the value tabulated for $F_{\alpha,k,N-1-k}$.

Assuming the null hypothesis is rejected, a quantity that is often used to quantify the degree of fit is the the *coefficient of determination*:

$$R^2 = 1 - \frac{SSE}{SST}$$

where SST is similar to SSR, being the sum of squares of deviations of the observed response from the mean response $(\sum_{k=1}^{N} (y_k - \overline{y}))^2)$. A little arithmetical manipulation will show $SSR + SSE = SST$, and therefore:

$$R^2 = \frac{SSR}{SST}$$

Thus, R^2 is the proportion of the variation in y "explained" by the model. Clearly, $SSR \leq SST$ and therefore $0 \leq R^2 \leq 1$. In general, adding more terms to the regression model will only increase R^2 as the model tends to overfit the data. A quantity that takes overfitting into account is the "adjusted" coefficient of determination:

$$R^2_{adj} = 1 - \frac{N-1}{N-k-1}(1-R^2)$$

If there is a substantial difference between R^2 and R^2_{adj}, then the model is taken to be overfitting the data.

While R^2 or R^2_{adj} denote how well the regression model fits the observed data, it does not have anything to say on the model's performance on new data. An estimate of the predictive power of the model is obtained by performing a resampling exercise by leaving out each of the N data points, and obtaining the corresponding residual based on the model constructed with the remaining $N-1$ points. This is used to calculate a coefficient of determination for prediction R^2_{pred}. Since we will not be using regression models for prediction in this paper, we will not pursue this further here.

Assumptions about the form of the regression model tacitly include assumptions about the errors, namely that they are independent, identically distributed Gaussian variables with zero mean and variance σ^2. The validity of these assumptions are normally checked by visual tests. Graphs of the residual against the predicted response should show no specific pattern; and normal quantile-quantile plots of the residuals should be a straight line [15].

We turn now to the second major assumption, namely that the factors of relevance are known before obtaining the model. This requirement can now be relaxed, since we are able to also test the hypothesis that each of the coefficients b_i are individually equal to zero (the earlier test of significance simply tested that *all* of the b_i were zero: rejection of that hypothesis could still mean *some* of the b_i were zero). This test allows us to eliminate as irrelevant all those factors whose coefficients are not significantly different from zero. In fact, the test forms the basis for a "greedy" procedure that examines the stepwise addition and removal of factors. We reproduce the implementation described in [27] of this procedure in Fig. 18. It is normal to start procedure with $I = \emptyset$. Although it is not guaranteed to find the most relevant subset of factors, and in the worst case, the number of subsets examined can be exponential in $|V|$ the method has been found to work well in practice. Restricted variants of the method are also popular: *forward selection* starts with $I = \emptyset$ and dispenses with the exclusion steps (Steps 6–7 in Fig. 18); *backward elimination* starts with $I = V$ and dispenses with the inclusion steps (Steps 4–5 in in Fig. 18). Both variants examine no more than $O(|V|^2)$ subsets.

B A Note on Constructing and Optimising Response Surfaces

In this section we describe some issues that are relevant to constructing and optimising response surfaces. Specifically, we are concerned with: (1) A procedure for obtaining a fractional experimental design that is suitable for estimating the main effects using the regression procedure described just previously; (2) The search procedure along the gradient to the response surface.

$stepr(V, I, F_{in}, F_{out})$: Given a set of potential regressor variables V (factors in this paper); an initial subset of variables $I \subseteq V$; and minimum values of the F statistic that a variable must achieve to enter (F_{in}) or remain (F_{out}) in the regression equation, returns a subset $S \subseteq V$ identified by a stepwise variable selection procedure.

```
1. $i = 0$
2. $S_i = I$, $V_i = V \setminus I$
3. Increment $i$
4. Let $v_{in}$ be the single best variable in $V_{i-1}$ that can be included (that is, on
   inclusion, gives the greatest increase in the coefficient of determination)
5. If $f(v_{in}|S_{i-1}) \geq F_{in}$ then $S = S_{i-1} \cup \{v_{in}\}$; otherwise $S = S_{i-1}$
6. Let $v_{out}$ be the single best variable in $S$ that can be excluded (that is, on
   exclusion, gives the greatest increase in the coefficient of determination)
7. If $f(v_{out}|S \setminus \{v_{out}\}) \leq F_{out}$ then $S_i = S \setminus \{v_{out}\}$; otherwise $S_i = S$
8. If $S_i = S_{i-1}$ then return $S_i$; otherwise continue
9. $V_i = V \setminus S_i$
10. Go to Step 3
```

Fig. 18. A stepwise variable selection procedure for multiple linear regression (reproduced from [27]). The coefficient of determination (often denoted by R^2) denotes the proportion of total variation in the dependent variable that is explained by the fitted model. Given a model formed with the set of variables X, it is possible to compute the observed change in R^2 due to the addition of some variable v. The probability that the true value of this change is 0 can be obtained from a use of the F statistic [30]. The function $f(v|X)$ returns the value of the F distribution under the null hypothesis that there is no change in R^2 by adding variable v to those in X. The thresholds F_{in} and F_{out} thus specify acceptable probability levels for the inclusion (and exclusion) of variables. It is evident that $F_{in} > F_{out}$ in order to avoid the same variable from repeatedly being included and excluded. A correct implementation of $svs(\ldots)$ also requires sample data and the appropriate regression function to be provided as parameters. We have ignored these here for simplicity.

B.1 Fractional Factorial Designs

We begin by assuming that we have k main effects and that the response surface is approximated by a first-order model with main effects only. That is, we are required to estimate $k+1$ coefficients in a linear model. This requires at least $k+1$ data points, and we simply reproduce a recipe described in [15] that produces a suitable two-level fractional factorial design:

1. Two-level fractional designs are obtained by dividing the full factorial design of k factors by some number 2^p $(1 \leq p < k)$. It is common to refer to such a design as a 2^{k-p} design. Thus, we want to reduce the number of experiments from 2^k to some number 2^{k-p} such that $2^{k-p} \geq (k+1)$. That is, $p = \lfloor k - log(k+1) \rfloor$. Select any $k-p$ factors and construct a two-level full factorial design with these factors. Clearly, this will contain $k-p$ columns (one for each factor). Next, extend this table with columns containing all products of factors. Thus, suppose we initially had $k = 4$ factors $(A, B, C, D$ say), and wanted to construct a 2^{4-1} factorial design (that is $p = 1$). We commence by selecting $k - p = 3$ factors (A, B, C) say, and first construct the following table (this example is from [15]):

Expt.	A	B	C	AB	AC	BC	ABC
E1	-1	-1	-1	$+1$	$+1$	$+1$	-1
E2	-1	-1	$+1$	$+1$	-1	-1	$+1$
E3	-1	$+1$	-1	-1	$+1$	-1	$+1$
E4	-1	$+1$	$+1$	-1	-1	$+1$	-1
E5	$+1$	-1	-1	-1	-1	$+1$	$+1$
E6	$+1$	-1	$+1$	-1	$+1$	-1	-1
E7	$+1$	$+1$	-1	$+1$	-1	-1	-1
E8	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$

   It should be evident that the resulting table will contain $2^{k-p} - 1$ columns.
2. From the $2^{k-p} - 1 - (k-p)$ "product" columns on the right of this table, select p columns and rename them with the p factors not selected in the step above. For example, if we select the ABC column and replace it with D:

Expt.	A	B	C	AB	AC	BC	D
E1	-1	-1	-1	$+1$	$+1$	$+1$	-1
E2	-1	-1	$+1$	$+1$	-1	-1	$+1$
E3	-1	$+1$	-1	-1	$+1$	-1	$+1$
E4	-1	$+1$	$+1$	-1	-1	$+1$	-1
E5	$+1$	-1	-1	-1	-1	$+1$	$+1$
E6	$+1$	-1	$+1$	-1	$+1$	-1	-1
E7	$+1$	$+1$	-1	$+1$	-1	-1	-1
E8	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$

This design will allow us to estimate the main effects A, B, C, D, as well as the interactions AB, AC and BC. However (by construction) it will be impossible to distinguish between the effect of D and that of ABC: the two effects are said to be *confounded* and the terms said to be *aliased*. These are not the only effects that are confounded, and it can be verified that each main effect is confounded with a three-way interaction ($A = BCD$ and so on), and that each two-way interaction is confounded with other two-way interactions ($AC = BD$ and so on). If we are only interested in estimating main effects, then, provided we can assume that three-way interaction effects are negligible, then a table containing just the four A, B, C, D columns above would be adequate. That is, the fractional design is:

Expt.	A	B	C	D
E1	−1	−1	−1	−1
E2	−1	−1	+1	+1
E3	−1	+1	−1	+1
E4	−1	+1	+1	−1
E5	+1	−1	−1	+1
E6	+1	−1	+1	−1
E7	+1	+1	−1	−1
E8	+1	+1	+1	+1

The reader will recognise this as the design used to estimate main effects in the paper. It is clear that the choice of replacing the ABC column with D was an arbitrary one (as indeed, was the choice of A, B, C in the first place): we could, for example, have elected to replace the AB column with D. Thus, there are several 2^{4-1} fractional factorial designs that could have been devised. The difference lies in the assumptions that need to made when estimating main effects: in general, it is considered better to confound main effects with higher order interactions, as these are assumed to be smaller. That is, a design that confounds D with AB will probably yield poorer estimates of the effect of D than one that confounds D with ABC.

Some additional points are in order:

1. The column vectors in the two-level full and fractional factorial designs satisfy some properties: (a) The sum of each column is zero; (b) The sum of products of each column is zero; and (c) The sum of squares of each column is equal to the number of experiments. These properties result in some advantages in computing the main effects: see [15].
2. In a fractional design some factor combination, usually called *identity* and denoted by I, contains 1 in all rows. Such a combination is called the *generator* for the design. For example, $I = ABCD$ is the generator for the design above.

3. Two-level fractional factorial designs are categorised by their *resolution*. The resolution R of a fractional factorial design can be computed as the smallest number of factors that are confounded with the generator I. In the 2^{4-1} design above terms with I is confounded with just one factor combination ($ABCD$). Thus the resolution of the design is 4. Resolutions are normally denoted by Roman numeral subscripts. Thus, the fractional design in Fig. 5 is a 2^{4-1}_{IV} design [20]. In Resolution II designs, main effects are aliased with other main effects. In Resolution III designs, main effects are aliased with with two-factor interactions, and two-factor interactions may be aliased with each other. In Resolution IV designs, main effects are not aliased with each other or with two-factor interactions, but two-factor interactions may be aliased with each other. In Resolution V designs, the only aliasing that occurs is between two- and three-factor interactions, and so on.
4. Two desirable properties relating resolution and linear models with two-level factors (± 1) are those of orthogonality and rotatability. Orthogonal designs result in minimal variance when estimating coefficients, and both full factorial designs and fractional designs in which main effects are not aliased with each other (that is, Resolution III or more) are known to be orthogonal for first-order models [20]. Rotatability concerns variance in prediction across the factor space. Designs that yield predictions whose variance changes symmetrically from the centre of the factor space are said to be rotatable. That is, the variance of prediction at points equidistant from the centre of the factor space should be the same. Once again, full factorial designs and fractional designs of Resolution III or more are rotatable designs for first-order models. Rotatable designs for models with higher order terms $(x_1^2, x_2^2, \ldots)$ will require additional experiments (we will describe these in the following section).
5. In general, if there is a variation in response y even for fixed values of the factors, then we will need to perform several replicates of each experiment, and attempt to model the average response $\overline{y}$. Also, to ensure that there is no dependency in the response variable across experiments, we may need to run the experiments in a randomised order. We will ignore this aspect here, and assume a single replicate for each experiment. One consequence of the latter assumption is that factor levels need to be spread out widely (that is, in two-level experiments, the difference between values corresponding to -1 and $+1$ should be as large as possible), so that effect estimates are reliable (see [20]).

It is evident from from these points that increasing the resolution will allow the construction of models that contain more terms from the full factorial model. Thus, with Resolution III and IV designs, it will only be possible to obtain models that contain the main effects (first-order models). With a Resolution V model, a model with both main effects and two-way interactions can be obtained. Rotatable designs also provide some theoretical guarantees on the estimates, both of coefficients and the response, on these models.

B.2 Gradient Ascent

The primary device used in the paper is to seek local improvements in the response y by making small movements in the direction of the gradient to a response surface. The rationale for gradient ascent can be found in any text on optimization: we present a version here (from [7]) for completeness. Let us suppose that the response surface is given by a scalar field f defined on points that are some subset of $\Re^k$, and whose values $f(x_1, x_2, \ldots, x_k)$ we denote using a vector notation as $f(\mathbf{X})$. We wish to determine a point $\mathbf{x}*$ for which $f(\mathbf{x}*)$ is a (local) maximum.

From the vector calculus, it is known that for any fixed point $\mathbf{x}$ and a unit vector $\mathbf{U}$, the rate of change of $f(\mathbf{X})$ at $\mathbf{x}$ in the direction of $\mathbf{U}$ is given by $\nabla f|_{\mathbf{X}=\mathbf{x}} \cdot \mathbf{U}$, where ∇f is a k-dimensional vector of partial derivatives given by $\left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_k}\right)$ and $\cdot$ denotes the inner, or scalar product of a pair of vectors. For vectors $\mathbf{a}$ and $\mathbf{b}$ the inner product $\mathbf{a} \cdot \mathbf{b}$ is given by $|\mathbf{a}||\mathbf{b}|cos\theta$, where θ is the angle between the vectors $\mathbf{a}$ and $\mathbf{b}$. With some slight abuse of notation, the rate of change of $f(\mathbf{X})$ at $\mathbf{x}$ in the direction of $\mathbf{U}$ is:

$$\nabla f|_{\mathbf{X}=\mathbf{x}} \cdot \mathbf{U} = |\nabla f||\mathbf{U}|cos\theta = |\nabla f|cos\theta$$

The rate of change is therefore greatest when $cos\theta = 1$, or $\theta = 0$. That is, $\mathbf{U}$ is in the same direction of ∇f. Thus, of all non-unit vector displacements of size δ from the point $\mathbf{x}$, the rate of change of $f(\mathbf{x})$ will be greatest for the vector $\delta \nabla f|_{\mathbf{x}}$ (since this vector is clearly along the direction of ∇f). Further, the best value of δ will be the one that maximises $f(\mathbf{x} + \delta \nabla f|_{\mathbf{x}})$.

Search along the gradient In order to use the differential calculus to obtain a value of δ that maximises $f(\mathbf{x} + \delta \nabla f|_{\mathbf{x}})$ in any interval, the function has to be known analytically and the resulting equation for stationary points $f'(\mathbf{x} + \delta \nabla f|_{\mathbf{x}}) = 0$ should be solvable algebraicly. In our case, we do not know the functional form of f: the first-order response surface is simply a local approximation to f that ceases to be appropriate after some value of δ. We therefore have to adopt some form of search for an appropriate value of δ. The simplest of these—and widely used in response surface methods [23]—is the enumerative search we have used in the paper, along with a "k-in-a-row" stopping rule (that is, the search terminates when k steps yield no improvement). Improved versions have been suggested in the literature. The enumerative search could be improved by using better sequential search techniques (for example, a three-point interval search, or a Fibonacci search). In fact, this search itself can be posed as an optimisation problem. In [11] data from experiments performed along the gradient are used to construct a higher order polynomial function of response values in terms of δ. For example, with 3 data points along ∇f obtained from step sizes of $\delta = \delta_1, \delta_2, \delta_3$, and corresponding response values $y = y_1, y_2, y_3$ it will be possible to obtain least-squares estimates for the α_i in $y = \alpha_0 + \alpha_1 \delta + \alpha_2 \delta^2$. The optimal value for δ can then be easily estimated from this function, as $\delta^* = \frac{-a_1}{2a_2}$

(where a_1 and a_2 are the least-squares estimates of α_1 and α_2). We illustrate this in Fig. 19 below, that uses data points from the gradient ascent steps in Fig. 10. The procedure, although not perfect, is reasonably good: the step size estimate (-1.14) results in an actual response value of 0.798 (the regression model predicts 0.819).

Expt.	δ	Acc
E9	0.0	0.761
E10	-0.5	0.806
E12	-1.5	0.814
E15	-3.0	0.665

$$Acc = 0.762 - 0.103\ \delta - 0.045\ \delta^2$$
$$\delta^* = -1.14$$

Fig. 19. Data from steps of the gradient ascent used to estimate a polynomial regression model relating response (Acc) to step-size (δ). The data shown here are from Fig.10(a). The "optimal" value δ^* is obtained using standard techniques from the differential calculus applied to this model.

Other techniques have been proposed as improvements on gradient search, which we do not elaborate further here. We refer the reader to [24] for descriptions and pointers to these.