USING OVERSIZED MODELS TO FIND ACTIVE VARIABLES IN SCREENING EXPERIMENTS

by

Mark Anthony Wolters

B.A.Sc., University of British Columbia, 1996

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the Department
of
Statistics and Actuarial Science

© Mark Anthony Wolters 2007
SIMON FRASER UNIVERSITY
Spring 2007

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.

APPROVAL

Name:	Mark Anthony Wolters
Degree:	Master of Science
Title of project:	Using Oversized Models to Find Active Variables in Screening Experiments
Examining Committee:	Dr. Richard Lockhart Chair

Dr. Derek Bingham
Senior Supervisor
Simon Fraser University

Dr. Randy Sitter
Simon Fraser University

Dr. Tom Loughin
External Examiner
Simon Fraser University

Date Approved: February 28, 2007


SIMON FRASER UNIVERSITY library

DECLARATION OF PARTIAL COPYRIGHT LICENCE

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <www.lib.sfu.ca> at: <http://ir.lib.sfu.ca/handle/1892/112>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada

Revised: Spring 2007

Abstract

Nonregular factorial designs can be used to conduct screening experiments involving many factors and their interactions, using a small number of runs. Linear model selection is challenging in this case because the design is not orthogonal, the number of potential models is huge, and the number of observations is small. A new procedure is proposed to aid model selection in such cases. A non-convergent simulated annealing algorithm is used to generate a large set of good models that are too big; common submodels within this set are then identified using visualization techniques. An automatic method of extracting the best smaller model from the oversized-model set is also proposed. The new method has good performance, and provides graphical output that can be very helpful in decision making. Although developed for industrial screening experiments, it can be applied to any suitable regression problem.

Keywords: linear regression; model selection; nonregular factorial designs; simulated annealing; effect sparsity; effect heredity

Subject Terms: regression analysis; experimental design

Acknowledgments

Pride of place in acknowledgement goes to my parents, Lawrence and Elizabeth Wolters. Their unwavering love and support are behind any of my past, present, or future achievements. Thank you, Mom and Dad, for giving me the opportunities that you yourselves never had.

I would also like to thank the faculty, staff, and students in the Department of Statistics and Actuarial Science at Simon Fraser University, for making my stay there so worthwhile. Particular thanks go to my supervisor, Derek Bingham, for many productive chats; and to my examining committee, Randy Sitter and Tom Loughin, for careful review and thoughtful comments about this document.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the most common goals in statistical work is to identify which variables have an important effect on a process. An investigator may have a large collection of potential predictor variables, and the goal of the study is to find a small number of variables from this collection that, together, have the biggest influence on a response of interest.

Frequently, the investigator will use the standard multiple linear regression model to analyze the data. In this case, the problem of choosing important variables is called *linear model selection* or *subset selection*. It is vital that the model selection step be done well, since all subsequent inferences assume that the chosen model is, in fact, true.

Industrial *screening experiments* constitute one case in which model selection is difficult to do well. This type of experiment involves considering a large number of variables in a small number of experimental runs. The goal is not to build a precise final model, but rather to reduce the pool of possible predictor variables in the most cost-efficient way.

The small number of data points in screening experiments can make the results subject to significant sampling variability, and the large number of variables means there are a huge number of possible models to consider. These features combine to pose a serious challenge to existing model selection methods. The present work proposes improved model selection tools for finding the set of active variables in a screening experiment.

1.1 The Problem: Model Selection

In a broad sense, model selection describes the task of choosing a mathematical representation for the system under study—a ubiquitous task in empirical disciplines. In the present

case, a more restricted definition is used. It is assumed that linear regression will be used to model the response, and that a fixed (though large) pool of candidate predictors has been established. The model selection problem then becomes a question of choosing the subset of predictors that best explains the variation in the response.

Linear Model Selection

One goal of regression is to describe the dependence of a single response variable, Y, on a set of k predictor variables, $\{Z_1, Z_2, \ldots Z_k\}$. This pool of available predictors is assumed to contain all of the variables with significant influence on Y. The multiple regression model posits the following relationship between Y and the Z's:

$$Y = \beta_0 + \beta_1 Z_1 + \beta_2 Z_2 + \cdots + \beta_k Z_k + \epsilon. \tag{1.1}$$

In an experiment, a response will be measured for each of n trials, called *runs*. Let Z_{ij} be the value of the i^{th} predictor in run j. Each run will obey equation (1.1), so the model for the experiment can be written as

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & Z_{11} & Z_{12} & \cdots & Z_{1k} \\ 1 & Z_{21} & Z_{22} & \cdots & Z_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & Z_{n1} & Z_{n2} & \cdots & Z_{nk} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix},$$

or, more compactly:

$$\underset{(n\times 1)}{\boldsymbol{Y}} = \underset{(n\times(k+1))}{\boldsymbol{Z}} \underset{((k+1)\times 1)}{\boldsymbol{\beta}} + \underset{(n\times 1)}{\boldsymbol{\epsilon}}. \tag{1.2}$$

The $(k+1)$ β's in (1.1) and (1.2) are the *true coefficients*, which are unknown. The term β_0 is the *intercept* and accounts for the overall mean value of Y. The vector $\boldsymbol{Y}$ is a random vector of observations; when referring to observed data, it will be denoted $\boldsymbol{y}$. The matrix $\boldsymbol{Z}$ will be called the *full matrix*. Each of its columns represents a particular predictor variable, and its columns, together, represent the entire universe of potential predictor variables. Note that the first column of $\boldsymbol{Z}$ is a column of ones, to account for the intercept term. The column of $\boldsymbol{Z}$ corresponding to the j^{th} variable will be denoted $\boldsymbol{Z}_j$, so that the full matrix is formed by appending these columns onto an intercept column: $\boldsymbol{Z} = [\mathbf{1}|\boldsymbol{Z}_1|\boldsymbol{Z}_2|\cdots|\boldsymbol{Z}_k]$.

The term ϵ represents the error variability or noise in the process. Under the model assumptions, the ϵ_i terms are assumed to be independent and identically distributed normal

random variables, so that ϵ is an n-variate normal random vector: $\epsilon \sim N_n(\mathbf{0}, \sigma^2 \boldsymbol{I})$. The unknown quantity σ is referred to as the *residual standard deviation*, and σ^2 the *error variance*. The set $\{\boldsymbol{\beta}, \sigma^2\}$ constitute the $k+2$ unknown parameters in the model.

The model formed by including the entire list of predictors is known as the *full model*. The general term *model* will be used to describe a particular subset of these predictors. Operationally, a model is formed by selecting a number of columns from the full matrix; these can be combined with an intercept column to form the corresponding *model matrix*. Model matrices will be denoted by $\boldsymbol{M}$. For example, the model formed by including only variables 1, 3, and 7 would be $\boldsymbol{M} = [\mathbf{1}|\boldsymbol{Z}_1|\boldsymbol{Z}_3|\boldsymbol{Z}_7]$.

The *size* of a model is defined as the number of variables in the model, not including the intercept. So the model matrix for a model of size p has dimensions $n \times (p+1)$. There are $\binom{k}{p}$ different models of size p; this number can become very large as the number of variables increases.

Model selection is concerned with the task of choosing a subset of the variables in the full model, to form a smaller model that is best in some sense. If one could determine the true coefficients for the full model, then there would be no need to do model selection. The full model itself would be best, as it would provide the most complete description of the truth. In real applications, however, there are several factors that provide strong motivation to do model selection:

1. It will usually not be possible to estimate all of the coefficients with high precision. In such a case, the full model may have worse predictive or explanatory power than a well-chosen smaller model.

2. Even if good estimation of the full model is possible, many of the predictor variables may have little or no influence on the response. In this case the model may be considered unnecessarily complex. A researcher will usually prefer a small model that is easy to interpret, and contains only the influential variables.

3. The true data–generating process may not conform exactly to the linear model 1.2. In this case the full model may perform worse than a smaller model would.

4. In cases where the number of runs is less than the number of predictors, it is not possible to estimate the full vector of coefficients in the first place. Model selection is a necessity.

For the normal-error regression model, model selection problems may be divided into two types based on the orthogonality or non-orthogonality of the columns of $\boldsymbol{Z}$. If the columns of of $\boldsymbol{Z}$ are all mutually orthogonal ($\boldsymbol{Z}_i^T \boldsymbol{Z}_j = 0$ for all $i \neq j$), then the coefficient of any variable may be estimated independently of the others. This effectively separates the parameter estimation and model selection portions of the analysis. Model selection reduces to a problem of separating the statistically (or practically) significant effects from the insignificant ones.

The situation is considerably different if any of the pairs of $\boldsymbol{Z}$ columns are not orthogonal. In this case some or all of the coefficient estimates will be correlated with each other, and the estimate for a chosen variable will depend on which other variables are included in the candidate model. As a result, model selection and coefficient estimation steps cannot be separated. Model quality must be assessed on a model-by-model basis, necessitating a search through the space of possible models. The present work focuses on problems of this type, particularly cases for which the model search is difficult.

The Case of Screening Experiments

An experiment is called a screening experiment primarily based on its objective: to select the few important variables from the many available ones, with a minimum of runs. This definition suggests that a screening experiment will have a wide full matrix—small n and large k. Small n and large k both make model selection particularly challenging:

- A small number of runs means that the results of the experiment may be subject to high sampling variability. Replications of the experiment might yield $\boldsymbol{y}$-vectors that support different models.
- A large pool of predictor variables means that the number of possible models is huge. A huge model set will worsen the uncertainty about the best model, as there will be more models in competition.
- The huge model set will also make it more difficult to search the space of candidate models. Exhaustive search of all models of size p may not be possible, making it hard even to find the best model.

When the best choice of model is very sensitive to sampling variability, or if there are many models that are almost equally good, *model selection uncertainty* is said to exist.

The characteristics of screening experiments (few data points, huge model sets) make them prone to model selection uncertainty.

1.2 Application Area: Nonregular Factorial Designs

The domain of experimental design covers studies where predictor variables may be set or controlled to pre-specified values in each run. The variables under direct control of the experimenter are usually called *factors*. Traditionally, the ability to set factors to specific values (or *levels*) has been used to ensure that the full matrix for the experiment consists of mutually orthogonal columns. Hence model selection in many designed experiments does not involve model search.

In designs used for screening, however, it is often advantageous to sacrifice orthogonality in order to consider more variables in the same number of runs. Designs of this type fall into the category of *nonregular factorial designs*. Nonregular designs share the property, common in regression cases, of having predictor columns which are neither collinear nor orthogonal. As a result, coefficient estimates are correlated. In the experimental design literature, this situation is called *partial aliasing*[1]; designs with extensive partial aliasing are said to have *complex aliasing*.

A number of types of design exhibit complex aliasing. These include nearly-orthogonal arrays; 3^{k-p} factorial designs with linear-quadratic parametrization; supersaturated designs; and Plackett-Burman designs with interaction terms considered (see Wu and Hamada 2000, chapters 7 and 8 for more detail). In terms of their analysis, such designs bear a strong resemblance to the regression case. Because of complex aliasing, effect estimation is inseparable from model selection.

Plackett-Burman designs are two-level designs based on Hadamard matrices (orthogonal matrices with entries ± 1), with run sizes that are multiples of four but not powers of two. The smallest Plackett-Burman designs therefore have $n = 12$ and $n = 20$. The 12-run design (PB_{12}) can be used to study up to 11 factors in 12 runs, and the 20-run design (PB_{20}) to study up to 19 factors in 20 runs. These two designs will be used as exemplary cases of

[1] More precisely, partial aliasing refers to the existence of bias in factorial effect estimates due to omission of other non-confounded effects from the model. Since the concept of omitted variables has not been considered yet ($\boldsymbol{Z}$ is defined to contain the complete universe of predictors), the concept of partial aliasing has been linked to its root cause—non-orthogonal predictor columns.

screening designs with complex aliasing, for which model selection is difficult. The PB_{12} design is shown in Figure 1.1.

The matrix of factor settings, $\boldsymbol{X}$ in the figure, is also known as the *design matrix*. The columns of the design matrix represent the variables under direct control of the experimenter. These variables, the factors, will also be called *design variables* or *main effects*[2]. Note that, as defined here, the design matrix does not include an intercept column.

In the case of the PB_{12} design, the columns of the design matrix $\boldsymbol{X}$ are mutually orthogonal. In many situations it is necessary to consider not just main effects, but also *two-way interactions*. A two-way interaction is used to capture the joint effect of two factors. Screening experiments usually will assume that interactions involving more than two factors are negligible; so whenever the term *interaction* is used, two-way interactions will be implied.

Interactions are included in the model as the product of two design variables. For example, to include the interaction between X_1 and X_2, an extra column could be added to the model matrix, consisting of the elementwise products of the column vectors $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. Where it will not be ambiguous, such an interaction column may be written using the notation $\boldsymbol{X}_1\boldsymbol{X}_2$. The two main effects that combine to form a particular interaction will be called the *parent* main effects for that interaction.

It is common for the columns of the design matrix to be represented as the letters A, B, C, and so on, as shown in Figure 1.1. An interaction is indicated by pairing the letters associated with the parent main effects; e.g., the model consisting of factors A, B, and their interaction would be written as (A, B, AB). It may also be convenient to refer to factors by their column number in the design matrix; so the same model could be written $(1, 2, 1*2)$.

For a design with w factors, there are $\binom{w}{2}$ interactions. For the PB_{12} case, with $w = 11$, there are 55 interactions, yielding 66 columns plus an intercept in the full matrix. A key feature of this expanded full matrix is that orthogonality is lost once the interaction columns are added. Each main effect is partially aliased with all interactions not containing itself, and each interaction is partially aliased with all variables not containing one of its parents.

[2] A regression approach is followed in this project, so reference will usually be made to coefficients rather than to effects. The term "main effect" is retained as a convenient name for the corresponding *variable*.

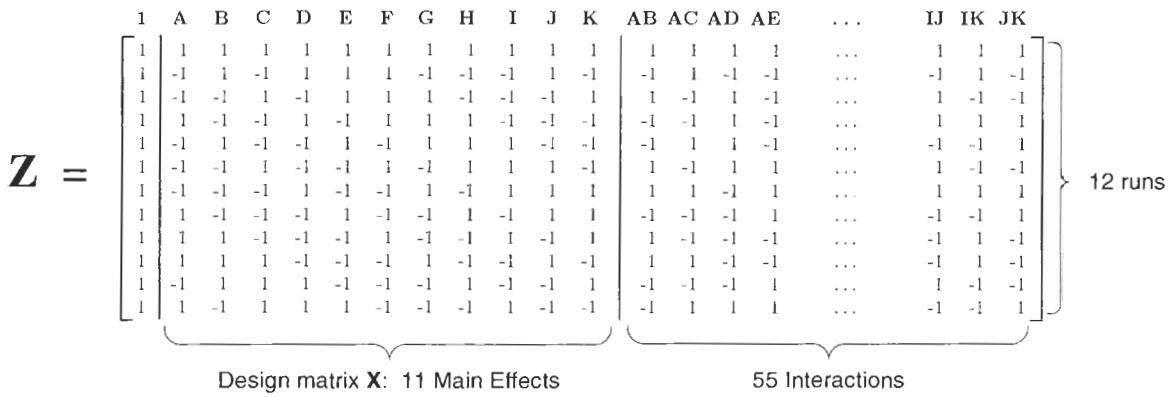

Figure 1.1: Schematic representation of the full matrix for the PB_{12} design.

1.3 Operating Assumptions: Sparsity and Heredity

The analysis of screening experiments can be facilitated considerably by making use of two reasonable assumptions: *effect sparsity* and *effect heredity*.

Effect sparsity is a term for the assumption that only a small fraction of the possible predictor variables are actually important (or *active*).

Effect heredity is a term used to describe the relationships among the main effects and the interactions in a given model. The usage of these terms follows Wu and Hamada (2000): a model respects *strong heredity* if all of its interactions have both of their parent main effects also in the model; it respects *weak heredity* if all of its interactions have at least one of their parent main effects also in the model.

The existence of effect sparsity is a central motivator behind the decision to conduct a screening experiment, where a large number of variables are explored in relatively few runs. The choice to use so few runs (often $n \ll k$) reflects the belief that only a few variables are active. For example, in a 12-run experiment, it is possible to study 11 design variables and their 55 interactions; but with only 12 data points, it is probably not reasonable to entertain models with more than four or five predictors.

The importance of effect heredity as a guiding principle is that it makes precise the intuitive notion of what a sensible model is. In many situations, a researcher would not be satisfied with a model containing interactions that violate heredity. By restricting the model space to only hereditary candidates, a large number of unreasonable models are thrown out

of contention. For example, the models (A, B, AB) and (A, B, BC) respect strong and weak heredity, respectively, while the model (A, B, CD) does not respect heredity at all. For present purposes, effect heredity will always be taken to mean weak heredity. A model that respects weak heredity will be called *hereditary*.

1.4 Summary

The preceding sections have built up the concepts necessary to define the problem in more detail, and to allow a hint at the proposed solution.

Review of the Problem

The defining features of the problem can be briefly summarized. The problem to be solved is model selection (choosing active variables) for screening experiments. The characteristics of screening experiments are a small number of runs (small n), a large pool of candidate predictors (large k), and the assumption of effect sparsity. Small n and large k make screening experiments a challenging model selection problem. There is high model selection uncertainty due to huge model sets and not much data.

The particular application considered will be analysis of nonregular factorial designs that consider main effects and two-way interactions. Weak heredity will be constantly enforced. There will be particular emphasis on Plackett-Burman designs.

Sketch of the Solution

The proposed solution intends to use the characteristics of screening experiments—especially the huge model set and the sparsity assumption—to its advantage. In brief, the method is based on a two step process:

1. Generate a large set of models that are too big, but all have good fit.
2. Find the combination of variables that is most strongly represented among the good large models, and throw out the rest as noise.

The operating principle behind the method is that the true or best model should be contained within most of the well-fitting large models; and that there will be a large number of such oversized good models.

Turning this simple concept into a working method will involve development of a number of tools and algorithms. These will be described in Section 3. Before these tools and algorithms can be made clear, however, a review of model selection concepts is required.

Chapter 2

Model Selection Background By Example

A simulated experiment can be studied to illustrate and expand on the ideas presented in the previous chapter. The example is based on a constructed case, for which the true model is known. An exhaustive search through all candidate models was used, so the exact properties of the model set can be observed. Let the design matrix, $\boldsymbol{X}$, be the PB_{12} design. Let the full matrix, $\boldsymbol{Z}$, be the usual full matrix, formed by adding columns for the intercept and the interactions, as in Figure 1.1. The model is set up as follows:

Example of Chapter 2
Active variables: (A, B, AI).
Model: $\boldsymbol{Y} = \mathbf{1} + 2\boldsymbol{Z}_A + 1.5\boldsymbol{Z}_B + \boldsymbol{Z}_{AI} + \epsilon$.

The residual standard deviation was set to $\sigma = 1$, so that the magnitudes of the true coefficients for A, B, and AI could be considered large, medium, and small, respectively, relative to the noise in the process.

A realization $\boldsymbol{y}$ was generated from this model, and then all models of size 7 or smaller were fit to the data. The design matrix and the observed response are shown in Table 2.1. The expected values for the response are also shown. The observed $\boldsymbol{y}$ differs considerably from its expected value for several runs, especially runs 5, 7, and 8. In this situation some model selection uncertainty is to be expected. The sections that follow discuss model selection concepts and methods using this constructed example as an illustrative case.

Table 2.1: Design matrix for the 12-run Plackett-Burman design, with observed data $\boldsymbol{y}$ and the expected responses.

Run	A	B	C	D	E	F	G	H	I	J	K	$\boldsymbol{y}$	$E[\boldsymbol{Y}]$
1	1	1	1	1	1	1	1	1	1	1	1	5.56	5.5
2	-1	1	-1	1	1	1	-1	-1	-1	1	-1	2.08	1.5
3	-1	-1	1	-1	1	1	1	-1	-1	-1	1	-0.88	-1.5
4	1	-1	-1	1	-1	1	1	1	-1	-1	-1	-0.32	0.5
5	-1	1	-1	-1	1	-1	1	1	1	-1	-1	-2.84	-0.5
6	-1	-1	1	-1	-1	1	-1	1	1	1	-1	-2.51	-3.5
7	-1	-1	-1	1	-1	-1	1	-1	1	1	1	-5.16	-3.5
8	1	-1	-1	-1	1	-1	-1	1	-1	1	1	2.58	0.5
9	1	1	-1	-1	-1	1	-1	-1	1	-1	1	4.72	5.5
10	1	1	1	-1	-1	-1	1	-1	-1	1	-1	3.09	3.5
11	-1	1	1	1	-1	-1	-1	1	-1	-1	1	2.21	1.5
12	1	-1	1	1	1	-1	-1	-1	1	-1	-1	3.47	2.5

2.1 General Concepts

Setting out to find the best model immediately raises the question of what it means for a model to be "best." A model can be best in terms of predictive power, interpretability, or distance from the truth (assuming the nature of the truth can be agreed upon). The appropriate measure of goodness for a model is to some extent situation-dependent and philosophical—see, for example, Royall (1997), Burnham and Anderson (2002), Miller (2002), and Taper and Lele (2004).

For present purposes the following practical (but somewhat evasive) definition will be used: the best model is the one that contains all the practically significant variables and no extra ones. This definition is useful in a screening context, where the chosen model is not likely to be used for prediction or inference. In the screening context, model selection is only a means to the end of choosing active variables. The definition proposed is also the obvious one to use in the current illustrative example, where the true data-generating process is a known linear model.

Before describing the different approaches to finding the best model, several concepts need to be explored. The following sections discuss the types of models in the model set, a way of measuring goodness of fit, and the problems of overfitting and model aliasing.

2.1.1 Structure of the Model Set

The solution space for a model selection problem is the *candidate model set*—the set of all admissible models. This set is finite, but possibly very large. The starting point for building the set is the full matrix, $\boldsymbol{Z}$. Say that $\boldsymbol{Z}$ contains $w > 1$ main effect columns and $\binom{w}{2}$ interaction columns, for a total of k variables plus the intercept. With k predictors and no constraints on admissible variable combinations, the total number of possible models is 2^k. This unconstrained set is typically far too large to work with, and it also contains many models that are not of practical interest.

The notion of effect sparsity can be used to constrain the candidate model set. Based on the assumption of sparsity, only models of size τ or smaller may be considered admissible. Let $\mathcal{M}^*$ represent this set of all models—hereditary or not—with sizes from 1 through τ. Further constraining the admissible variable combinations, let $\mathcal{M}$ represent the subset of models in $\mathcal{M}^*$ that also respect effect heredity. Normally, only hereditary models will be of interest, so $\mathcal{M}$ will be the default model set; but it is useful to consider $\mathcal{M}^*$ for comparison purposes.

The number of models of size p in $\mathcal{M}^*$ is simply the number of combinations of p taken from k, since there are no restrictions on which variable combinations are valid. Let $N_p^*(w)$ represent this number, written as a function of the number of main effects:

$$N_p^*(w) = \binom{w + \binom{w}{2}}{p}. \tag{2.1}$$

The number of models of sizes 1 through τ in $\mathcal{M}^*$ is denoted by $N_{1:\tau}^*(w)$, and given by the sum

$$N_{1:\tau}^*(w) = \sum_{p=1}^{\tau} N_p^*(w). \tag{2.2}$$

For the hereditary model set, $\mathcal{M}$, model counts will be denoted by the same notation, but without the asterisk. For an experiment with w main effects, $N_p(w)$ denotes the number of models of size p, and $N_{1:\tau}(w)$ denotes the number of models of sizes 1 through τ. The equivalent formulas are not as simple, because of the heredity restriction. Derivations of the formulas are given in Appendix A; only the results are presented here. The number of hereditary models of size p is

$$N_p(w) = \sum_{\gamma=1}^{p} \binom{w}{\gamma} \binom{\gamma(w-1) - \binom{\gamma}{2}}{p-\gamma}, \tag{2.3}$$

Table 2.2: Number of models of size p for the 12-run and 20-run Plackett-Burman designs.

	PB_{12}		PB_{20}	
p	**All Models**	**Hereditary**	**All Models**	**Hereditary**
1	66	11	190	19
2	2145	165	1.796×10^4	513
3	4.576×10^4	1705	1.125×10^6	9861
4	7.207×10^5	1.551×10^4	5.260×10^7	1.705×10^5
5	8.937×10^6	1.252×10^5	1.957×10^9	2.680×10^6
6	9.086×10^7	9.026×10^5	6.033×10^{10}	3.857×10^7
7	7.788×10^8	5.894×10^6	1.586×10^{12}	5.147×10^8

so that the total number of models up to size τ is

$$N_{1:\tau}(w) = \sum_{p=1}^{\tau} N_p(w). \tag{2.4}$$

Note that for equation 2.3 to be valid, p must be less than w, and $\binom{x}{y}$ is defined to be zero when $x < y$.

The above formulas are applied to the PB_{12} design, to produce Table 2.2. The table shows that as the model size increases, the total number of models quickly becomes very large. Enforcing heredity helps to keep the size of the model set down; only a small percentage of possible models respect heredity. Despite this, however, even the set of hereditary models can grow very large. Taking models of size 7 in the PB_{20} design as an example, only about 0.03% of the possible models respect weak heredity; but this small fraction still constitutes over 500 million models.

The huge size of the model set has two important influences on model selection:

1. Exhaustive search of the entire model set becomes impractical as the number of design variables (w) or the model size (p) grows.

2. Measures of model goodness are vulnerable, since there is an enormous pool of competing models. The more models there are in competition, the more likely it is that some of them will look good just by chance.

When the true data-generating model is known or assumed, it is useful to partition the candidate model set based on the relationship between the candidates and the truth. Let

the size of the true model be s, and say that a candidate model of size p is chosen through a model selection process. Every variable in the chosen model will be either correct (it occurs in the true model) or wrong (it does not occur in the true model). So the structure of the candidate model can be described by n_c, the number of correct variable choices, and n_w, the number of wrong variable choices. With these two indices, any candidate model can be assigned membership in one of the following five sets of models:

The true model, $\mathcal{T}$. There is only one model in the set for which $n_c = s$ and $n_w = 0$; this is the true model itself.

Overfitted models, $\mathcal{O}$. A model is called overfitted if it has $n_c = s$, but $n_w > 0$. That is, an overfitted model contains the truth plus one or more spurious variable. If $\mathcal{M}$ contains more than one size of models, then $\mathcal{O}$ can be further partitioned. The set of models overfitted by j variables will be called $\mathcal{O}_j$.

Underfitted models, $\mathcal{U}$. A model is underfitted if it has $n_c < s$, and $n_w = 0$. The model contains only truly-active variables, but it is too small—at least one correct variable has been left out. In the example of this section the true model is (A, B, AI), so there are only four heredity-respecting underfitted models: $\mathcal{U} = \{(A),\ (B),\ (A, B),\ (A, AI)\}$[1].

Partial-truth models, $\mathcal{P}$. Any model having $0 < n_c < s$ and $n_w > 0$ is called a partial-truth model. These models contain some, but not all, of the active variables, in addition to one or more inactive variables.

Wrong models, $\mathcal{W}$. A wrong model is composed completely of inactive variables ($n_c = 0$ and $n_w > 0$).

The number of models in each of these sets is shown in Table 2.3, for the example of this section, with $\mathcal{M} = \{$all hereditary models of size 1–7$\}$. The table illustrates the extent of the competition that is present when searching for the best model. It is particularly interesting to note that the set $\mathcal{P}$ contains over 3.7 million models—more than half of the entire model set. Finding models with one or two correct variables, therefore, will not be

[1]The intercept-only model has $n_c = n_w = 0$ and is also an underfitted model as defined here. This model is considered a special case, however.

Table 2.3: Number of models in the different categories, for the example of this section: all hereditary models of size seven or less, PB_{12} design, true model (A, B, AI).

Set		No. Models
$\mathcal{T}$		1
$\mathcal{U}$		4
	$\mathcal{O}_1$ 27	
	$\mathcal{O}_2$ 423	
	$\mathcal{O}_3$ 5013	
	$\mathcal{O}_4$ 48870	
$\mathcal{O}$		54333
$\mathcal{P}$		3738756
$\mathcal{W}$		3145948
$\mathcal{M}$		6939042

hard. It is reasonable to expect that, whatever the particular $\boldsymbol{y}$-vector observed, many of these models in $\mathcal{P}$ will fit the data quite well.

The table also shows that while the number of overfitted models is small as a fraction of the total number of models, the set $\mathcal{O}$ can be large for practical purposes. In this case, there are about 5000 models overfitted by 3 variables, and almost 50000 models overfitted by 4 variables. Assuming the error variance is not too large, all of these overfitted models will have very good fit to the data. The multiplicity of models can make it very difficult to choose one of the models in $\mathcal{O}$ when the true model size is unknown.

The numbers in Table 2.3 were obtained by brute force tabulation using a computer. In subsequent discussions, the number of overfitted models will be of primary interest, and knowing the cardinality of the set $\mathcal{O}$ will be important. To that end, a formula will now be presented for calculating the number of models in $\mathcal{O}_j$.

Consider the case where a candidate model, M, having a main effects and b interactions, is proposed as the true model. Extending the previous notation, define $N_p(w, a, b)$ as the number of hereditary models of size p that contain a *specific* combination of a main effects and b interactions, when the experiment consists of w main effects and their interactions. For example, in the current PB_{12} case, let M be the true model (A, B, AI). Then $N_7(11, 2, 1)$ is the number of models of size 7 that contain the truth: $N_7(11, 2, 1) = |\mathcal{O}_4| = 48870$ as shown in Table 2.3.

The formula for calculating $N_p(w,a,b)$ is (see Appendix A):

$$N_p(w,a,b) = \sum_{\gamma=0}^{p-a-b} \binom{w-a}{\gamma} \binom{a(w-1) - \binom{a}{2} - b + \gamma(w-a-1) - \binom{\gamma}{2}}{p-a-b-\gamma}. \quad (2.5)$$

This equation requires $p > a+b$, $a \geq 1$, and $p < w$. As before, $\binom{x}{y}$ is defined to be zero when $x < y$. When the value of w is clear from context, w may be suppressed from the notation of equations 2.3 and 2.5. So for the PB_{12} example, where it is known that $w = 11$, N_6 would refer to all models of size 6, and $N_6(3,0)$ would refer to all models of size 6 that contain a specific combination of 3 main effects.

2.1.2 Overfitting

A natural goodness-of-fit measure for the candidate models is the residual sum of squares (RSS). The RSS is calculated from standard regression formulas:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T\boldsymbol{y} \\ \hat{\boldsymbol{y}} &= \boldsymbol{M}\hat{\boldsymbol{\beta}} \\ \boldsymbol{e} &= \boldsymbol{y} - \hat{\boldsymbol{y}} \\ RSS &= \boldsymbol{e}^T\boldsymbol{e}, \end{aligned}$$

Where $\boldsymbol{M}$ is the model matrix, $\hat{\boldsymbol{\beta}}$ are the the estimated coefficients, $\hat{\boldsymbol{y}}$ are the fitted values, and $\boldsymbol{e}$ is the vector of residuals. RSS is actually a lack of fit measure, so that smaller RSS indicates a better fit to the data.

The primary problem with RSS from a model selection standpoint is that adding an extra variable to the model can never cause RSS to increase. Introducing one more variable will cause the RSS to get better (smaller) by some amount, and will improve the fit to the observed data; but several trade-offs will be made at the same time. As the model gets larger, predictive power may decrease, model complexity goes up, and the interpretability of the model may be reduced. For these reasons, the best fitting model will not necessarily be the most useful model. In model building, there is a constant problem identifying whether the better fit of a larger model is due to real explanatory power, or just to sampling variability.

The end result of this model size problem is that model selection procedures will often choose a model that is larger than necessary. This outcome is called *overfitting*. In a real data analysis context, when the true model is unknown, one would say the chosen model

overfits the data if the number of variables chosen is too high, leading to poor predictive or explanatory power for the resulting model. In a simulation context, where the truth is known, overfitting means selection of a model that contains the true model plus extra inactive variables—a model in the set $\mathcal{O}$.

The results of the exhaustive search in the PB_{12} example can be used to illustrate some aspects of overfitting. Figure 2.1 shows two different views of the distribution of RSS values over the model set.

The left panel of Figure 2.1 shows the frequency histograms of RSS for models of sizes 5, 6, and 7 (smaller models were not shown because there are too few such models). The dashed vertical line is at 12.05, the RSS of the true model. The true model has a very low RSS for a model of size 3, but as larger and larger models are entertained, more and more models outperform the true model in terms of RSS.

The right panel of the figure illustrates the distribution of RSS values across all model sizes, for models in the sets $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{W}$. As one might expect, models that contain at least some of the truly-active variables (those in $\mathcal{O}$ or $\mathcal{P}$) dominate the low-RSS end of the distribution. For example, there are 481608 models with $RSS < RSS_{truth}$[2]. Of these, only about 16% are totally wrong; 26% contain one correct variable, 46% contain two, and the overfitted models—which contain all three correct variables—constitute the remaining 11% of these top models. The idea that most of the good large models contain some truth will be exploited in Section 3, where the new model selection approach is introduced.

In general, larger models have lower RSS, but there is a distribution of values for models of each size. One approach to resolve the overfitting problem is to penalize the RSS by an amount that increases with model size, thus making the values for different-sized models more comparable. The problem with introducing a penalty is that the best of the larger models, the ones at the left tail of their distribution, will often be very convincing, even in spite of the penalty. Penalty approaches inherently walk a fine line between over- and under-penalizing. If there are many overfitted models, then there is a good chance that a few of them will fit well enough to overcome the penalty. A number of existing model selection criteria can be cast as penalized goodness-of-fit approaches; further discussion of such criteria will be given in Section 2.2.2.

[2]These numbers include only *estimable* models in the model set; there are actually 3960 models of size 7 in $\mathcal{M}$ that have linear dependencies in their model matrices.

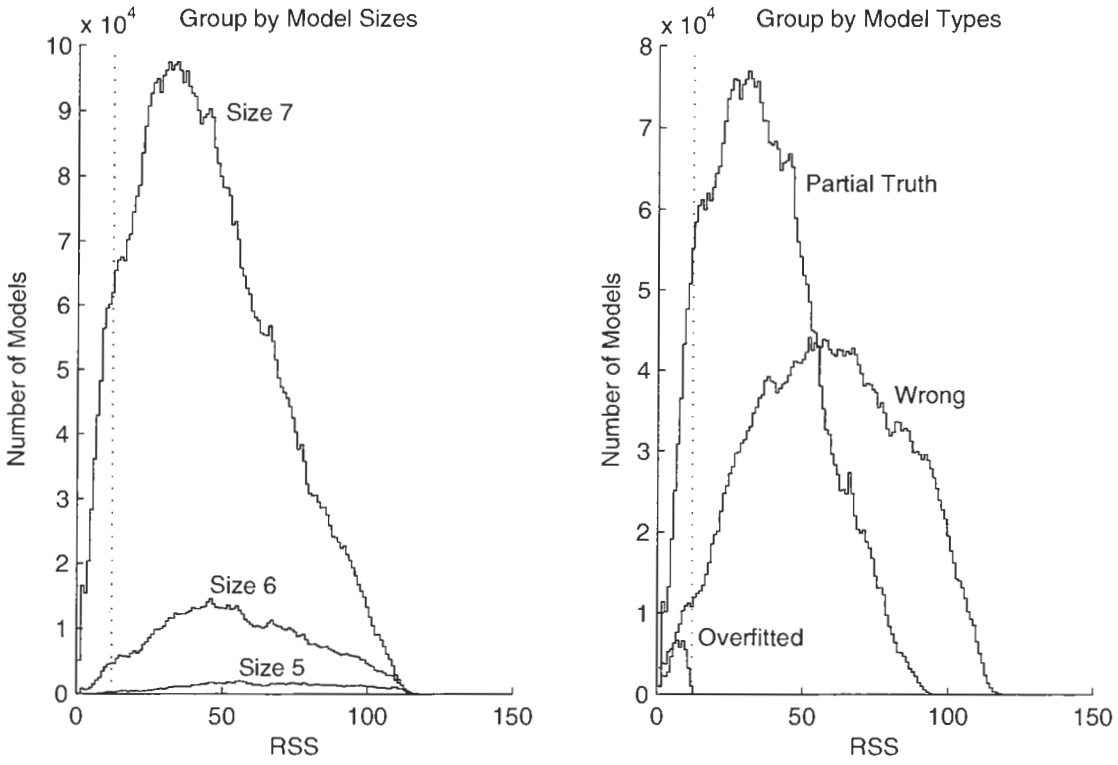

Figure 2.1: Histograms of RSS for the model set of the illustrative example. Left panel: distribution of RSS for models of sizes 5, 6, and 7, showing the increasing number of well-fitting models. Right panel: distribution of RSS for the sets $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{W}$. In both plots, the histogram bins are of unit width, and the dashed line shows the RSS of the true model.

2.1.3 Model Aliasing

Overfitting refers primarily to the problem of deciding among sets of *nested* models. That is, models in $\mathcal{O}_3$ contain the models in $\mathcal{O}_2$, so the problem is deciding whether adding an extra variable causes enough improvement in the model to be worth the extra complexity.

Another type of problem, which will be called *model aliasing*, arises when there are non-nested models that are similarly good at explaining the observed $\boldsymbol{y}$. Model aliasing arises when there is strong competition among models, and the data $\boldsymbol{y}$ does not clearly support just one set of variables. The problem of model aliasing gets worse when the number of candidate models increases, and when the residual standard deviation is higher. The more models to choose from, and the more noise in $\boldsymbol{y}$, the more likely it is that two incompatible models will fit $\boldsymbol{y}$ about equally well.

The extent of model aliasing in this section's example problem can be examined by looking at the expected RSS values for models from different groups. Say the proposed model has size p and model matrix $\boldsymbol{M}_1$. If there are truly-active variables that have been left out from $\boldsymbol{M}_1$, put these in a second matrix $\boldsymbol{M}_2$ so that the combined model matrix $\boldsymbol{M} = [\boldsymbol{M}_1|\boldsymbol{M}_2]$ contains the truth. It can be shown (see, e.g., Rencher (2000), section 7.9) that the expected value of RSS for model $\boldsymbol{M}_1$ is

$$E[RSS] = \sigma^2(n-p-1) + (\boldsymbol{M}_2\boldsymbol{\beta}_2)^T(\boldsymbol{I} - \boldsymbol{H}_1)(\boldsymbol{M}_2\boldsymbol{\beta}_2), \tag{2.6}$$

where $\boldsymbol{\beta}_2$ is the vector of true coefficients corresponding to the variables in $\boldsymbol{M}_2$, $\boldsymbol{I}$ is an $n \times n$ identity matrix, and $\boldsymbol{H}_1 = \boldsymbol{M}_1(\boldsymbol{M}_1^T\boldsymbol{M}_1)^{-1}\boldsymbol{M}_1^T$ is the "hat" matrix for $\boldsymbol{M}_1$.

If the proposed model $\boldsymbol{M}_1$ contains the true model, then the second term in equation 2.6 vanishes and $E[RSS] = \sigma^2(n-p-1)$. If, on the other hand, the proposed model is misspecified, then the expected RSS goes up by an amount that depends both on the proposed model and on the values of the coefficients of the omitted variables.

Equation 2.6 was used to calculate the expected RSS for the overfitted, partial-truth, and wrong models in the illustrative example. The results are shown in Figure 2.2. The figure is analogous to the RSS histograms previously shown in Figure 2.1, but shown in expectation. It is clear that all of the overfitted models, a large number of partial-truth models, and even the left tail of the wrong models, can be expected to have very low RSS values.

In this situation, when many models can be expected to fit the data almost equally well, *model discrimination* becomes a problem. For a fixed truth, model discrimination becomes

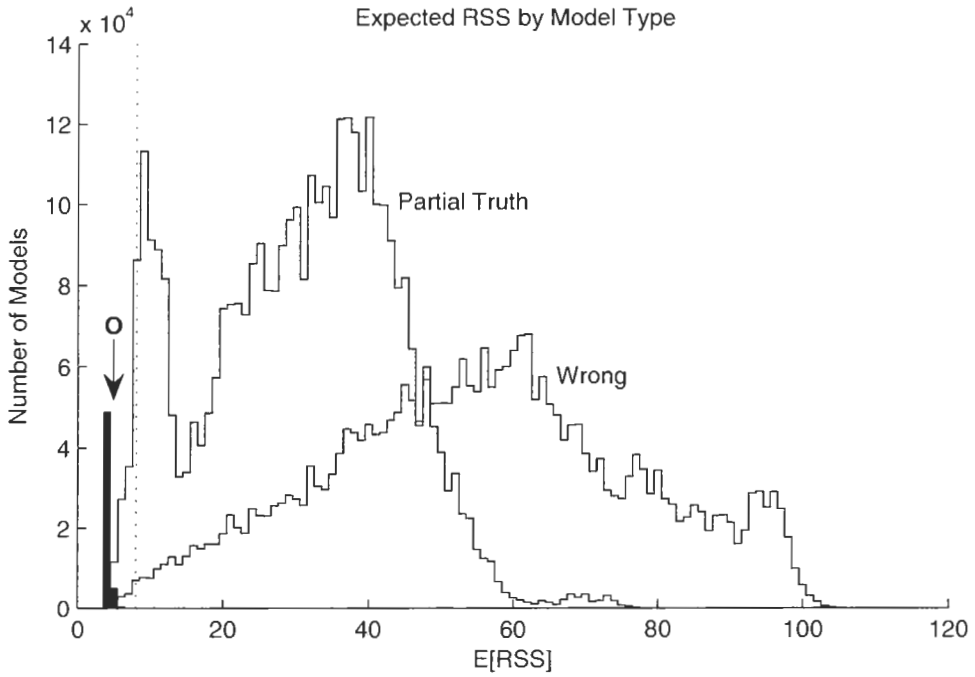

Figure 2.2: Histograms of E[RSS] for the model set of the illustrative example, with true model (A, B, AI). Expected RSS distributions are shown separately for models in $\mathcal{O}$, $\mathcal{P}$, and $\mathcal{W}$. Histogram bins are of unit width, and the dashed line shows the expected RSS of the true model.

more difficult (model aliasing becomes worse) as the number of runs goes down, the residual standard deviation goes up, or the size of the model set increases. The dependence of model aliasing on the size of the candidate model set is significant; common penalty-based model selection criteria do not take the size of the model set into account.

Model aliasing and overfitting both contribute to the general state of *model selection uncertainty* in subset selection problems. When circumstances are such that model selection uncertainty is high, one expects *model selection variability*. That is, on hypothetical repetitions of the experiment, the best model by any criterion will be different most of the time. Equivalently, it is very unlikely that the true best model (however defined) will be the top model in a ranked list.

2.2 Review of Model Selection Methods

Model selection is fundamental to data analysis in many scenarios, and so a large number of methods have been developed to address the problem. The discussion below reviews a number of the most common strategies for choosing a model, with particular emphasis on methods that are used for model selection in linear regression situations. For the analysis of screening experiments like the PB_{12} design, not all of the common methods are feasible. Those methods that will actually work for screening experiments can be divided into three categories: testing-based methods, criterion-based methods, and Bayesian methods. The infeasible methods are discussed in a final section.

There is an extensive literature on identifying significant variables in regular designs, especially unreplicated factorial designs. See, for example, Hamada and Balakrishnan (1998) and Loughin and Noble (1997). The methods usually recommended for regular designs are not of particular interest here, however, because they do not incorporate a model search component. As a result, the methods considered below come largely from regression settings, where nonregular design matrices and complex aliasing are considered the norm.

2.2.1 Testing-Based Methods

The general approach of testing-based methods is to start with an initial proposed model, and then to change the model sequentially by adding or deleting variables, until some stopping criterion is satisfied. At each step, statistical hypothesis testing is used to guide decisions about which variables to add or remove. Model building by inspection, where an investigator tries out different models sequentially, aided by goodness-of-fit tests, falls into this category. When the number of candidate variables is large, more formal implementations of this strategy are commonly used. These automated methods include forward selection, backward elimination, and stepwise regression (as described in, e.g., Miller 2002; Montgomery, Peck, and Vining 2001).

Stepwise regression is a major representative of this family of approaches. The model is built up by successive application of addition and deletion steps, starting from an intercept-only model. In the addition steps, the variable causing the largest decrease in RSS is added, if this decrease is sufficiently large—greater than an F-to-enter cutoff. In the deletion steps, the variable causing the least increase of RSS is deleted, if the increase is sufficiently small—smaller than an F-to-delete cutoff. The process is repeated until no changes can be made.

Pseudocode 1 *Stepwise regression*

Let RSS_p be the RSS of the currently-selected subset, which has p variables. Define RSS_{p-1} to be the smallest RSS obtainable by dropping any one of the variables currently in the model; similarly, define RSS_{p+1} to be the smallest RSS obtainable by adding any of the remaining variables to the model.

Choose "significance levels" P_c and P_d for the addition and deletion steps. Let $F(P, a, b)$ be the $100(1-P)^{th}$ percentile of the F distribution with a and b degrees of freedom.

Start with a model containing only the intercept.

Perform an addition step:

Find the variable that causes the greatest reduction in RSS when added to the model.

Add the variable if $\frac{RSS_p - RSS_{p+1}}{RSS_p/(n-p-2)} > F(P_c, 1, n-p-2)$.

Perform a deletion step:

Find the variable that causes the smallest increase in RSS when removed from the model.

Delete the variable if $\frac{RSS_{p-1} - RSS_p}{RSS_p/(n-p-1)} < F(P_d, 1, n-p-1)$.

Continue alternating addition and deletion steps until the model stops changing. Return this final model as the selected model.

The algorithm is listed in Pseudocode 1.

Testing-based methods like stepwise regression combine the search through the model space with the definition of what a good model is. The criterion for which model is best is defined implicitly in the algorithm—whichever model is finally chosen is declared to be the best by virtue of the fact that it was chosen.

Stepwise regression has been a very popular method, particularly in problems with huge model sets—a model with reasonable properties can be found with only a small amount of computation. The simplicity of implementing stepwise regression is also undoubtedly responsible for its acceptance.

Its popularity notwithstanding, stepwise selection suffers from many well-documented deficiencies, and other methods are generally recommended when they are available. Harrell

(2001) goes so far as to say:

> Stepwise variable selection has been a very popular technique for many years, but if this procedure had just been proposed as a statistical method, it would most likely be rejected because it violates every principle of statistical estimation and hypothesis testing.

Harrell's objections are primarily with the statistical-testing aspects of the procedure. The "F ratios" formed in the addition and deletion steps are not F-distributed, since the variable being added or deleted was specifically chosen for its extreme effect on the RSS. So the P_e and P_d values are essentially tuning parameters. Even ignoring these limitations, however, stepwise regression has two additional problems that are significant in the screening scenario. First, by adding and removing variables in a greedy fashion, the search may miss many good candidate models. Second, it provides no mechanism for ensuring effect heredity in the chosen model. This limitation is particularly serious in the case of screening experiments, where enforcing heredity is usually essential to get useful results.

One stepwise-based method that reduces the problem of non-hereditary models is given in Wu and Hamada (2000). The method, proposed specifically as an analysis strategy for designs with complex aliasing, is given in Pseudocode 2. It consists of a series of applications of stepwise regression, with the admissible set of variables changed at each stage in a way that encourages hereditary models to be found. Note, however, that it is still possible to obtain a non-hereditary model from this method.

The method of Pseudocode 2 will be used as a comparator method in the simulation studies of Chapter 5. The stepwise regression components will be run with $P_e = P_d = 0.05$. Applied to the example problem of this section, the Wu/Hamada method returns model (A, B, AI, BD) as the best model when the model is limited to size four. The truly-active variables are (A, B, AI), so in this case an overfitted model is found. When the model is allowed to be up to size six, a larger overfitted model is found: (A, B, G, AI, BD, DG). Ordinary stepwise regression (Pseudocode 1), starting from a null model, returns the non-hereditary, partial-truth model (A, AI, CE).

2.2.2 Criterion-Based Methods

An alternative to testing-based approaches is to explicitly separate the model search and model criterion segments of the problem. A *model selection criterion* is a metric that is

---

Pseudocode 2 *Comparator method 1: a hybrid stepwise algorithm*

Choose the largest plausible model size, τ. In subsequent stepwise regressions, restrict final models to have size $< \tau$.

Set values for P_e and P_d to be used in the stepwise procedure (default: $P_e = P_d = 0.05$).

For each main effect X_j:

Set the candidate variables to X_j and all of its interactions.

Perform stepwise regression on the candidate variables , with X_j forced into the model.

Select the model from the above set having the lowest RSS. Call this model M.

while stop = false

Set the candidate variables to those in M as well as all main effects.

Set M = result of stepwise regression on the candidate variables.

Set the candidate variables to those in M, plus those two-factor interactions that respect effect heredity.

Set M^* = result of stepwise regression on the candidate variables.

if $M^* = M$ then set stop = true; else set $M = M^*$.

return

Return M as the selected model.

---

used to measure the suitability of a proposed model. The criterion, however defined, is a function of the model structure (the particular set of variables that make up that model), as well as the data. Once a criterion is chosen, model selection becomes an optimization problem: the best model is defined as the model with the optimum value of the criterion.

There are many perspectives on how to define what it means for a model to be best, and as a result there are a large number of possible model selection criteria. These criteria differ in their aims and their theoretical bases, but most of them can be thought of as consisting of two counteracting terms: a goodness-of-fit term and a complexity penalty term. The penalty term attempts to overcome the model size problem discussed in Section 2.1.2, yielding what is essentially a model-size-adjusted goodness-of-fit measure.

Several of the most common selection criteria are listed in Table 2.4, along with a short description. Of these, Mallows's C_p is probably the most commonly used in experimental design circles, while AIC and AIC_c receive the most support in many other fields. All of the criteria can be made to perform well or poorly in specific cases; an analyst's preference for one criterion over another will depend on their definition of what makes a model good, as well as which simulation studies they put the most credence in. It is worth noting that many of these criteria can be cast as equivalent to each other, either asymptotically or upon suitable generalization of their penalty terms.

In the present study, AIC_c will be used as the criterion of choice. This criterion exhibits good general performance in most simulation studies (Hurvich and Tsai 1989; Burnham and Anderson 2002). Note that C_p is not a viable criterion in this case, because in its usual formulation it requires an unbiased estimate of σ^2 from fitting the full model. In many of the screening-experiment cases under study, this requirement cannot be satisfied because there are more variables than data points.

Having chosen an appropriate criterion, there is still the problem of searching for the optimal model. The choice of model search method can be made independently from the choice of criterion. Exhaustive search of all possible subsets is the best-performing method, as it guarantees that the global optimum will be found; but often there will be too many variables for exhaustive search to be economical.

One search method that may be suitable for analysis of screening experiments is *sequential replacement*. The basic sequential replacement algorithm begins with a model of a fixed size, and then attempts to improve on the criterion value of this model by entertaining substitutions of each excluded variable for the included variables. All possible substitutions

Table 2.4: Summary of common model selection criteria, for a model with p predictors plus intercept. Further description, and original references, for all of these methods can be found in Miller (2002).

Crit.	Formula	Comments
R_a^2	$1 - \left(\frac{n-1}{n-p-1}\right)\left(1 - \frac{RSS}{SS_{tot}}\right)$	Adjusted R^2. Larger-the-better. Only increases on addition of a new variable if the residual mean square is reduced.
AIC	$n \log\left(\frac{RSS}{n}\right) + 2(p+2)$	An Information Criterion, or Akaike Information Criterion. Smaller-the-better. An estimate of the expected relative Kullback-Liebler discrepancy between the fitted model and the unknown truth.
AIC_c	$n \log\left(\frac{RSS}{n}\right) + \frac{2n(p+2)}{n-p-3}$	A small sample, bias-corrected version of AIC. Smaller-the-better.
C_p	$\frac{RSS}{\hat{\sigma}^2} - (n - 2p - 2)$	Mallows's C_p. Smaller-the-better. Based on an estimate of the squared prediction error of the observed $\boldsymbol{y}$.
BIC	$-n \log\left(\frac{RSS}{n}\right) - (p+1)\log(n)$	Bayesian Information Criterion. Based on asymptotic equivalence to Bayes factors. Larger-the-better.

are considered for each included variable in sequence, with the best replacement being made at each step. Once all variables in the model have been considered, the process is repeated on the new model until no more improvements are possible. A few variations can be made to the basic algorithm; these are described in Miller (2002).

Sequential replacement is a fast search heuristic, but it will often stop at local optima and miss the global optimum. It also has the disadvantage that effect heredity is not built into the process. For these reasons it will not be considered further here.

A second method that is promising for screening experiments will be called *two-stage search*. Versions of this approach are discussed in Wu and Hamada (2000) and Miller and Sitter (2001). The premise is to perform an initial analysis to discover active main effects only, and then do an exhaustive search over models that contain the chosen main effects. Exhaustive search is feasible because restricting the main effects reduces the size of the model space. The exhaustive search can also be constrained to consider only hereditary models.

In the implementation of two-stage search used here, Lenth's method (Lenth 1989) is

Pseudocode 3 *Comparator method 2: search plus AIC_c*

Part I: two-stage search

Apply Lenth's method to select the active main effects (note: to increase the chance of capturing active effects, do not correct for simultaneous inference).

Perform exhaustive search over all hereditary models that contain the active main effects from the previous step. Save the AIC_c-best model found.

Part II: borrow results from the new method

Evaluate AIC_c for the five most frequent models of each size, as found by the branch-and-bound method in the SAMS algorithm (Section 3.4.2).

Pool all of these most-frequent models with the model found by two-stage search, to form a set of candidates.

Return the model from either method with lowest AIC_c as the chosen model.

used to select the active effects[3]. Any of the extant methods for analysis of unreplicated factorial experiments could be used for this purpose, however.

Applied to the example problem using AIC_c as the criterion, the two-stage algorithm finds model (A, AI)—an underfitted model, missing the active variable B. The main effect B was omitted because it was not deemed significant by Lenth's method in the first stage.

A search-plus-criterion method has been chosen as a comparator method in the simulation studies of Chapter 5. The chosen criterion is AIC_c. An implementation of two-stage search forms the first part of the method; the second part uses results from the new model selection method (to be discussed in the next chapter) to improve the chances of finding the AIC_c-best model. For completeness, this criterion-based comparator method is described here in Pseudocode 3.

The central problem in any criterion-based model selection is choosing the appropriate model size (or alternatively, defining the penalty term appropriately so that the correct-sized model is chosen). As mentioned in Section 2.1.2, the balance between over-penalizing and under-penalizing can never be managed perfectly. Most penalties tend to under-penalize as the number of variables (and hence the model set) gets larger. It should be explicitly noted

[3]The use of Lenth's method makes two-stage search partially testing-based. This approach was used as a criterion-based method regardless, as no better heredity-respecting search heuristics could be found.

that none of the selection criteria incorporate information about the size of the model set, so it is not surprising that larger model sets result in more overfitting.

The overfitting problem can be seen in the example, where exhaustive search was actually performed. The AIC_c values were calculated for two sets of models: i) the complete set of 6939042 hereditary models up to size seven, and ii) the complete set of 720720 models of size four, disregarding heredity. The distribution of AIC_c values for these two sets are shown in Figure 2.3. The vertical dotted line in the figure indicates 20.05, the AIC_c value for the true model.

At first glance, the histograms appear to indicate that the criterion is working well: the AIC_c for the truth is very far out into the lower tail of the distribution. Although the penalty causes the truth to beat out the vast majority of competing models, there are still quite a few models that have lower AIC_c. This illustrates the difficulty using criterion-based methods with huge model sets: even though the criterion may separate the truth from almost all other models, it is very likely that at least a few of the millions of competing larger models will do as well just by chance.

Depending on the extent of model aliasing taking place in a particular case, there is no guarantee that the top AIC_c models will even contain all of the true variables. In the example of the hereditary set, 30 models have better criterion values than the truth; and only seven of these contain all three truly-active variables. Nineteen of the remaining 23 models contain two correct variables and from two to four spurious variables. The last four models are completely wrong—they contain none of the truly-active variables.

Considering now only models of size four, note that there are only 15510 hereditary models, but 705210 models that do not respect heredity. As noted in Figure 2.3, the smaller hereditary set includes only three models that have criterion values better than the true model; but this number jumps to 80 when the heredity restriction is removed. Of these 80, only one contained all three active variables; and 13 contained none of the truly-active variables. This example shows in a particular case how the quality of a penalty or criterion can depend on the set of candidate models.

2.2.3 Bayesian Methods

A Bayesian approach to model selection, suitable for screening experiments, exists (George and McCulloch 1993; Chipman, Hamada, and Wu 1997; Chipman 1998). This method is fundamentally different than testing- or criterion-based methods. Rather than simply

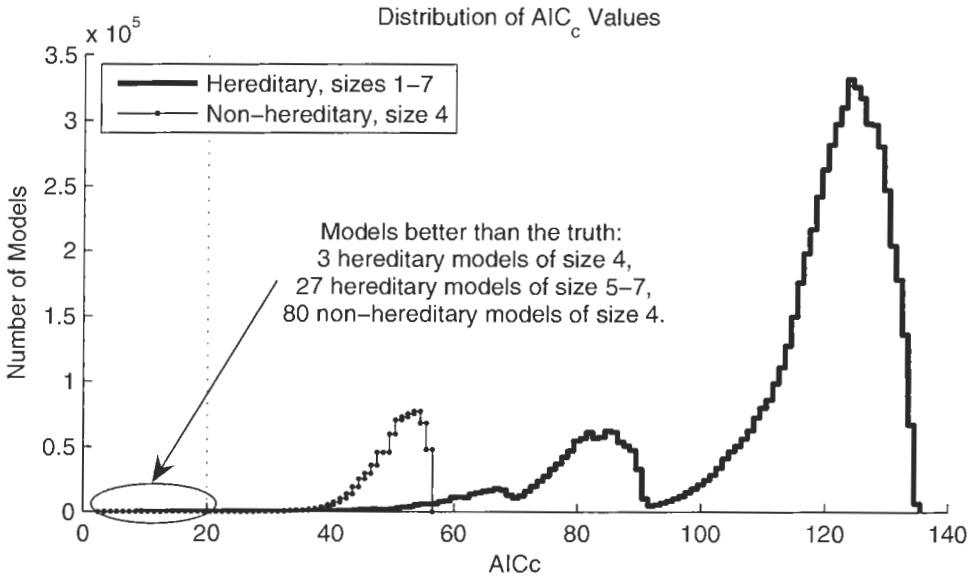

Figure 2.3: Histograms of AIC_c values for the example problem, for exhaustive search through two model sets: the set of hereditary models of sizes 1–7, and the set of all models (hereditary or not) of size 4. Histogram bins are unit width. The dashed line shows the criterion value for the true model.

choosing a best model, the Bayesian methods provide a posterior distribution of models—each candidate model is assigned a probability that can be used to judge the support for that model relative to the other candidates.

The Bayesian variable selection method is briefly described here (following Chipman, Hamada, and Wu 1997), for a case involving k predctors. An unobserved indicator vector $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_k)$ is assumed to exist, with $\delta_i = 1$ if the i^{th} variable is active, $\delta_i = 0$ otherwise. The goal of the procedure is to obtain the posterior distribution of this vector. The data $\boldsymbol{y}$ are assumed to come from the full linear model, $\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{Z}$ is the full matrix and $\boldsymbol{\beta}$ is the vector of all true coefficients. The prior density of each β_i is specified as one of two normal distributions, depending on the value of the corresponding δ_i:

$$f(\beta_i|\delta_i) = \begin{cases} N(0, \tau_i^2) & \text{if } \delta_i = 0 \text{ (effect i inactive)} \\ N(0, (c\tau_i)^2) & \text{if } \delta_i = 1 \text{ (effect i active)} \end{cases}$$

The constant c is chosen to be greater than unity (e.g. $c = 10$) to imply that active variables are expected to have larger coefficients than inactive ones. A prior is also specified for the error variance, usually an inverted gamma distribution based on two parameters,

ν and λ. Heredity relationships are encouraged through the specification of the prior for the vector $\boldsymbol{\delta}$. If δ_{AB} represents the indicator for an interaction effect, and δ_A, δ_B are the indicators for its parent main effects, then priors can be specified using three probabilities:

$$P(\delta_{AB} = 1 | \delta_A, \delta_B) = \begin{cases} p_0 & \text{if } \delta_A = \delta_B = 0 \\ p_1 & \text{if only one of } \delta_A, \delta_B \text{ equals 1} \\ p_2 & \text{if both of } \delta_A, \delta_B \text{ equal 1} \end{cases}$$

The value of p_j gives the prior probability of an interaction being active if it has j active parent effects. Using $p_0 = 0$, for example, enforces weak heredity—it is impossible for an interaction to be declared active unless at least one parent main effect is active as well. The complete specification of the prior for $\boldsymbol{\delta}$ requires a fourth probability, p_m, to be specified for the probability of each main effect being active. The prior density for any value of $\boldsymbol{\delta}$ can then be formed as a product of (p_m, p_0, p_1, p_2) terms.

The posterior distribution of $\boldsymbol{\delta}$ is found by forming the full conditional distributions of $\boldsymbol{\beta}$, $\boldsymbol{\delta}$, and σ^2, and using the Gibbs sampler to make draws from the posterior. Each possible value of $\boldsymbol{\delta}$ represents one candidate model, so the posterior distribution of $\boldsymbol{\delta}$ directly yields a list of highest-probability models.

Bayesian variable selection has some significant advantages over more traditional methods. By presenting a posterior distribution, it allows the investigator to consider several alternative models that may all be well supported by the data, rather than giving the false impression of a single good model. The desire for hereditary models can be smoothly incorporated into the specification of the prior distributions. The question of appropriate model size is less problematic, since the distribution of $\boldsymbol{\delta}$ includes models of all sizes automatically.

The drawbacks to the Bayesian method primarily relate to complexity in running and interpreting the procedure. The machinery of Bayesian inference may be hard for a practitioner to feel comfortable with; good software would probably be required before non-statisticians would feel comfortable with the procedure. The prior parameters (c, τ, ν, λ) represent four tuning parameters—the results may be sensitive to these parameters, so they must be well-chosen. The choice of probabilities (p_m, p_0, p_1, p_2) can also have a strong effect on the solutions returned. Even with good parameter choices, the final interpretation can be difficult based on a table of highest-posterior-density models. For example, should the investigator take the highest-density single model as the best guess at the truth? What about considering the marginal probability of each variable as a basis for choosing active

predictors?

The Bayesian variable selection method is considered an important comparator for the new oversized-model sets method. For the Bayesian method, however, simulation-based comparison is difficult, since it is not clear how to automate the extraction of a single best model. As a compromise, the new method will be compared to the Bayesian approach using examples in Chapter 4.

2.2.4 Other Methods

Many other methods have been proposed to do linear model selection, reflecting the importance and difficulty of the problem. Below, three common types of methods are introduced: cross-validation methods, bootstrap methods, and shrinkage methods. These methods are only very briefly discussed; model selection or regression texts (e.g. Miller 2002; Harrell 2001) are recommended for details. Unfortunately, all of these methods are inappropriate for the screening experiment case, where there may be many more predictors than observations.

Cross-validation (CV) refers to separating the data set into the training and test data sets; the training set is used to fit the models, and the test data set is used to measure the model's predictive performance. CV performance can be considered a model selection criterion. The variable subset giving the smallest CV prediction error is chosen as best. Prediction sum of squares (PRESS) is a common special case of cross-validation, where each single observation is left out in turn, and its value is predicted from the model fit to the remaining $n-1$ observations.

The problem with doing cross-validation with data from screening experiments relates to estimability of the candidate models. The design matrix for a screening experiment will typically be a two-level orthogonal or nearly-orthogonal array. For such a design, removing some rows from the matrix (or even removing only one row, as in PRESS), will render many of the possible model matrices rank-deficient. Cross-validatory methods are therefore better suited to more standard regression cases, where there are more observations than variables and the predictors are at more than two levels.

Bootstrap model selection can be done in two basic ways. In the first approach, the full model is fit and the residuals from this model are resampled to form the bootstrap sample. In the second approach, $(\boldsymbol{x}, y)$ pairs are resampled. The end result in both cases is a bootstrap sample of predictors and responses. For each bootstrap data set, the "best"

model is selected (e.g. by stepwise regression), and its coefficients are estimated. Outputs from the procedure can be model selection frequencies, coefficient estimate distributions, or an estimate of mean prediction error. These outputs can be used to aid selection of the final model.

Bootstrapping residuals is not feasible for most screening experiments that include interactions, because the full model will not be estimable, so no residuals can be formulated. Bootstrapping pairs requires combining randomly-sampled rows of the model matrix; for typical screening experimental designs, this will lead to rank-deficient matrices and an estimability problem similar to the CV case.

The class of shrinkage methods for model selection includes ridge regression, lasso, and penalized likelihood methods (Hastie, Tibshirani, and Friedman 2001; Fan and Li 2001). All of these methods trade off some bias in the coefficient estimates for a variance reduction, and simultaneously do variable selection and coefficient estimation by "shrinking" small coefficients toward zero during the estimation procedure.

Shrinkage methods appear to be most applicable to cases more data-rich than screening experiments. The methods are focused on obtaining good coefficient estimates and good predictive performance, more than on separating active from inactive variables. All shrinkage methods involve a shrinkage parameter, that in many cases is chosen through cross validation. So determining the tuning parameter introduces the same estimability problem found in direct CV model selection.

A final method, that does not fit well into the categories described in this section, is the parallel genetic algorithm (PGA) method of Zhu and Chipman (2006). This recent method employs a genetic algorithm to search through the model space, using any model selection criterion as the objective function. A number of genetic algorithm searches are conducted in parallel, with each search being stopped early, so that a variety of good solutions are found. The output of the PGA searches are combined to identify variables that are most likely to be active. Though it has not been developed specifically for screening experiments (in particular, it does not enforce heredity), the PGA method is very similar in spirit to the new method to be described in the next chapter. Both methods aim to identify many good solutions, and then combine the information among them to give a better picture of the true model.

Chapter 3

Model Selection Using Oversized-Model Sets

The previous chapter illustrated that overfitting and model aliasing are to be expected when considering a huge model set with a small amount of data. Whatever the truth may be, there will be a great many models that contain the truth plus (say) 1–3 spurious variables; all of these models will have good fit, and it will be hard to choose one of them as the best.

The new model selection method, to be introduced below, uses the multiplicity of overfitted models to its advantage. It searches for common variable combinations among a large number of overfitted models.

3.1 Overview of the Method

The first step toward the new method is to define a *maximum plausible size for the truth.* Let this model size be τ. The value of τ indicates the largest number of active factors one would expect to exist, under the assumption of effect sparsity. For example, if analyzing a 12-run Plackett-Burman experiment, the idea of effect sparsity suggests that the maximum number of active variables is about four; otherwise a 12-run design is probably inappropriate in the first place. This suggests that one should set $\tau = 4$ as a reasonable choice.

A key idea in the proposed method is to *only consider models of a single fixed size, about 2 or 3 variables larger than* τ. This large model size will be called p. Any model of size p will be called an *oversized model.* Referring to the PB_{12} example, one would set $p = 6$ or

$p = 7$, since a reasonable value for τ was chosen to be 4. By considering only models of a fixed, large size, the trade-off between goodness-of-fit and model size is avoided. The quality of each model can be evaluated simply by its RSS, and all models will be comparable.

The set of all oversized, hereditary models will be called $\mathcal{M}$. The size of the set can be calculated according to equation 2.3; it will typically be very large. The subset of $\mathcal{M}$ of most interest is the set of overfitted models—the models containing the truth. As before, this set will be called $\mathcal{O}$. Its complement, $\mathcal{O}^c$, contains all of the underfitted, partial-truth, and wrong models. Because the truth is unknown, the analyst does not know which models belong in $\mathcal{O}$, and which belong in $\mathcal{O}^c$. If the membership of $\mathcal{O}$ could be identified, the true model could be found easily—it would be the largest submodel that appears in all models in $\mathcal{O}$.

Not being able to find $\mathcal{O}$ directly, one may instead look at models with good fit to the observed data. Let the *good model set*, $\mathcal{G}$, be the collection of the m lowest-RSS models in $\mathcal{M}$. If the complete model set is small enough, then $\mathcal{G}$ can be found exactly by exhaustive search; if the full set is prohibitively large, an approximate $\mathcal{G}$ can be built through a search heuristic.

The method of oversized models starts with the idea that $\mathcal{G}$ should be a good surrogate for $\mathcal{O}$. It is based on the following logic:

- Because p is chosen to be larger than τ, the overfitted set $\mathcal{O}$ will contain a large number of models.
- Because models in $\mathcal{O}$ contain the truth, they should have very low RSS, and thus they will occur very frequently in $\mathcal{G}$. The other models in $\mathcal{G}$ will be models from $\mathcal{O}^c$ that have good fit by chance.
- Since models in $\mathcal{O}$ dominate the good model set, the true model (which occurs in all the overfitted models) should be discernible as the most over-represented combination of variables in $\mathcal{G}$.

At a high level, then, the algorithm consists of two steps:

1. Search the space of all oversized models; keep the top m of these to form the good model set, $\mathcal{G}$.

2. Examine $\mathcal{G}$ to find a combination of τ or fewer variables that stands out as the main feature or pattern among the good models. Return this set of variables as the selected model.

The method just described can be illustrated on the motivating example of the previous chapter, for which an exhaustive search was performed. Recall that the true model in the example had size three—the model was (A, B, AI). Let $p = 7$ and $m = 5000$, so that $\mathcal{G}$ consists of the five thousand best-fitting models of size seven. In this top set, 1022 models contain the true model, and another 2361 models contain two of the three active variables. So the truly-active variable combinations will occur very frequently in the set $\mathcal{G}$; these combinations can be extracted and used to make decisions about the likely true model.

To become a generally useful algorithm, a number of additional details beyond the brief sketch above need to be worked out. Rules for choosing τ, p, and m need to be established; a general search method for building $\mathcal{G}$ must be developed; and the notion of *most over-represented variable combination* needs to be clarified. These topics will be discussed in the sections that follow.

3.2 Generating The Good Set: Simulated Annealing Model Search

The first major step in the oversized models algorithm is to generate a good model set. Ideally, this set should consist of the m lowest-RSS hereditary models of size p. For small problems, it may be possible to search exhaustively to find the m best models, but in general the number of possible models will be prohibitively large.

When the set of all models is too large to search exhaustively, a search heuristic is required. Finding a well-fitting model is a combinatorial optimization problem, posed as follows: find the subset of p variables taken from k possibilities, such that the RSS is minimized. The familiar technique of stepwise regression is one example of a search algorithm that has been applied to the model selection problem; there are numerous other possibilities available in the model selection and combinatorial optimization fields (Miller 2002; Reeves 1993).

There are two significant complications in the present case, that makes it inappropriate

to choose any of the available combinatorial optimization algorithms as-is. The first complication is precisely that available algorithms are designed to perform optimization—to converge toward one optimum, hopefully the global one. For the purpose of generating a model set, it is rather preferred to *generate a large number of near-optimal solutions*; the goal of the search is not just to find one quality solution, but to find good solutions in quantity. The second problem is the requirement that solutions must respect effect heredity. Putting constraints on the form of solutions typically makes it necessary to modify the standard algorithms.

One of the novel aspects of the present work is the way in which these two complications are resolved. It turns out that the combinatorial optimization heuristic known as simulated annealing (SA) can be readily modified to have the desired properties. Two modifications of the basic SA algorithm have been developed:

1. A hereditary move, to permit search through the model space without stopping on non-hereditary models.

2. An alternative temperature control scheme, to ensure that the search keeps visiting good solutions indefinitely, without converging.

The basic SA algorithm, the required modifications, and the final implementation of the method are described below.

3.2.1 The Generic Simulated Annealing Algorithm

Simulated annealing is a well-established *local search* method for combinatorial optimization. A brief outline of the method is presented here. See, e.g., Reeves (1993) for a more detailed description.

The algorithm starts at one candidate solution, and then entertains the possibility of a *move* to a randomly-chosen *neighbour* solution. The neighbourhood and the possible moves at each step are defined in a problem-specific way, depending on the relationships among the solutions. At each step, a decision is made whether to accept or reject the new solution. If new solution is better than the old one, then the move is always accepted. If the new solution is worse than the old one by an amount δ, then a randomized decision is made: the move is accepted with probability (usually) given by $exp(-\delta/t)$. The tuning parameter t, called the *temperature*, is used to control the convergence rate. At the beginning of the search, t is made large, but as the search progresses, the value of t is decreased.

Pseudocode 4 *A typical simulated annealing algorithm with cooling at every step.*

Let the temperature be t. Let the objective function value at solution s be $f(s)$.

Set the initial value of t. Choose an initial candidate solution, s.

While stop = false:

Choose a new solution, s', from the neighbourhood of s.

Calculate $\delta = f(s') - f(s)$.

if $\delta < 0$
then Accept the move (set $s = s'$).
else Accept the move with probability $P = \exp(-\delta/t)$.
Set $s = s'$ if accepted, leave $s = s$ if not.

Decrease temperature one increment.

If the stopping rule is satisfied, set stop = true.

Return

The purpose of the randomized decision on bad moves is to permit the search to escape local optima. At the beginning of the search, when the temperature is large, many bad moves are made—convergence is sacrificed for a more thorough exploration of the solution space. As the temperature is reduced during the search, fewer and fewer bad moves will be accepted, until ultimately the algorithm converges to what is hopefully the global optimum.

The basic SA algorithm for minimization is given in Pseudocode 4. In the implementation shown, the temperature is reduced at every iteration. Another variety of this algorithm involves reducing the temperature only every j^{th} step.

3.2.2 Modification 1: Hereditary Moves

Simulated annealing is a neighborhood search method; new candidate solutions are generated by *moves* from the current solution to one of its neighbours. The neighbourhood should be defined such that any two solutions can be connected by a finite number of moves.

The goal at this stage is to develop a *hereditary move* that takes a hereditary model of size p, and returns another hereditary model of size p with a small number of different active factors. In this way, the search algorithm can move through the space of candidate models efficiently without stopping on non-hereditary models.

Pseudocode 5 *An algorithm for a hereditary move.*

Let M be the currently selected hereditary model, consisting of p variables.

Choose one of the variables in M, uniformly at random.

Drop the chosen variable from the model. If the chosen variable is a main effect, and removing it from the model will cause one or more interactions to violate heredity, then remove those interactions as well. Call this reduced model M^*.

Add variables to M^* to build it back up to size p, as follows:

Form a list of all variables that would satisfy heredity if added to M^*. This list should not include those variables deleted in previous steps.

Choose one variable at random from this list and add it to M^*.

Repeat until M^* has size p.

The newly-developed hereditary move is shown in Pseudocode 5. The move is simple and effective: one variable from the current model is selected at random; this variable is removed from the model; any other variables that would violate heredity are also removed; and then the model is built back up to size p by adding admissible variables.

Defining a hereditary move in this way implicitly defines the neighbourhood of a hereditary model as any model that may be reached in one move. Note that in the variable-dropping step, it is possible that more than one (or even all) of the variables may be removed. Therefore the number of possible moves depends on the number of main effects in the current model.

For example, if the current model is $M = (A, B, C, AB, AH)$, then dropping out variable A will leave $M^* = (B, C, AB)$. Interaction AB is retained because it still has a parent main effect (B) in the model; conversely, interaction AH is dropped out because it would violate heredity to retain AH after A has been dropped.

Table 3.1 provides a more extensive example, giving a sequence of 12 hereditary moves from a starting model of (A, B, C, AK). Line numbers 10 and 11 are particularly interesting in this example—they are cases of a model having only one main effect and $p-1$ interactions. In such a case, the neighbourhood of the model is actually the entire model set. If the main effect happens to be selected for removal, then all the interactions will be removed as well; when the model is built back up to size p, any model could result. This happens in row 11

Table 3.1: An example showing 12 hereditary moves for a PB_{12} design, with main effects A-K. The initial model is (A, B, C, AK). Boxes indicate which variables were selected to be dropped at each move. Underlined variables were removed to maintain effect heredity.

Move	Model			
0	A	B	C	AK
1	A	B	C	CD
2	A	B	AC	BK
3	A	B	AC	BE
4	A	B	AG	BE
5	A	B	BC	BE
6	B	C	BC	BE
7	B	C	K	BC
8	A	B	C	BC
9	B	C	AC	BC
10	C	AC	BC	CJ
11	C	AC	BC	CI
12	B	H	J	BD

of the table.

The nature of the move defined here is quite useful for the purpose at hand. For the majority of moves, the new model will differ from the previous one by only one or two variables—this is necessary for the algorithm to seek out good solutions. Occasionally, however, the new and previous models will be very different, possibly totally different—this will help ensure good coverage of the model space, and consequently make it easier to find a variety of good solutions.

3.2.3 Modification 2: Preventing Convergence

Having defined a useful move to permit searching of the hereditary model space, the basic SA algorithm must still be modified to prevent convergence, so that the search will produce a large set of good models as required. Two methods are proposed to achieve the desired behaviour:

1. Change the temperature control scheme to introduce occasional temperature *increases*. If the temperature is never allowed to get too small, then there will always be an

appreciable probability of accepting a poor move, and the search can be made to carry on indefinitely.

2. Set a lower bound on the probability of acceptance, so that any move, no matter how bad, will always have a small chance of being accepted.

The second strategy will have a similar effect to introducing several random restarts throughout the search. Its purpose is to provide further assurance that the model space is well-explored. Implementation of this strategy is discussed in the next section; here the focus will be on the first point—temperature control.

The literature on simulated annealing contains a variety of suggestions for adaptive control of temperature, some of which include temperature increases (Reeves 1993). The desired behaviour in the present case is to have the search remain in the neighbourhood of good solutions long enough to visit many good options, without ever getting trapped in a particular neighbourhood. Using a variation on an idea in Dowsland (1993), this is achieved by *cooling the system on every accepted move, and heating the system on every rejected move.* The amount of cooling is controlled by a constant ρ, $0 < \rho < 1$; every time a move is accepted, the temperature is set to $t = \rho t$. The amount of heating is controlled by a constant α, $\alpha > 1$; every time a move is rejected, the temperature is set to $t = \alpha t$.

When the SA search is at a good solution, there are not many improving moves, and there will be many rejections. Making the system hotter with each rejection, as proposed here, ensures that the search must eventually make an uphill move as the rejections keep accumulating. In this way convergence is guaranteed not to happen. The non-convergence tendency is counterbalanced by cooling the system on each accepted move; doing so tends to keep the search moving in a good direction while moves are still being accepted reasonably often.

The relative magnitudes of ρ and α determine the balance between searching good neighbourhoods and jumping away to new parts of the model space. Making ρ smaller relative to α puts more pressure on the algorithm to search good neighbourhoods—many rejections (heating steps) will be required to balance one acceptance (cooling step).

Based on this interpretation, it is convenient to define the temperature control not by specifying ρ and α directly, but rather by specifying ρ and a third parameter, κ, as follows:

1. Choose a value for κ. κ is defined as the number of heating iterations required to balance one cooling iteration. The larger κ is, the more time the algorithm will spend

near local optima. Equivalently, κ may be thought of as the desired ratio of rejected moves to accepted moves.

2. Choose the cooling fraction, ρ, to be some appropriate value, such as 0.95 or 0.99.

3. Calculate α from ρ and κ. The definition of κ implies that $\rho\alpha^\kappa = 1$, and hence $\alpha = (1/\rho)^{1/\kappa}$.

The temperature control scheme is illustrated for a hypothetical sequence of moves in Figure 3.1. In this figure, the search starts in some initial state with $t = 1$; the search then undergoes two accepted moves, followed by eight rejected ones. The curves in the figure show the acceptance probability $P_{acc} = \exp(-\delta/t)$ used to make the randomized decision on a move that increases the RSS by an amount δ. The curve gets shifted to the left on accepted moves, making it harder to accept bad moves in the future; it gets shifted to the right on rejected moves, having the converse effect. The shifts to the right are smaller than those to the left, so it takes κ right-shifts (rejections) to balance one accepted move.

3.2.4 The Modified Simulated Annealing Algorithm

The hereditary move and the adaptive temperature control can be incorporated into the generic simulated annealing algorithm to yield a search heuristic with the desired properties—a non-convergent search through good hereditary models. The complete algorithm is given in Pseudocode 6. The main points will now be described in more detail.

Before beginning the search, a number of control parameters must be set. The most important of these is p, the size of the oversized models. As previously discussed, the goal is to choose p so that it is two or three variables larger than τ, the maximum plausible size of the truth. Ideally, the value of τ could be established based on subject-matter considerations. For cases where no guidance is available to help choose τ, the following rule of thumb is proposed: *the assumption of effect sparsity implies that the truth is no larger than about* $n/3$. Adding two variables to this number gives the rule of thumb for choosing p: $p \approx n/3 + 2$ (p must, of course, be an integer).

Temperature control is managed by setting values for the two parameters ρ and κ. There are various combinations of these inputs that will give good search behavior. The recommended default is to set the cooling fraction, ρ, to some value close to 1 (say, 0.95) so that cooling does not happen too fast; and to set κ to 4 so that only four heating iterations

Move	t	Outcome
0	1.000	A
1	0.500	A
2	0.250	R
3	0.297	R
4	0.354	R
5	0.420	R
6	0.500	R
7	0.595	R
8	0.707	R
9	0.841	R
10	1.000	-

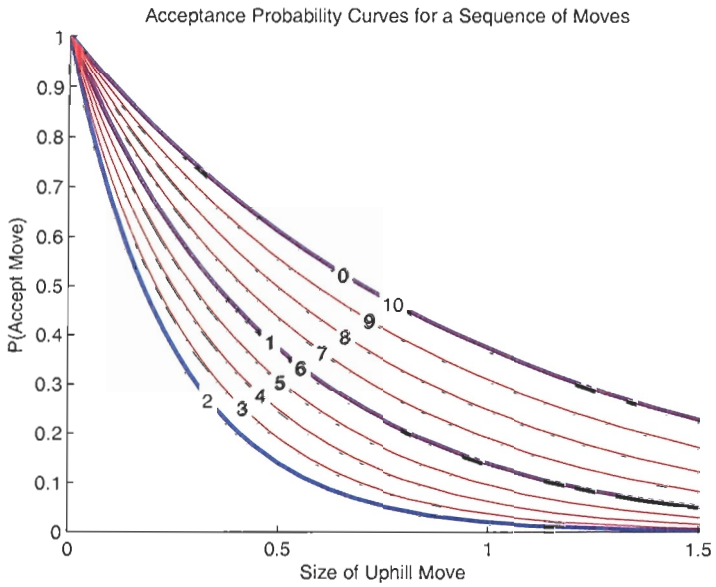

Figure 3.1: Illustration of the temperature control scheme. This example has an initial temperature $t = 1$, and parameters $\rho = 0.5$ and $\kappa = 4$; it shows two accepted moves (thick blue curves) followed by eight rejections (thin red curves). The curves show the randomized decision rule from the initial state (curve 0) to the last move (curve 10). The table shows the temperature at each step, and the outcome of the move: accept (A) or reject (R). Note that κ rejections are needed to balance one acceptance.

are required to balance one cooling iteration. This combination allows the algorithm to accept many good solutions, without rejecting such a high fraction of moves that the search is too slow.

At the beginning of the search, the good model set $\mathcal{G}$ is initialized to an empty set. The objective of the search is to populate $\mathcal{G}$ with a large number of well-fitting models. This is done by running the modified simulated annealing algorithm until a large number of accepted moves have been generated. As a default this number, n_{gen}, will be set to 10000. The modified SA algorithm may accept the same model multiple times throughout the search, so after n_{gen} accepted moves, only the m unique models are retained in $\mathcal{G}$.

The algorithm is insensitive to the initial temperature setting, and to the starting solution fed to the search. In the implementation used, the initial temperature is set to one tenth of the maximum RSS, that is, to $t = (n-1)\text{var}(Y)/10$. The initial model is chosen randomly from the set of hereditary models. Subsequent candidate models are always generated using the hereditary move described in Section 3.2.2.

The annealing algorithm itself proceeds in the same way as the generic SA algorithm, but with two key differences. The first difference is the temperature control: each time a move is accepted, the temperature is reduced by setting $t = \rho t$; each time a move is rejected, it is increased by setting $t = \alpha t$.

The second difference is a small change to the randomized decision made when the proposed move is to a model with higher RSS. In the standard SA method, a move that increases the RSS by δ units is accepted with probability $\exp(-\delta/t)$, which means that the probability of acceptance vanishes as δ gets large. In the modified algorithm, the acceptance probability is given a lower bound, so that $P_{acc} = \max(\exp(-\delta/t), P_{min})$. P_{min} is set to a default of 0.01. Putting a lower bound on the acceptance probability provides further assurance that convergence cannot occur, and helps keep the search spread more widely over the solution space.

The result of the model search is a set $\mathcal{G}$, containing m unique models. At the end of the search, $\mathcal{G}$ is sorted in ascending order of RSS, so that the best models are listed first. The majority of the models in $\mathcal{G}$ will have very good fit (low RSS). There will be a fairly wide range of RSS values in $\mathcal{G}$, however, because the search accepts some poorly-fitting models in order to cover the model space. This is not a concern, since the poor models should contain a heterogeneous mixture of variables; the pattern of true variables in the overfitted models in $\mathcal{G}$ should still be evident despite this noise. Methods for detecting and extracting the

truly-active variables are discussed next.

3.3 Visualization of the Oversized-Model Set

Two key graphics are proposed to make interpretation of the good model set easier. The first, a raster plot, graphically shows the coefficients of all models in $\mathcal{G}$ at once, so that common patterns in the model set can be seen easily. These patterns may also be made more evident by performing an optional clustering step before plotting. The second plot, a link diagram, shows information on the occurrence frequencies of different variable pairs in $\mathcal{G}$. These two plots can be used together to infer much about the data-generating process.

The graphical methods will be shown for two cases, to illustrate the typical spectrum of outcomes that may arise. The two cases are based on the same data-generating process, but with two different observed $\boldsymbol{y}$ vectors. The PB_{12} design is used, with design matrix $\boldsymbol{X}$. The true model is set to $(2, 5, 2{*}7, 5{*}9)$.

Case one
Model: $E[Y] = 0.5 - 1.1\boldsymbol{X}_2 - 1.5\boldsymbol{X}_5 + 0.9\boldsymbol{X}_2\boldsymbol{X}_7 + \boldsymbol{X}_5\boldsymbol{X}_9$.
Residual standard deviation: $\sigma^2 = 0.25$.
Response: $\boldsymbol{y}$-vector randomly generated from the model.
Case two
Model: $E[Y] = 0.5 - 1.1\boldsymbol{X}_2 - 1.5\boldsymbol{X}_5 + 0.9\boldsymbol{X}_2\boldsymbol{X}_7 + \boldsymbol{X}_5\boldsymbol{X}_9$.
Residual standard deviation: $\sigma^2 = 0.75$.
Response: generated $\boldsymbol{y}$'s and chose one with high model selection uncertainty.

The two cases use the same model with the same coefficients, but differ in their error variance. For case one, models containing the truth should be considerably better than other models. For case two, the model selection process should be much more subject to sampling variability—replications of the experiment would yield a range of best models.

3.3.1 Raster Plot

The starting point for the raster plot is a matrix representation of the good model set. In the computer implementation, $\mathcal{G}$ is represented by an $m \times k$ matrix, with rows representing models, and columns representing variables. The j^{th} row of the matrix contains the coefficients of the j^{th} best-fitting model; coefficients of variables not in the j^{th} model are set to

Pseudocode 6 *The modified SA algorithm: Simulated Annealing Model Search*

Choose p, the size of the models to generate. Rule of thumb: $p \approx n/3 + 2$.

Set input parameter values: n_{gen} = number of accepted models to generate; ρ = cooling fraction; κ = acceptance/rejection ratio; P_{min} = minimum acceptance probability (defaults: $n_{gen} = 10000$, $\rho = 0.95$, $\kappa = 4$, $P_{min} = 0.01$). Calculate heating factor, $\alpha = (1/\rho)^{1/\kappa}$.

Initialize counter for number of accepted moves: $n_{acc} = 0$. Initialize the temperature: $t = (n-1)\text{var}(Y)/10$.

Initialize the good model set, $\mathcal{G}$, to the empty set.

Choose an initial hereditary model, M_{old}, at random. Set $n_{acc} = 0$.

While $n_{acc} < n_{gen}$:

Use a hereditary move (see Pseudocode 5) from M_{old} to choose a new candidate model, M_{new}. M_{new} must be estimable.

Calculate difference in RSS: $\delta = RSS(M_{new}) - RSS(M_{old})$.

If $\delta < 0$ (new model is better) then

Add M_{new} to $\mathcal{G}$.

Set $M_{old} = M_{new}$.

Set $n_{acc} = n_{acc} + 1$.

Set $t = \rho t$.

Else (new model is worse)

Calculate acceptance probability: $P_{acc} = \max(\exp(-\delta/t), P_{min})$.

If rand() $< P_{acc}$ (accept randomized decision) then

Add M_{new} to $\mathcal{G}$.

Set $M_{old} = M_{new}$.

Set $n_{acc} = n_{acc} + 1$.

Set $t = \rho t$.

Else (reject randomized decision)

Set $t = \alpha t$.

End if

End if

Return.

Discard all duplicate models in $\mathcal{G}$, and sort $\mathcal{G}$ in ascending order of RSS.

zero. The result is a sparse matrix with nonzeros at the positions of active variables.

The raster plot is one way to visualize the information in the $\mathcal{G}$ matrix. The matrix is plotted as an image, with one rectangular pixel per element. Zero-valued coefficients are drawn as white, negative coefficients are drawn as red, and positive coefficients are drawn in blue. The intensity of the red and blue is scaled for each model, so that the coefficient with the largest magnitude in each model appears as fully saturated red or blue (see Figure 3.8 for the color scaling).

Figures 3.2 and 3.3 provide examples of the raster plots drawn for the example cases 1 and 2, respectively. About 5200 unique models were found in case 1, and over 7000 in case 2. The raster plots permit all of these top models to be summarized simultaneously, and make it easy to see commonalities across models. In the raster plot, the predictor variables are shown by their column number in the full matrix. The intercept column is not shown. An additional plot at right gives the RSS values for all of the displayed models, to help gauge the relative goodness of fit among the model set.

A significant amount of information can be gathered from reviewing the raster plot. The variable occurrence frequencies are the most obvious thing to notice in the plot. Variables with a strong effect on the response will appear in most of the models, creating a vertical line on the plot. The relative magnitudes of coefficients can also be observed; small coefficients have a lighter shade. Coefficient signs may also be observed across models, which may aid interpretation in some cases. If a variable has an even mixture of positive and negative coefficients across many different models, for example, one would likely question the validity of that variable's effect.

Another very useful aspect of the raster plot is that it also helps the analyst to understand the extent of model confounding or model selection uncertainty in a particular case. If one model dominates the entire raster plot, then this model can be selected with confidence; but if there is no clear winner in the model set, then this will also be very clear from the graph. The investigator will not be mistakenly led to believe that the best choice of model is unambiguous. This is particularly important in screening experiments, where it is important to make good decisions about which follow-up trials are appropriate.

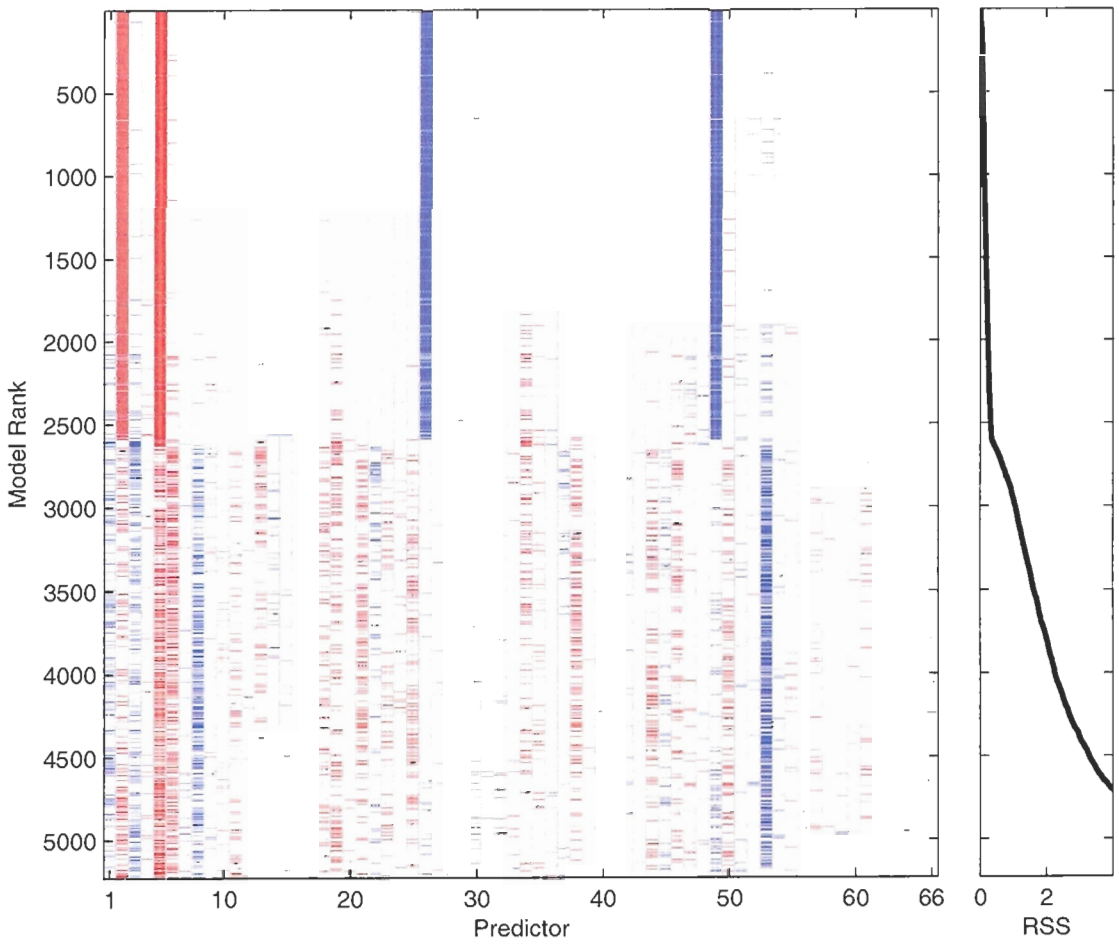

Figure 3.2: Example of a raster plot of the good model set for case 1. The true model $(2, 5, 26, 49)$ dominates the top half of the set and is clearly distinguishable. The remaining models comprise a noisy mixture of many different variables.

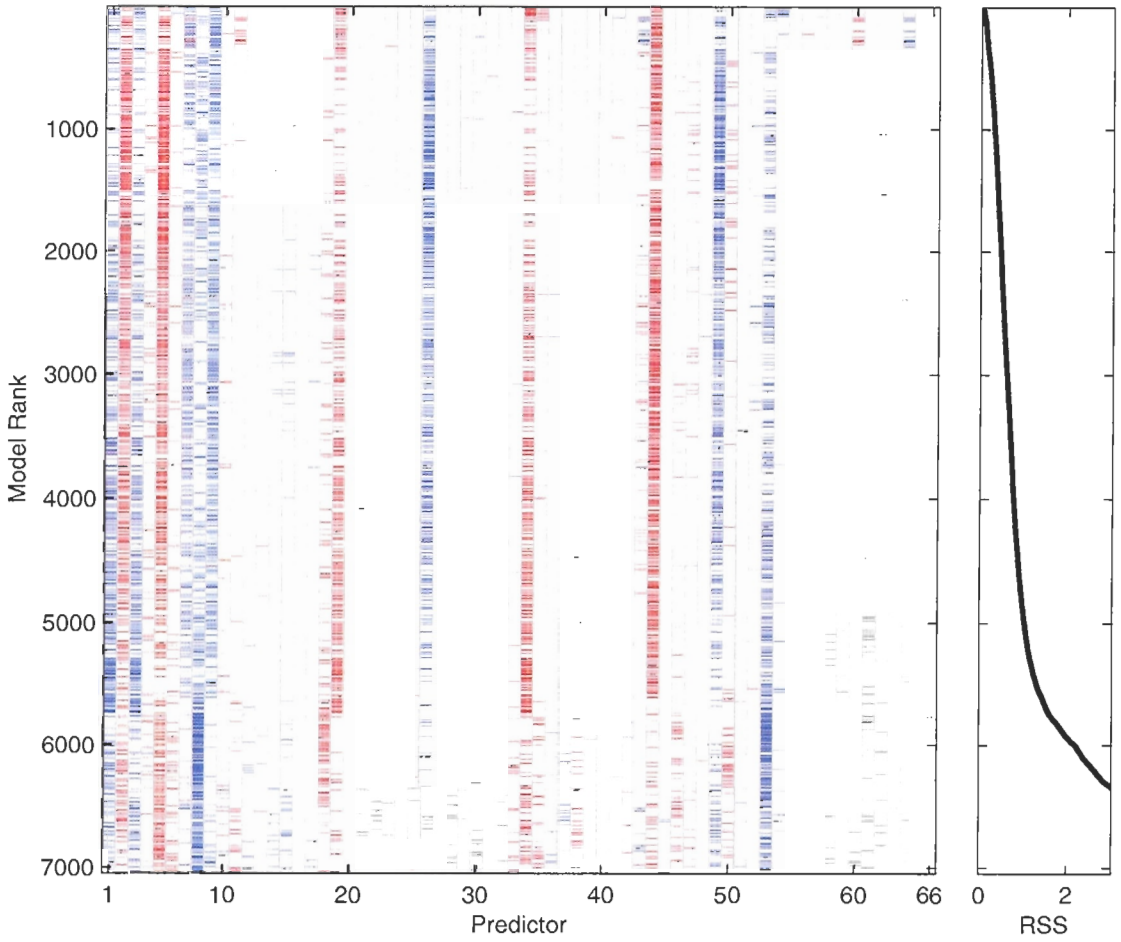

Figure 3.3: Example of a raster plot of the good model set for case 2. This case has much higher model selection uncertainty. There are about a dozen variables that occur frequently in the top models, and it is not clear which variable combinations are most common.

3.3.2 Clustered Raster Plot

The raster plots shown in figures 3.2 and 3.3 show the models sorted in ascending order of RSS. This arrangement helps to distinguish the best-fitting models from the more poorly-fitting models. An alternative to this view is to group similar models together, and then plot the raster diagram in grouped order. Viewing the grouped raster plot can help to make the common submodels in $\mathcal{G}$ more clear.

Grouping similar models can be done using one of many available clustering methods. For present purposes, K-means clustering was chosen. K-means clustering is a method for dividing a set of multivariate observations into K self-similar groups. Group membership is assigned to each observation (using an iterative algorithm) such that the total distance between the observations and their group mean is minimized. See Johnson and Wichern (2002), or Hastie, Tibshirani, and Friedman (2001) for a review of clustering methods, including K-means.

Clustering is done on rows (models) in the matrix representation of the good-model set. Each row is a vector of mostly zeros, with p nonzero elements. To perform clustering, an appropriate distance measure must be defined for comparing two rows. The rows were first converted to indicator vectors—the nonzero components were set to one. Then the Hamming distance (the number of elements that are different) was used to measure distance.

The MATLAB statistics toolbox implementation of K-means clustering was used to cluster the models in $\mathcal{G}$. The best clustering arrangement (minimum total point-to-centroid distance) out of five replications was taken as the final grouping. The resultant raster plots are shown in figures 3.4 and 3.5.

As illustrated in the figures, the clustered raster plot can help to suggest a list of competing models, especially when this is not clear from the original raster plot. The clustered and un-clustered plots together can help to better understand any model aliasing problems that may exist.

3.3.3 Link Plot

The raster plot provides a convenient way to visualize the overall composition of the good-model set. One thing that is not clear from raster plots, however, is the hereditary relationships among variables. Also, the physical layout of the graph prevents the variable names from being clearly identifiable on the plot. The link plot is a second visualization

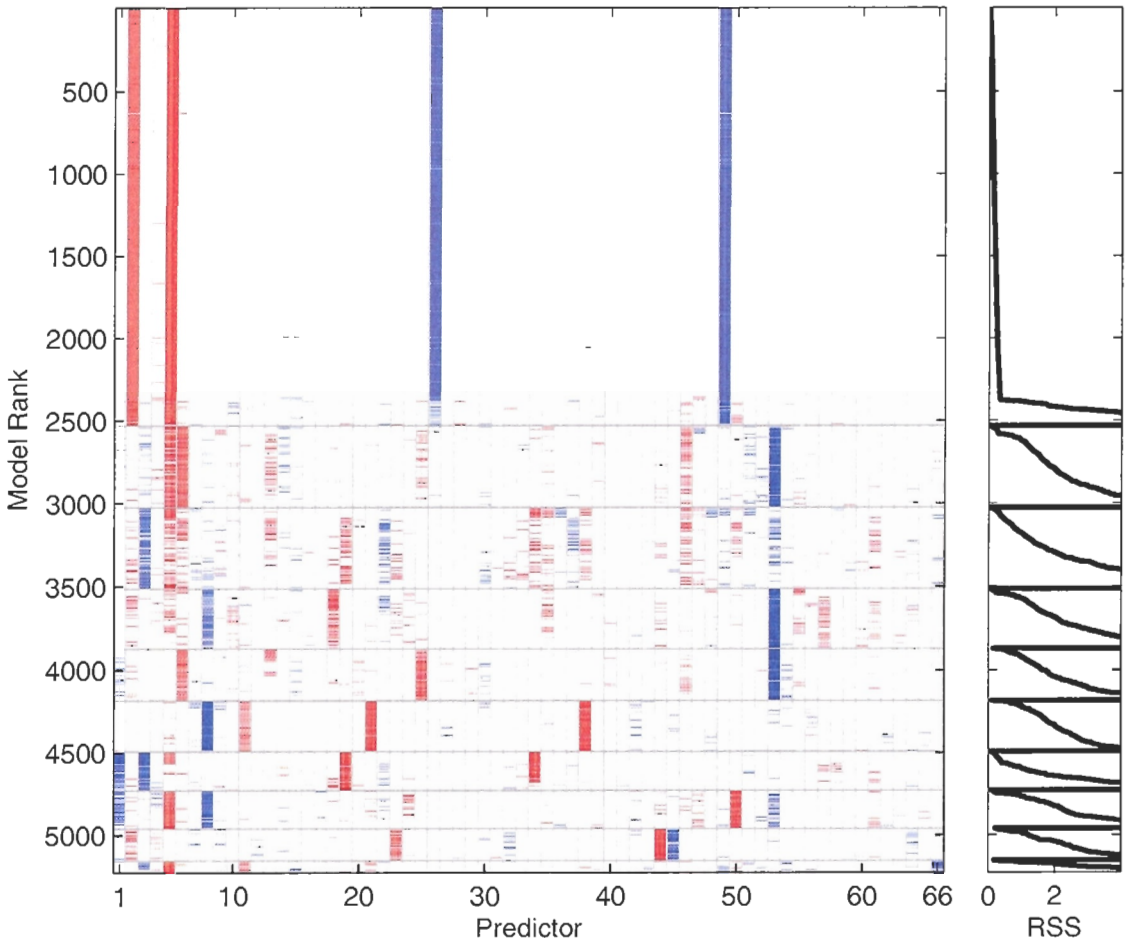

Figure 3.4: Clustered version of the raster plot for case 1. The true model remains clearly visible, but the noisy lower portion of the plot has been reorganized into clearer groupings of models.

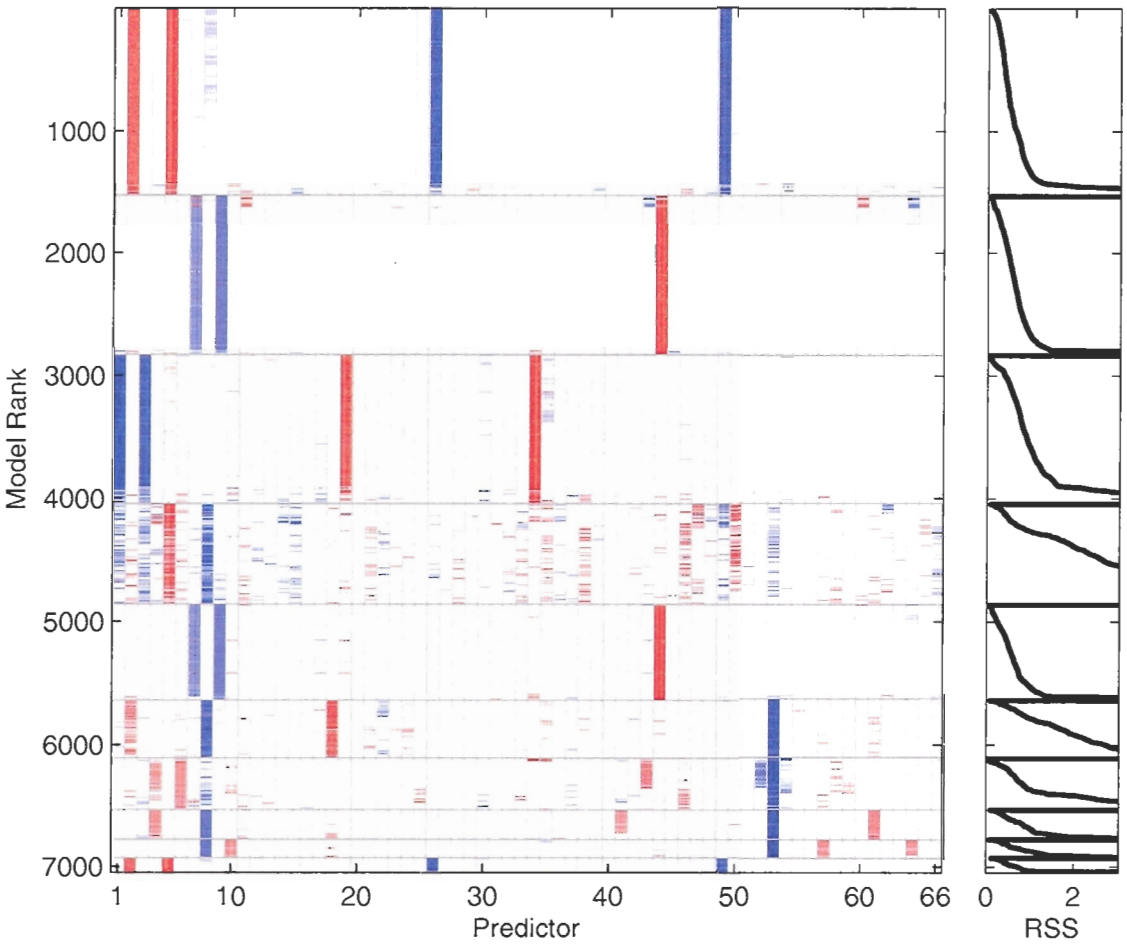

Figure 3.5: Clustered version of the raster plot for case 2. In this case clustering has cleared up the plot considerably, revealing several models that are very common in the good set (compare Figure 3.3). Note that the true model $(2, 5, 26, 49)$ is now visible, but it is only one of several well represented models.

tool proposed to view the information in the set $\mathcal{G}$, with more focus on heredity and model identification.

The motivation for the link diagram comes from the realization that it is not just frequently-occurring variables, but frequently-occurring combinations of variables, that help identify a good candidate model. The starting point for the link plot, then, is to count the frequency of occurrence of all variable *pairs* in $\mathcal{G}$. The occurrence frequencies are then scaled relative to the most frequently-occurring pair, so that every pair of variables is given a link weight between zero and one.

Link plots are shown for the two example cases in figures 3.6 and 3.7. The diagram plots each candidate variable as a point in space; the design variables are displayed on an arc at left, while the interactions are arranged in a vertical line at right. Lines are drawn between pairs of variables, with the width and color of the line reflecting the link weight. Variable pairs with higher occurrence frequencies (link weights) are drawn with thicker, darker blue lines, while pairs with lower weights are drawn with thinner, lighter red lines. The color map used to color the lines is shown in Figure 3.8 (along with the color map used in drawing the raster plot).

The end result of drawing the pairwise frequencies in this way is a web of lines, with thicker and darker lines delineating models that are more strongly supported by the good-model set. Hereditary relationships can be clearly seen on the link plot. Note that links between two interactions are not drawn; they are not necessary, since under the heredity assumption, every interaction must have a link to a main effect.

A model with only two variables will naturally be represented by a line on the link plot. Models of any larger size will appear as a set of connected triangles, formed by the various combinations of variable pairs. In this way, larger models can be nicely represented even though only pairwise frequencies are used to construct the plot. Note also that by using pairwise frequencies, the partial-truth models in $\mathcal{G}$ can still contribute to the emergence of the true model in the figure. If, for example, the true model is (A, B, AC), then any models containing (A, B), (A, AC), or (B, AC) will still help to make the true model more visible.

As with the raster plot, inconsistencies in the model set, due to model aliasing or model selection uncertainty, can also be easily detected. If the darkest lines on the plot are not connected, or do not form triangles, then that means that some links are "missing," and the pattern is actually created by a set of incompatible models. Figure 3.6 shows a typical case where the best model is unambiguous, while Figure 3.7 illustrates the ambiguous case.

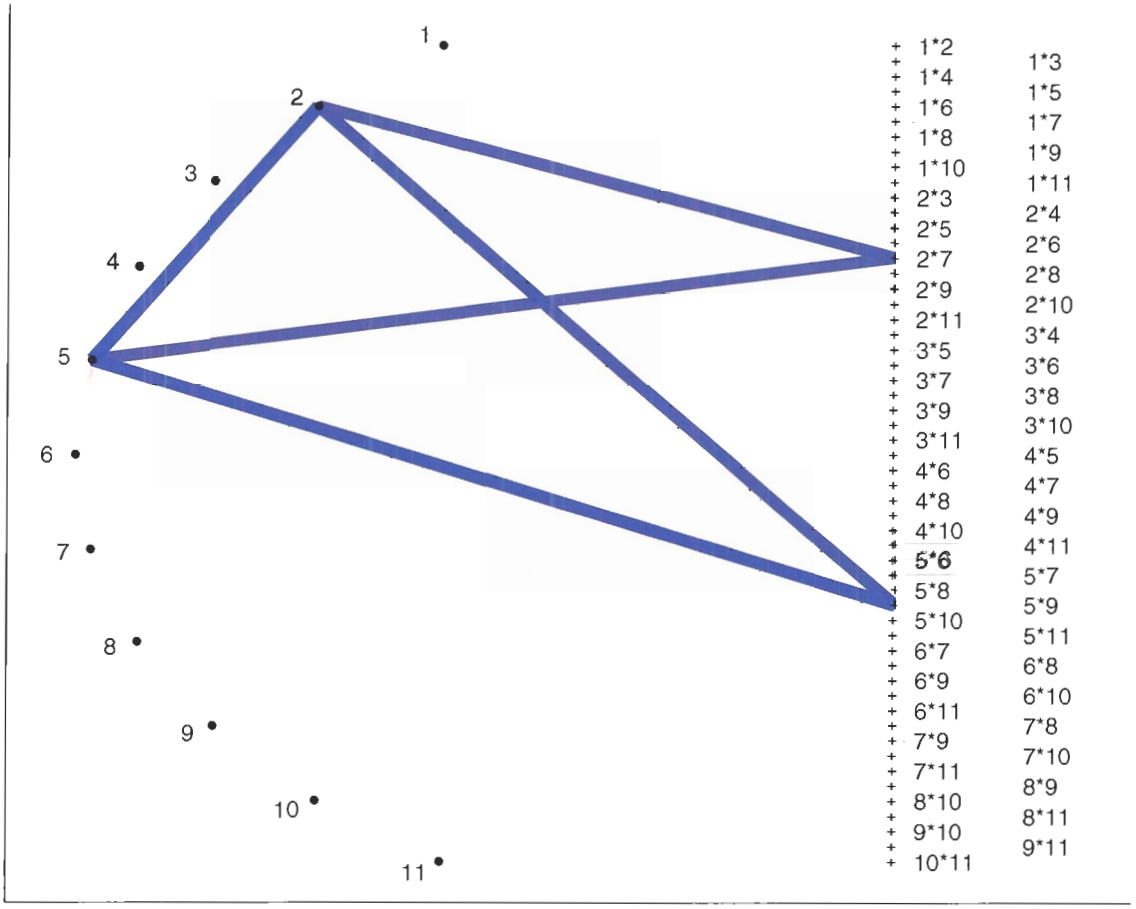

Figure 3.6: Example of a link plot of the good model set for case 1. The frequently-occurring variable pairs, when viewed together, clearly delineate the true model.

The link plots shown here are developed specifically for the case of an experiment considering main effects and two-way interactions. A more flexible version of the link plot could be developed for general regression cases or arbitrarily-defined heredity relationships among variables. The main complication is choosing a sensible spatial layout for the diagram's vertices, so that all important links (and variable names) are easy to see.

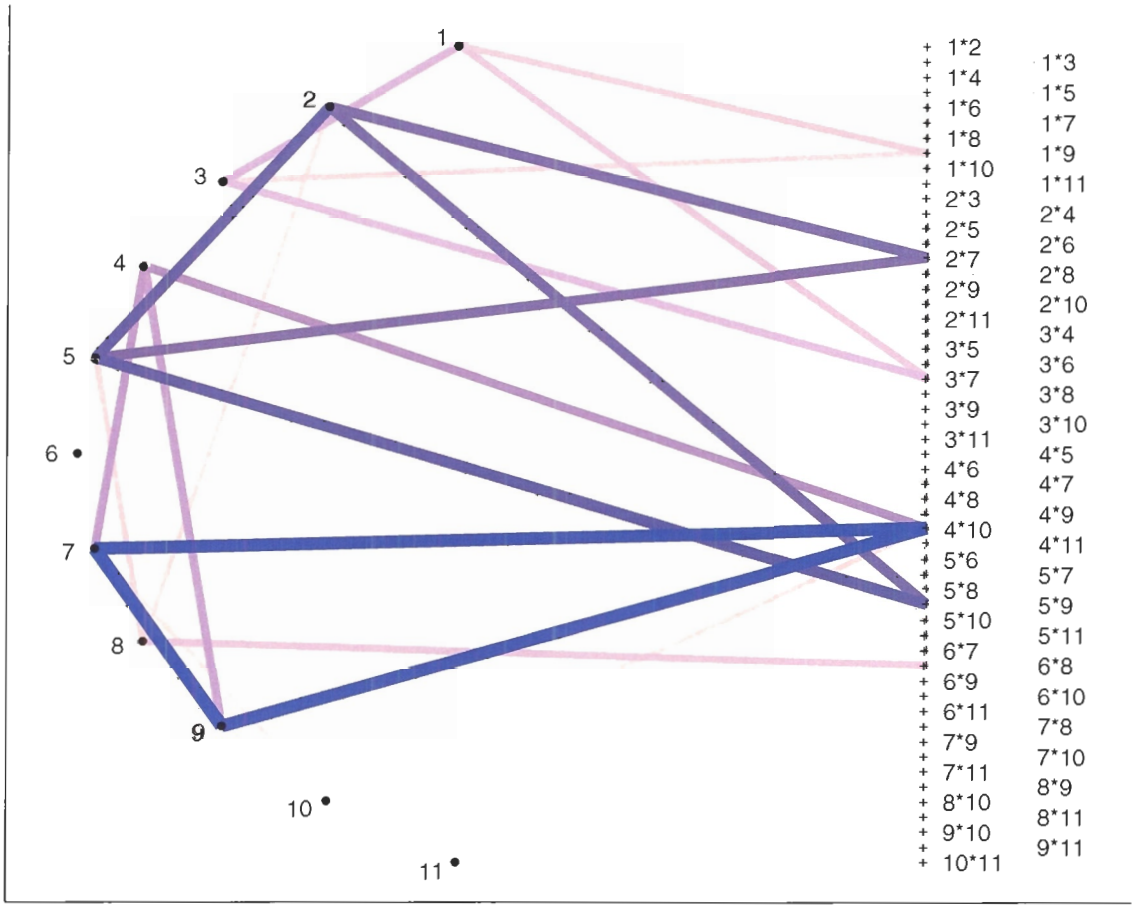

Figure 3.7: Example of a link plot of the good model set for case 2. There are a number of disconnected sets of triangles with significant weight. This is indicative of model aliasing. For example, one could pick out $(1, 3, 1*9, 3*7)$, $(2, 5, 2*7, 5*9)$, and $(4, 7, 9, 4*10)$ as leading candidates.

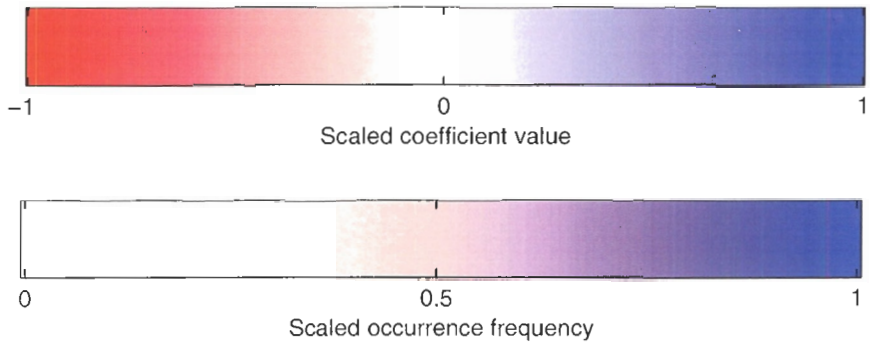

Figure 3.8: Color maps used for the raster diagram (top), and for the link diagram (bottom). In the raster plot, all coefficients are scaled relative to the largest coefficient in the model; in the link plot, link frequencies are scaled relative to the most frequently-occurring variable pair in the model set.

3.4 Automatic Extraction of the Best Model(s)

The visualization approach to interpreting the good model set is simple and intuitive. It allows the investigator to bring in subject-matter knowledge in deciding how many variables, and which ones, are strongly supported by the data. It also has the advantage of clearly showing when the experiment has a high degree of model aliasing or model selection uncertainty. For these reasons it is believed that the graphical procedures of the previous section should be enough on their own to enable good decision making in a real-world application. The procedure from a practitioner's standpoint would be simple:

1. Run the simulated annealing model search to generate the set of well-fitting models.
2. Create the raster plot and link plot. Use them to judge which variables are likely to be active, and what follow-up experiments might be justified.

The above comments notwithstanding, there is a natural motivation to automate the process of extracting a best model (or set of best models) from the good model set $\mathcal{G}$. A key driver for developing such an automatic process is to enable simulation studies such as those performed in Chapter 5. An automatic process would also be useful as a decision support tool to augment the interpretation of the raster plot and link plot.

The guiding principle behind the method of oversized-model sets is that the models in $\mathcal{G}$ contain all of the useful information about the true model. This notion is generally borne out in the raster plot and the link plot, where a human reader's eye can usually see the

important combinations of variables quite easily. The problem of programming a computer to do this process automatically could be approached in many ways. In the present work, the task has been viewed as a feature extraction problem. A solution has been developed, making use of an entropy measure to quantify the degree of support for a particular variable combination.

3.4.1 An Entropy Measure of Support for a Candidate Model

The set of good oversized models, $\mathcal{G}$, contains only models of size p. The assumption throughout has been that the best model is no larger than τ. The feature extraction task at hand, then, is to choose a subset of $q \leq \tau$ variables (a model) that is, in some sense, best supported by $\mathcal{G}$. The chosen variable combination will be proposed as the best approximation to the true model.

Let M be a particular candidate model, consisting of q variables. The models in $\mathcal{G}$ are larger than M; so every member of $\mathcal{G}$ either contains or does not contain M as a submodel. The premise of the method is that most of the models in $\mathcal{G}$ should contain the truth; so the occurrence frequency of M seems like an obvious measure of degree of support for M as the truth.

Unfortunately, it is not possible to simply take the candidate with the highest occurrence frequency as the best model, because for any model, all of its nested models must occur at least as many times. If, for example, model (A, B, AB, BD) occurs 1000 times in the model set, then any chosen submodel—say, (A, B, BD)—will occur at least 1000 times as well. Just looking at model occurrence frequency effectively re-introduces the problem of the appropriate model size.

To get around this problem, a notion of *degree of over-representation* of a candidate in $\mathcal{G}$ is needed. The best model is not just the most frequent one, but the one which is most frequent relative to an expected occurrence frequency. To find the expected occurrence frequency for some candidate M, a reference distribution needs to be defined.

The reference distribution used to define over-representation is the null distribution of model occurrence frequencies assuming *random sampling from the entire population of hereditary models.* In other words, the reference occurrence frequency for a candidate M in $\mathcal{G}$ is the expected number of times M would occur if $\mathcal{G}$ were a simple random sample of m models from $\mathcal{M}$.

The reference frequency can be calculated using the formula for the number of models

that overfit a particular variable combination (equation 2.5). The value N_p is the total number of models in $\mathcal{M}$, and the value $N_p(a,b)$ is the number of overfitted models if M contains a main effects and b interactions. So, under random sampling, the expected proportion of $\mathcal{G}$ containing M is

$$\pi = \frac{N_p(a,b)}{N_p}. \tag{3.1}$$

The value π gives a reference point for how frequently the candidate should be seen in $\mathcal{G}$. Note that this expected frequency depends on the structure of M—specifically, on how many main effects and interactions M contains.

The reference frequency can be used to measure the degree of over-representation of a model in $\mathcal{G}$. Let f be the observed relative frequency of occurrence of candidate model M. One way of measuring the surprise or information content of f relative to the expected frequency π, is to calculate the *relative entropy* of this observed frequency. The relative entropy of M, given $\mathcal{G}$ is defined as

$$H(M) = f \log_2(\frac{f}{\pi}) + (1-f) \log_2(\frac{1-f}{1-\pi}). \tag{3.2}$$

Equation 3.2 is the standard equation for calculating the entropy of a binary random variable with probability f, relative to a reference distribution with probability π (Hamming 1986; Jessop 1994). In this case, each model in $\mathcal{G}$ can either contain or not contain M; so the observed frequency summarizes the result of m binary outcomes.

The entropy measure defined above essentially constitutes a model selection criterion, but one based on the information in $\mathcal{G}$, rather than directly on goodness-of-fit. This criterion indirectly addresses the problem of model size, inasmuch as the model size affects is reference probability of occurrence (π).

The entropy $H(M)$ is intended to measure the degree of over-representation of M in $\mathcal{G}$. It is a larger-the-better criterion; the candidate model with maximum entropy is the one best supported by the data to be the truth. Having defined this criterion, it remains to actually find the M that maximizes $H(M)$.

3.4.2 Finding the Best Candidates

Having defined the entropy measure, the problem of choosing the best variable subset is equivalent to finding the model M, of size $q \leq \tau$, that maximizes $H(M)$. Let this best model be M_B.

Finding M_B is itself a combinatorial optimization problem, much as finding the set of well-fitting models was. Models of size 1 to τ are admissible, so there are $N_{1:\tau}(w)$ candidate models, possibly a large number. Thus it is not practical to find M_B through brute-force evaluation of all candidates.

One alternative is to find the r most frequently-occurring models of size 1 through τ in $\mathcal{G}$, and then to evaluate the entropy of these most common models. This approach works well, because the frequently-occurring subsets can be found quickly using a branch-and-bound algorithm.

The branch-and-bound method is described in more detail in Appendix B. The key idea of the algorithm is that if a particular subset of variables occurs f times in the good set, then no larger model containing that subset can occur more than f times. So identifying smaller models with low occurrence frequencies allows large branches of the search to be eliminated without explicitly visiting all the models in the branch. Given good initial guesses and an appropriate algorithm for moving through the tree of possible models, the r most frequent models of each size can be found quite quickly.

The entropy $H(M)$ depends not only on the the model's occurrence frequency in $\mathcal{G}$, but also on the its structure and size. So the j^{th} most frequent model of size q will not necessarily be the j^{th} highest entropy model. The occurrence frequency dominates the entropy calculation, however; and usually there will only be a small number of frequently-occurring models of each size. So the set of the 5 most frequent models of each size is almost certain to contain the maximum entropy model.

Summarizing these ideas, the following steps are recommended for finding the models best-represented in $\mathcal{G}$:

1. Perform branch-and-bound search as in Appendix B, to find the 5 most frequent models of each size from 1 to τ. Pool these 5τ models into a single list.

2. Calculate the entropy of each of these models as in equation 3.2. Sort the list in descending order of entropy.

3. (Optional) Delete from the list any models that are nested within another model of higher entropy. This step may help clean up the list for presentation purposes.

The end result of these steps is to produce a list of several models that are well represented in the good model set, along with their entropy measures. The entropy measures

Table 3.2: Highest-entropy models, for case one and case two of the example.

Case 1		
M	$H(M)$	f
$(2, 5, 2{*}7, 5{*}9)$	3.539	2291
$(2, 5, 6, 2{*}7, 5{*}9)$	0.941	563
$(2, 5, 2{*}7, 5{*}6, 5{*}9)$	0.502	309
$(2, 5, 2{*}7, 2{*}11, 5{*}9)$	0.462	288
$(2, 5, 8, 2{*}7, 5{*}9)$	0.433	292
$(2, 5, 2{*}7, 2{*}8, 5{*}9)$	0.430	271
$(6, 6{*}8)$	0.287	1075
(9)	0.045	1019
(7)	0.045	1021

Case 2		
M	$H(M)$	f
$(2, 5, 2{*}7, 5 * 9)$	1.175	1338
$(7, 9, 10, 4{*}10)$	0.878	1177
$(1, 3, 1{*}9, 3{*}7)$	0.703	887
$(4, 7, 9, 4{*}10)$	0.692	978
$(2, 5, 8, 2{*}7, 5{*}9)$	0.415	402
$(2, 5, 2{*}7, 2{*}11, 5{*}9)$	0.256	250
$(1, 3, 1{*}9, 3{*}7, 3{*}8)$	0.217	217
$(8, 6{*}8)$	0.215	1327
$(7, 9, 10, 4{*}10, 9{*}10)$	0.202	221
$(7, 9, 10, 4{*}9, 4{*}10)$	0.188	208

may be used to interpret the relative amount of support for each model. If it is necessary to report a single model as the best guess at the truth, the top model in the list can be chosen.

Steps 1–3 above were applied to the example of this section, for both case one and case two. Recall that the true model was $(2, 5, 2{*}7, 5{*}9)$. The results are listed in Table 3.2.

For case one, where there is little model selection uncertainty, the top model was indeed the true model, and its entropy was far higher than the rest of the candidates. Note also that the next five highest-entropy models are all made up of the true model plus one additional variable.

For case two, where model selection uncertainty is higher, the results are not as clear. The true model still gets flagged as the maximum-entropy model, but there are a few other models, distinct from the best one, that have reasonably high entropy values.

The entropy method presented here provides a useful way of automating the extraction of submodels from the good-model set. Experience has shown that it typically does well in making selections that agree with the visual appearance of the raster and link plots. Comparison of the plots in figures 3.2 through 3.7 with Table 3.2 will illustrate this point.

The entropy criterion will be used to choose a best model in the simulation studies to follow. In a real data analysis, however, it is not recommended to rely solely on the entropy method. The results of the model extraction should be used as a decision support tool, to help ensure that the plots are interpreted correctly, without overlooking some evidence.

Chapter 4

Performance on Literature Examples

Published model selection examples provide an opportunity to illustrate the characteristics of the new method. Four such examples are presented in this chapter. The examples illustrate different characteristics of the new simulated annealing method, and show how it can be applied to different types of experimental designs. Two of the examples also provide an opportunity to compare results with Bayesian variable selection, for which it is difficult to formulate more structured, simulation-based comparisons.

4.1 Two PB_{12} Cases

The Bayesian variable selection approach has been used to study the 12-run Plackett-Burman design. The design matrix and responses for two examples are given in Table 4.1. In both examples, the responses were constructed from known true models.

Response one

The first response column, labelled Y_1 in the figure, is taken from Chipman, Hamada, and Wu (1997). The true data-generating process in this case is $Y_1 = A+2AB+2AC+\epsilon$, where ϵ has standard deviation $\sigma = 0.25$. This example is relatively easy, because the error variance is significantly smaller than the magnitude of the true coefficients. Any model containing the truth can be expected to fit the data better than other models.

Table 4.1: Design matrix and responses for the two examples of Section 4.1.

Run	A	B	C	D	E	F	G	H	I	J	K	Y_1	Y_2
1	1	1	-1	1	1	1	-1	-1	-1	1	-1	1.058	-1.358
2	1	-1	1	1	1	-1	-1	-1	1	-1	1	1.004	1.228
3	-1	1	1	1	-1	-1	-1	1	-1	1	1	-5.200	2.291
4	1	1	1	-1	-1	-1	1	-1	1	1	-1	5.320	9.432
5	1	1	-1	-1	-1	1	-1	1	1	-1	1	1.022	-5.719
6	1	-1	-1	-1	1	-1	1	1	-1	1	1	-2.471	-2.417
7	-1	-1	-1	1	-1	1	1	-1	1	1	1	2.809	-2.494
8	-1	-1	1	-1	1	1	-1	1	1	1	-1	-1.272	2.674
9	-1	1	-1	1	1	-1	1	1	1	-1	-1	-0.955	-5.943
10	1	-1	1	1	-1	1	1	1	-1	-1	-1	0.644	1.596
11	-1	1	1	-1	1	1	1	-1	-1	-1	1	-5.025	6.682
12	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	3.060	-5.973

Results for the Bayesian variable selection (BVS) and the simulated annealing model search (SAMS) are listed in Table 4.2. Only the top three models are shown, because the true model is identified by both methods as a clear winner. The second- and third-place models in each case are overfitted models, though the two methods do not select the same overfitted models.

Response two

The second response column in Table 4.1, labelled Y_2, is taken from Chipman (1998). The data come from the process $Y_2 = 2A + 4C + 2BC + 2CD + \epsilon$, with $\sigma = 0.5$. This example is called "a more difficult problem" because there are more active variables, the noise variance is larger, and the large effect of C tends to dominate the Bayesian model search.

The Bayesian search results are more difficult to report in this case. Using the weak heredity prior, Bayesian variable selection found that the single-factor model (C) had by far the highest posterior probability, 0.221. The true model (A, C, BC, CD) did have the highest posterior probability among models of size four, but this probability was only 0.02, which would lend little support to this model if the truth were unknown. The variables with highest marginal probability were C, CD, H, J, BH, and BC. So the marginal probabilities also do not help to clarify the true model in this case. Results were similarly ambiguous for other choices of the prior.

Table 4.2: Comparison of Bayesian variable selection results to the new method for the first PB_{12} example. True model is (A, AB, AC), response is Y_1.

BVS		SAMS	
Model	**Probability**	**Model**	**Entropy**
(A, AB, AC)	0.325	(A, AB, AC)	4.7
(A, C, AB, AC)	0.039	(A, AB, AC, AJ)	2.2
(A, B, AB, AC)	0.022	(A, F, AB, AC)	2.1

The results of the simulated annealing method are compared to the BVS results in Table 4.3. For the Bayesian method, the top models of different sizes, under the weak heredity prior, are reported. The new oversized-models method clearly outperformed the BVS method in this case. The true model is not only ranked first, but its entropy measure is clearly much higher than the next-highest models. The raster plot and link plot (not shown) also clearly indicate that (A, C, BC, CD) is most supported by the good-model set.

4.2 A Folded-Over PB_{12} Design

An example from Miller and Sitter (2001) can be used to illustrate the performance of the new model selection method on a somewhat larger experiment. The design considered is a folded-over 12-run Plackett-Burman design, denoted PB_{12+12}. This design has 24 runs and allows up to 12 main effects to be considered. The 66 two-way interaction columns can be added, yielding a full matrix that has 78 columns and exhibits complex aliasing. Miller and Sitter acknowledge the difficulty in searching the model space, and propose to use a two-stage search procedure like the one outlined in Section 2.2.2.

The design matrix (not reproduced here) is given with two response vectors: "reactor data" and "contaminant data." The oversized-model sets procedure was conducted on both of the responses. The model size, p, was set to 10 in this case, because the larger number of runs makes it possible to entertain larger models. The maximum plausible truth size, τ, was set to eight. The simulated annealing parameters were set to typical values $n_{gen} = 10000$, $\rho = 0.95$, $\kappa = 4$, and $P_{min} = 0.001$.

Table 4.3: Comparison of Bayesian variable selection results to the new method for the second PB_{12} example. True model is (A, C, BC, CD), response is Y_2.

BVS		SAMS	
Model	**Probability**	**Model**	**Entropy**
(C)	0.221	(A, C, BC, CD)	3.2
(C, CD)	0.065	(A, C, AC, BC, CD)	0.9
(C, J)	0.04	(A, C, I, BC, CD)	0.7
(C, BC)	0.031	(A, B, C, BC, CD)	0.6
(C, H, BH)	0.028	(A, C, H, AD, BH)	0.5
(C, E, EI)	0.021	(A, C, D, BC, CD)	0.5
(C, I, EI)	0.021		
(A, C, BC, CD)	0.02		
(C, D, CD, DG)	0.012		
(B, C, BC, BH)	0.01		
(A, C, D, BC, CD)	0.002		
(C, G, H, BH, GH)	0.002		
(B, C, BC, BH, CF)	0.002		

Reactor data

The reactor data is a reduced form of a data set previously used in Box, Hunter, and Hunter (1978). The model (B, D, E, BD, DE) is given as the set of factors identified as active in the original study.

The analysis based on oversized models agrees with the results of Miller and Sitter. The five largest-entropy models are listed in the left half of Table 4.4. The previously-suggested best model (B, D, E, BD, DE) has much higher entropy than any other model found in the good-model set.

Contaminant data

The contaminant data comes from an industrial experiment, and the two-stage analysis chooses the (A, B, AB) as the best model. In the output from the new algorithm (right half of Table 4.4), the suggested best model (A, B, AB) does have the highest entropy, but two overfitted models have almost equal entropy values.

The link plot for the contaminant data, shown in Figure 4.1, also seems to cast doubt on whether there should be three, four, or five variables in the chosen model. The raster

Table 4.4: Results of the new method applied to the PB_{12+12} example.

Reactor Data		Contaminant Data	
Model	**Entropy**	**Model**	**Entropy**
(B, D, E, BD, DE)	10.5	(A, B, AB)	5.95
(B, D, E, J, BD, DE)	4.6	(A, B, J, AB)	5.90
(B, D, E, BD, DE, EL)	3.5	(A, B, J, AB, AE)	5.68
(B, D, E, BD, BH, DE)	2.7	(A, B, J, AB, AE, AG)	4.29
(B, D, E, H, BD, DE)	2.5	(A, B, J, AB, DJ)	3.85

plot (Figure 4.2), however, suggests that variables J and AE are probably not necessary. These two variables correspond to columns 10 and 16 in the raster plot, and clearly their coefficients are very small relative to the coefficients of A, B, and AB.

This example illustrates the importance of using all of the information in the raster plot, link plot, and entropy calculation together to make decisions about the model. Although the three-variable model only slightly beats out the larger models on the entropy scale, it is still significant to note that a method that searches through models of size 10 can successfully suggest a good model with only three variables.

4.3 A Mixed-Level Experiment

Wu and Hamada (2000) analyzed an experiment studying the effects of one two-level factor (A) and seven three-level factors (B–H) on a blood glucose reading. The design matrix is given in Table 4.5. The linear-quadratic system has been used to decompose the three-level factors into linear and quadratic terms, so that there are effectively 15 design variables involved in the analysis: the two-level factor A, and the linear and quadratic components of B–H, denoted $Bl, Bq, Cl, Cq, \ldots, Hl, Hq$.

All valid two-way interactions were added to the design matrix to form the full matrix. These include the 14 interactions between A and the linear-quadratic columns, as well as the $\binom{14}{2} - 7 = 84$ linear-by-linear, linear-by-quadratic, and quadratic-by-quadratic interactions involving different three-level factors. The full matrix thus includes 113 predictors in 18 runs, and exhibits complex aliasing, making this a challenging model selection problem.

Bayesian variable selection was applied by Wu and Hamada to perform model selection. The priors were defined so as to encourage hereditary models. Quadratic terms were given

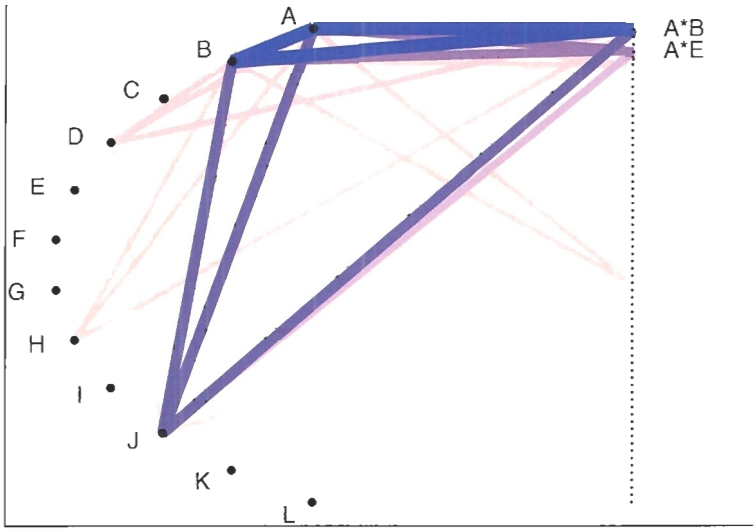

Figure 4.1: Link plot for the contaminant data example.

Table 4.5: Design matrix and responses for the blood glucose experiment. The seven three-level factors B–H have been separated into linear and quadratic components.

Run	A	Bl	Bq	Cl	Cq	Dl	Dq	El	Eq	Fl	Fq	Gl	Gq	Hl	Hq	Y
1	0	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	97.94
2	0	0	-2	0	-2	0	-2	0	-2	0	-2	-1	1	0	-2	83.40
3	0	1	1	1	1	1	1	1	1	1	1	-1	1	1	1	95.88
4	0	-1	1	-1	1	0	-2	0	-2	1	1	0	-2	1	1	88.86
5	0	0	-2	0	-2	1	1	1	1	-1	1	0	-2	-1	1	106.58
6	0	1	1	1	1	-1	1	-1	1	0	-2	0	-2	0	-2	89.57
7	0	-1	1	0	-2	-1	1	1	1	0	-2	1	1	1	1	91.98
8	0	0	-2	1	1	0	-2	-1	1	1	1	1	1	-1	1	98.41
9	0	1	1	-1	1	1	1	0	-2	-1	1	1	1	0	-2	87.56
10	1	-1	1	1	1	1	1	0	-2	0	-2	-1	1	-1	1	88.11
11	1	0	-2	-1	1	-1	1	1	1	1	1	-1	1	0	-2	83.81
12	1	1	1	0	-2	0	-2	-1	1	-1	1	-1	1	1	1	98.27
13	1	-1	1	0	-2	1	1	-1	1	1	1	0	-2	0	-2	115.52
14	1	0	-2	1	1	-1	1	0	-2	-1	1	0	-2	1	1	94.89
15	1	1	1	-1	1	0	-2	1	1	0	-2	0	-2	-1	1	94.70
16	1	-1	1	1	1	0	-2	1	1	-1	1	1	1	0	-2	121.62
17	1	0	-2	-1	1	1	1	-1	1	0	-2	1	1	1	1	93.86
18	1	1	1	0	-2	-1	1	0	-2	1	1	1	1	-1	1	96.10

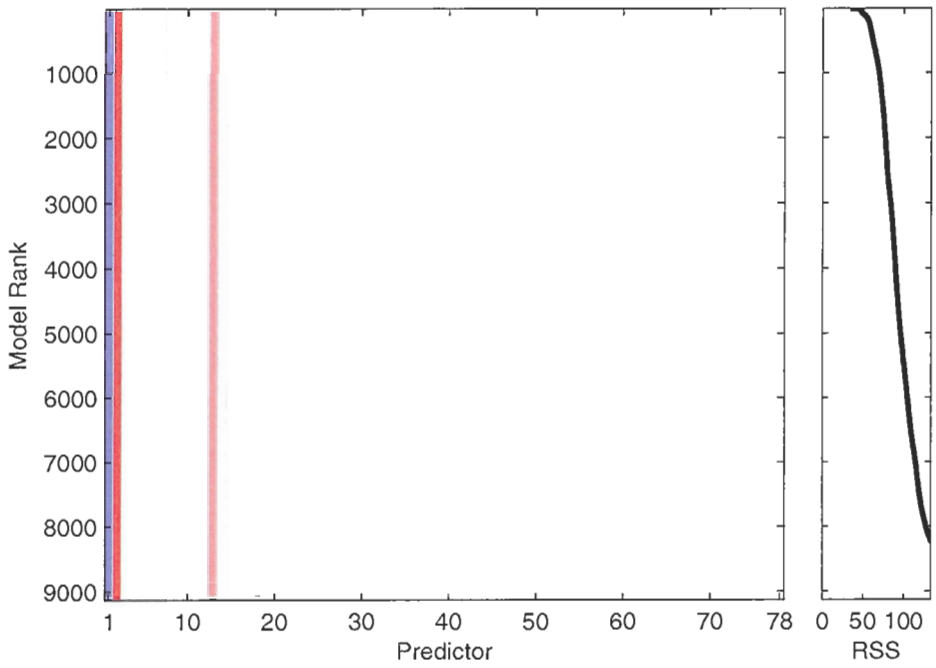

Figure 4.2: Raster plot for the contaminant data example. Variables J and AE (columns 10 and 16 in the plot) appear important on the link plot, but do not show up here because their coefficients are too small.

more prior weight if the corresponding linear term was included also. Interactions were given more weight if one parent effect was included in the model, and yet more weight if both parents were included.

The models found to have highest posterior probability are shown in Table 4.6. Results are shown for two cases, corresponding to different specifications of the priors. The relaxed weak heredity prior permits violations of heredity, but puts low prior probability on such models. The strict weak heredity prior puts probability of zero on non-hereditary relationships, so that only hereditary models will have nonzero probability in the posterior. The table indicates that model $(BlHq, BqHq)$ is most strongly supported using the relaxed prior, while $(Bl, BlHl, BlHq, BqHq)$ is favoured using the strict prior. Considering the top models as a whole, the list is dominated by the linear and quadratic components of B and H, as well as their various interactions.

The presence of quadratic terms in this example complicates the application of the simulated annealing search method. For simplicity, the 15 columns shown in Table 4.5 were treated as design variables, and the analysis was run in the usual fashion, considering all of the appropriate interactions. This means that the quadratic terms (such as Bq) are given equal footing with the linear terms, and may appear in any model without heredity restrictions. Only the interaction terms are required to satisfy heredity.

The other problem in applying the new method to this case is in the automated model-extraction procedure. The entropy calculation as given in Section 3 is valid only for the standard case where w main effects and all $\binom{w}{2}$ interactions are entertained. This is not the case here, since linear-quadratic interactions involving the same factor (e.g. $BlBq$) are not included. Mindful of this limitation, the analysis will be done solely through interpretation of the link plot.

Simulated annealing model search was performed with the model size p set to eight. The link plot, shown in Figure 4.3, clearly supports the model $(Bl, Bq, BlHq, BqHq)$ as the best model. This particular four-variable combination occurred in approximately one third of the more than nine thousand unique models generated by the search. The R^2 value for this model is 0.89; the highest-posterior-density models reported by Wu and Hamada all had R^2 values in the range 0.79–0.89.

The model $(Bl, Bq, BlHq, BqHq)$ agrees with the Bayesian results in the sense that it involves the same two factors B and H, but this particular model has relatively low posterior density: 0.008 for the relaxed prior, and 0.013 (for a model with one extra factor) in the strict

Table 4.6: Top posterior model probabilities from Bayesian variable selection, blood glucose experiment.

Relaxed Weak Heredity		Strict Weak Heredity	
Model	**Prob.**	**Model**	**Prob.**
$(BlHq, BqHq)$	0.183	$(Bl, BlHl, BlHq, BqHq)$	0.146
$(Bl, BlHq, BqHq)$	0.080	$(Bl, BlHl, BqHl, BlHq, BqHq)$	0.034
$(Bl, BlHl, BlHq, BqHq)$	0.015	$(Hl, Hq, BlHq, BqHq)$	0.033
$(Fl, BlHq, BqHq)$	0.014	$(Hl, BlHl, BlHq, BqHq)$	0.031
$(GlEl, BlHq, BqHq)$	0.013	$(Fl, Fq, DlFl, DqFl, ElFl)$	0.024
—3 others—	—	$(Hl, Hq, AlHq, BlHq, BqHq)$	0.017
$(Bl, Bq, BlHq, BqHq)$	0.008	$(Bl, Bq, BlHl, BlHq, BqHq)$	0.013

prior. One possible reason for this is that the specification for the prior puts more weight on interactions that have *both* parents in the model, so that good models with "single-parent" interactions (like $BlHq$ and $BqHq$) appear further down the list than they otherwise might. This underscores the importance of appropriately specifying the acceptable relationships among variables, regardless of the method being used.

4.4 A Regression Example

Designed experiments with complex aliasing have much in common with multiple regression problems in observational studies, in that the predictor variables are not orthogonal and cannot be considered independently. This suggests that any method suitable for analysis of nonregular factorial designs might also be useful for model selection in the general multiple regression setting.

This idea is tested by applying the oversized-models approach to the ozone data set studied in, among others, Breiman (1995) and Miller (2002). The data consist of 330 cases, with eight design variables and one response. The design variables are eight meteorological variables, and the response is the measured ozone concentration. Variable names and descriptions are listed in Table 4.7. The Miller and Breiman studies both included the 28 two-way interactions among the main effects, as well as the eight quadratic terms, giving a total of 44 predictor variables in the full matrix. The design variables were first centered to have mean zero, then the interactions and squared columns were formed, and then all variables were standardized.

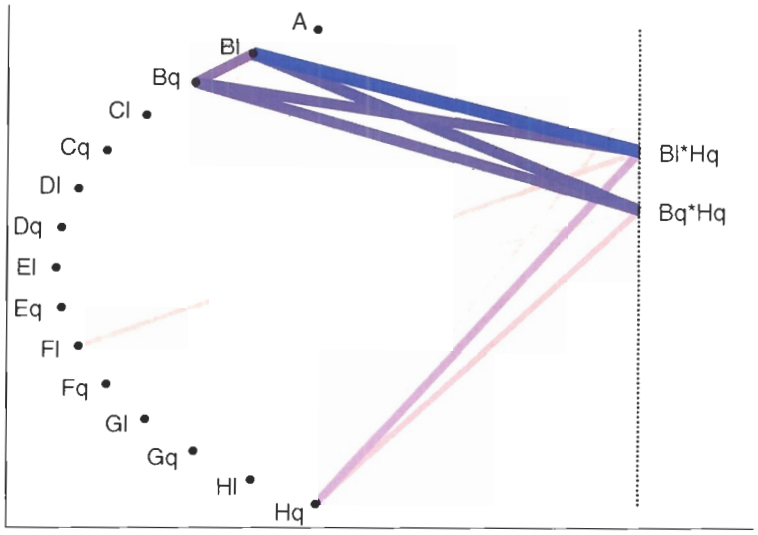

Figure 4.3: Link plot for the blood glucose example.

The ozone data provides an interesting test case for the new model selection method. Although the number of predictors (44) is relatively small compared to the previous examples, the large n in this problem means that larger models can be considered, so that the model space is still very large. Also, since $n > k$ in this case, any model up to and including the full model can be estimated. The assumption of effect sparsity may also be questionable in this analysis, since the system being studied (the weather) is complex and interactive.

As in the blood glucose example, the presence of quadratic terms makes it inappropriate to use the entropy measure for automatically selecting best models. The analysis will instead be done by visual inspection of the raster and link plots. The quadratic terms were not treated as design variables for this example. Effect heredity requires that any quadratic term must also have its corresponding main effect in the model.

The simulated annealing search was performed for two model sizes, $p = 9$ and $p = 13$. The searches were run for 10000 accepted moves, with other parameters left at their default values. The results are shown graphically in Figure 4.4.

The figure shows that two different models are suggested at the two different values of p. For $p = 9$, the four-variable model (RH, T, iT, RH*iT) appears to dominate. When $p = 13$, variable RH*iT is no longer frequently chosen, and five other variables come in, yielding

the eight-variable model (RH, T, iHt, iT, RH*T, T*P, P*iT, RH^2). This model appears to be strongly supported, as the majority of the top 13-variable models found contain this submodel.

Table 4.8 compares three sets of models: the full model (Full), the best-fitting subsets of sizes one through eight given by Miller (2002) (M1–M8), and the two models suggested by the link plots (L4 and L8). The table lists the RSS and R^2 values for each model, and indicates which models respect heredity. The column PE in the table gives the average squared prediction error associated with each model, measured by 10-fold cross-validation as in Breiman (1995). Breiman's models and results are not given here because they could not be reproduced from the information in the paper.

The full model has an RSS of 4392, R^2 of 0.79, and prediction error of 17.4. Models M1 through M8 show that although a number of variables appear repeatedly in most of the models, the best-fitting models are not nested within one another. This is a common problem when considering models purely based on goodness-of-fit. Variable P, for example, occurs in the best-fitting models of sizes 4–7, but not in the best-fitting models of sizes 3 or 8. These low-RSS models do fit the data well, however, and have good PE values. Models M4–M8 all have R^2 values close to the full model and prediction errors as good or better than the full model.

The models taken from the link plots (L4 and L8) can be proposed as competitors to the best-fitting models of the same size (M4 and M8). L4 and M4 have two predictors in common, and both respect heredity. The eight-variable models have four common predictors, and only L8 respects heredity. The RSS, R^2, and PE values for the L4 and L8 are good, but all slightly worse than those for M4 and M8.

It is clear from Figure 4.4 and Table 4.8 that there is model selection uncertainty in this case. If one wanted to choose a parsimonious model of, say, four or eight variables, there is little evidence in the data to suggest which element of {L4, M4} or {L8, M8} would be best to choose. An investigator might break such near-ties by choosing a model that is hereditary over one that is not, or choosing a model with simpler physical interpretations over a model that is more difficult to understand. Comparing models L4 and M4, for example, L4 satisfies strong heredity and has only one interaction; while M4 only satisfies weak heredity and has two interactions. Similarly, L8 satisfies weak heredity and involves only physically reasonable temperature-humidity and temperature-pressure interactions; model M8 doesn't satisfy heredity, and has (presumably) more difficult-to-explain predictors like P^2 and RH*P.

Table 4.7: Response variable and design variables for the ozone data.

Variable	Quantity	Label
Y	ozone concentration (ppm)	–
X_1	500 millibar height (m)	Ht
X_2	wind speed (mph)	Wd
X_3	relative humidity (%)	RH
X_4	surface temperature (°F)	T
X_5	inversion height (ft)	iHt
X_6	pressure gradient (mmHg)	P
X_7	inversion temperature (°F)	iT
X_8	visibility (mi)	V

This regression example illustrates the potential usefulness of simulated annealing model search and the associated plots to aid decision making in model building for multiple regression. It could be particularly useful in cases that match the description of screening experiments: large number of candidate models, hereditary models desired, and the assumption that the best model is relatively small.

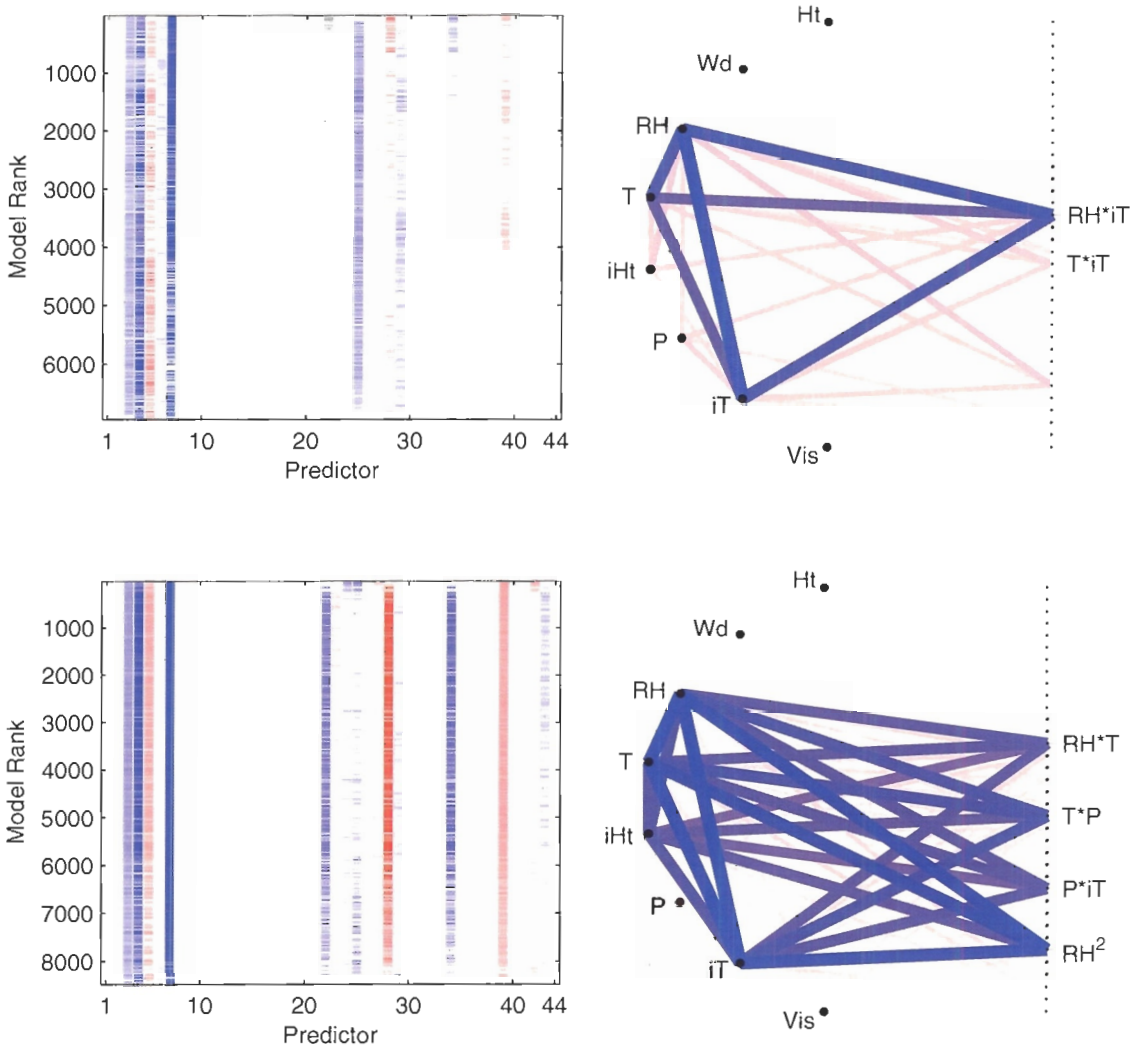

Figure 4.4: Raster plots and link plots for the ozone data, for model search done with $p = 9$ (top) and $p = 13$ (bottom).

Table 4.8: Comparison of proposed models for the ozone data. M1–M8 are the best subsets from Miller (2002); L4 and L8 are the models suggested by the link plots. PE is the cross-validation prediction error.

Model	**Variables**	**Hered?**	**RSS**	$\mathbf{R^2}$	**PE**
Full	(All 44 variables)	Y	4392	0.79	17.4
M1	(T)	Y	8246	0.61	25.3
M2	(T, Ht*RH)	N	7165	0.66	22.1
M3	(RH, iT, RH*iT)	Y	6140	0.71	19.1
M4	(P, iT, P^2, RH*iT)	Y	5565	0.74	17.4
M5	(P, iT, T*iHt, P^2, RH*iT)	N	5186	0.75	16.3
M6	(RH, P, iT, T*iHt, P^2, RH*iT)	N	5039	0.76	15.9
M7	(RH, P, iT, RH^2, T*iHt, P^2, RH*iT)	N	4984	0.76	15.9
M8	(RH, T, iT, RH^2, RH*P, P^2, RH*iT, T*iT)	N	4883	0.77	15.6
L4	(RH, T, iT, RH*iT)	Y	5756	0.73	18.0
L8	(RH, T, iHt, iT, RH*T, T*P, P*iT, RH^2)	Y	5042	0.76	16.1

Chapter 5

Simulation Studies

The oversized-models concept has exhibited good performance on a variety of test cases. Simulation studies were conducted to substantiate the claim of good performance in a more systematic way. The first study looks at the sensitivity of the method to its adjustable parameters. The second study assesses the performance of the new method relative to alternative methods. Both parts rely on a model-generating process that generates true models for the simulations. The true models have randomly-chosen active variables and randomly-assigned coefficients.

5.1 A Model-Generating Process

In the present simulations, emphasis is placed on assessing the average performance of model selection methods *across a wide range of plausible truths.* Mindful of this goal, the simulations are set up in a manner somewhat different from typical published studies. Many simulation studies start with a fixed "target" model, and perform model selection repeatedly on different $\boldsymbol{y}$-vectors realized from the model. The target model may be changed incrementally—for example by scaling the true coefficients, or by adding or removing certain variables—but usually only a few different scenarios are considered.

Rather than choosing only one or a handful of true model configurations, the simulations performed here sample from a *distribution of true models.* Each model selection iteration involves a true model with randomly-selected active variables and randomly-assigned coefficients. The mechanism of choosing predictors and their coefficients is designed so that the resulting models have characteristics that might be expected in real-world screening

experiments. Results of simulations performed in this way should be less dependent on the particular choice of true model, yielding more meaningful performance measures.

The simulations will focus on Plackett-Burman designs, with all main effects and two-factor interactions considered. The PB_{12} design (66 variables) and the PB_{20} design (190 variables) will be used as smaller- and larger-scale representative designs.

Setting up a true model for a simulation involves a sequence of decisions: first, what will the size of the model be; next, which particular combination of variables will be active; and finally, what will their coefficient values be. A model-generating process automates these decisions and effectively defines a distribution of possible truths. Simulations can then be based on true models sampled from this distribution.

The primary motivation for using a model-generating process is to better capture the real-world performance of the method, rather than its relative performance on specific cases. To specify what is meant by "real-world," one must address the question of what types of true models would seem plausible in the context of a screening experiment. Alternatively, one could ask what type of true model would be assumed to exist in order to justify doing a screening experiment in the first place. The following premises are proposed as an answer to these questions:

1. *The true model should satisfy effect heredity.*

2. *The true model should have between 1 and τ active predictors.* The maximum truth size, τ, is a user input, guided by the idea of effect sparsity. Coefficients for all other predictors should be zero.

3. *The chosen coefficients should not be restricted to have equal magnitudes or signs.* This is to avoid artificially influencing the amount and type of model aliasing that can occur.

4. *Variables to be considered active should have coefficients above some lower bound.* It is unreasonable to deem a variable "active," and then assign it a coefficient that is undetectable using the given design. A lower bound on coefficient magnitude can be based on the power to reject $H_o : \beta = 0$ in a standard t-test.

5. *The magnitude of active coefficients should have an upper bound as well.* Real-world experiments are not usually expected to have R^2 extremely close to one, even if the

true model is known. The true model, therefore, should not produce data with an unrealistically large R^2 value. A model having $E[R^2] > 0.95$, for example, might be considered unrealistic. To achieve this limitation on R^2, coefficients must be bounded above.

6. *The upper bound on coefficient magnitude should decrease with model size.* Larger models must have smaller upper bounds on their coefficients, otherwise they will tend to have higher R^2 values.

These premises are intended to create a definition of what types of true models would be considered realistic. The goal is to achieve a balance point where chosen models have coefficients that are large enough to be detected most of the time, but not so large as to overwhelm the error variance.

The implementation of these premises into a model-generating process is summarized in Pseudocode 7. The value of τ is first established; PB_{12} and PB_{20} designs use $\tau = 4$ and $\tau = 6$, respectively. The error variance is set to $\sigma^2 = 1$ for all simulations. Model size is selected uniformly from $(1, 2, \ldots, \tau)$. All coefficients, including the intercept, are assigned values sampled uniformly from (LB,UB), where LB and UB are the appropriate lower and upper bounds selected from Table 5.1. The coefficients are then assigned to be positive or negative with equal probability.

The bounds in Table 5.1 were chosen as follows. The lower bound is fixed for all model sizes. LB is the value for which the power to declare each coefficient active, when the true model is known, is 80% (using one-at-a-time t-tests, with $\alpha = 0.05$). The upper bound depends on model size. It is the value of UB for which, when coefficients are sampled uniformly on (LB,UB), the average R_a^2 equals 0.8 (appropriate upper bounds for $E[R_a^2] = 0.9$ are also given for reference)[1].

The need to adjust the upper bound for different model sizes is worthy of particular note. The restriction on very large R^2 values actually puts serious limitations on coefficient magnitudes when the model contains several variables. Consider, for example, two potential true models from a PB_{12} design, with $\sigma^2 = 1$ and all coefficients set to 2σ: $E[\boldsymbol{Y}] = \mathbf{1} + 2\boldsymbol{Z}_1$ and $E[\boldsymbol{Y}] = \mathbf{1} + 2\boldsymbol{Z}_1 + 2\boldsymbol{Z}_2 + 2\boldsymbol{Z}_3 + 2\boldsymbol{Z}_4$. The first model has $E[R^2] = 0.83$, while the second model has $E[R^2] = 0.97$. To ensure that true models of different sizes have about

[1] All of the UB values were extracted from $\overline{R_a^2}$-vs.-UB curves generated by simulation. Adjusted R^2 values were used to put models of different sizes on more equal footing.

Pseudocode 7 *Random generation of a model.*

Fix $\sigma^2 = 1$.

Let s be the model size. Choose s uniformly on $(1, 2, \ldots, \tau)$. $\tau = 4$ for PB_{12} $\tau = 6$ for PB_{20}.

Choose the particular variables to be in the model:

Let q be the number of main effects in the model. Select q from $1, \ldots, s$, with weights based on the population proportion of models with q main effects and size s.

Choose the q main effects at random from the available design columns.

Randomly choose $s - q$ interaction terms from the interactions that respect effect heredity.

Set LB, the lower bound for coefficient values. LB = 0.9 for PB_{12}, LB = 0.66 for PB_{20} (values are based on 80% power for one-at-a-time t-tests with $\alpha = 0.05$).

Choose UB, the upper bound for coefficient values, from Table 5.1 (values are based on requirement for average $R_a^2 = 0.8$).

Sample $|\beta|$ values from U(LB,UB).

Set the sign of each coefficient to +/- with equal probability.

the same goodness-of-fit on average, the coefficients of larger models must be constrained to smaller values. For the case of the PB_{20} design, for example, models of size six may only have coefficients in (0.66, 1.0)—a narrow range.

Table 5.2 contains ten models randomly generated from the process described here. The power to detect all coefficients active is given, along with the 10^{th} percentile, mean, and 90^{th} percentile of the distribution of R^2 for each model. The models shown exemplify some characteristics of the chosen model-generating process. First, models of all sizes will have R^2 values ranging from moderate to quite high, but with values greater than 0.95 being fairly rare. The trade-off made to achieve this is that larger models have smaller coefficients; consequently, the power to successfully detect all active variables in larger models can be low. The overall characteristics of the distribution of models suggest a challenging scenario for testing model selection methods.

Table 5.1: Bounds for coefficient values for randomly-generated models. Coefficient magnitudes are sampled uniformly on (LB, UB).

	Upper Bound, UB			
	PB_{12} (LB=0.9)		PB_{20} (LB=0.66)	
Model Size	$E[R_a^2] = 0.8$	$E[R_a^2] = 0.9$	$E[R_a^2] = 0.8$	$E[R_a^2] = 0.9$
1	3.6	7.8	4.5	9.4
2	1.9	3.4	2.2	3.9
3	1.4	2.5	1.6	2.8
4	1.1	2.1	1.3	2.3
5			1.1	2.0
6			1.0	1.8

Table 5.2: Example of ten random models from the model-generating process. PB_{20} design, main effects named $A - S$.

Model	**Pwr**	10^{th}	$E[R^2]$	90^{th}
$-3 - 1.34F$	1	0.681	0.773	0.861
$2.43 + 0.98E$	1	0.511	0.654	0.784
$-0.71 + 0.66A + 0.90F + 0.66Q + 0.86EF + 0.75FJ$	0.55	0.849	0.898	0.943
$-1.06 - 0.66Q - 0.92FQ + 0.90GQ - 0.95NQ - 0.80QR$	0.76	0.860	0.905	0.949
$1.61 - 0.67M + 2.17GM$	0.92	0.879	0.917	0.951
$-0.68 + 0.70F - 0.72L - 0.97N + 0.71BL + 0.87FH - 0.75FL$	0.53	0.816	0.879	0.937
$-0.78 - 0.67R - 1.87MR$	0.94	0.847	0.897	0.941
$-1.27 - 3.9N$	1	0.959	0.969	0.982
$0.89 - 0.84F + 0.74J - 0.91K + 0.68P - 0.99S - 0.73FM$	0.58	0.880	0.921	0.959
$-4 - 2.16G$	1	0.861	0.902	0.942

Table 5.3: Factors and levels for the parameter sensitivity study.

Parameter	Levels
p	6, 7, 8 (for PB_{12}) 7, 8, 9 (for PB_{20})
n_{gen}	1000, 5000, 10000
κ	2, 4, 8
P_{min}	0.001, 0.01

5.2 Study 1: Parameter Sensitivity

The first simulation investigates how sensitive the new method is to the choices of its adjustable parameters. The parameters in question are p, the number of variables in the oversized models; n_{gen}, the number of accepted models to generate; and ρ, κ, and P_{min}, the simulated annealing temperature control parameters. Recall from Section 3 that ρ is the factor by which the temperature is cooled on every accepted move, κ is the number of rejections required to balance one acceptance, and P_{min} is a lower limit on the acceptance probability.

The cooling fraction was left at its default value of $\rho = 0.95$ and not investigated further here. This default value had been used successfully throughout development of the method, so it was left fixed to reduce the number of possible parameter combinations. The remaining four parameters were assigned the levels given in Table 5.3. The parameters p, n_{gen}, and κ were given three levels, and the bound P_{min} was given two levels. All level combinations were tested in a $3^3 2^1 = 54$ run factorial design.

For comparing the 54 level combinations, one hundred models were generated at random using the model-generating rules of the previous section. A single $\boldsymbol{y}$-vector was then generated from each model. These hundred response vectors made up the test set for each run in the factorial design. For each run, model selection was performed (using the automated method of Section 3.4) on all of the response vectors, and the results were saved.

The result of each model selection was classified based on the type of model chosen. A chosen model may be assigned membership in one of the five sets discussed in Section 2.1.1: the true model, overfitted models, underfitted models, partial-truth models, or wrong models. As before, these sets are abbreviated $\mathcal{T}$, $\mathcal{O}$, $\mathcal{U}$, $\mathcal{P}$, and $\mathcal{W}$.

The idea behind this experiment is that if the model selection process is insensitive to the parameter choices, the same models should be chosen for almost all of the 100 cases, regardless of the experimental run. So all 54 runs should have approximately the same distribution of chosen models across the sets $\mathcal{T}$, $\mathcal{O}$, $\mathcal{U}$, $\mathcal{P}$, and $\mathcal{W}$. Any runs with very different proportioning of the models among these sets is a sign of parameter sensitivity.

The experiment was conducted for both the PB_{12} and PB_{20} cases. Results are displayed in Figure 5.1. In the figure, each bar corresponds to one experimental run, and the bars are divided to show the proportion of chosen models falling into each of the five outcomes. The bars have been sorted in order of increasing proportion of correct guesses (increasing numbers of choices in $\mathcal{T}$). The results for the two cases agree with one another. In the following discussion, let T, O, U, P, W represent the proportions of selected models that are members of the sets $\mathcal{T}, \mathcal{O}, \mathcal{U}, \mathcal{P}, \mathcal{W}$.

The proportion of selected models that are either true or overfitted (T+O) remains fairly constant across all of the runs. Results in $\mathcal{T}$ or $\mathcal{O}$ represent good outcomes, as the truth is within the selected model. The sum T+O is approximately 60% for the PB_{12} design, and about 45% for the PB_{20} design. The proportion of wrong choices, W, also remains fairly stable in the 5–15% range. The remainder of the models are divided among the underfitted and partial-truth models.

Although the sum T+O is roughly constant across runs, the relative size of T and O does vary considerably. This variation is the main feature of Figure 5.1 requiring explanation. A partial explanation can be found by inspection of the level combinations in the Figure. For both types of design, the runs with the low level of P_{min} (0.001 instead of 0.01) tend to dominate the left half of the Figure. For the PB_{12} design, 19 of the 27 runs with lowest values of T had $P_{min} = 0.001$; for the PB_{20} case, 24 of the 27 worst results involved $P_{min} = 0.001$. In addition, the runs having the very lowest values of T are those combining the lowest number of generated models ($m = 1000$) with the low level of P_{min}. The temperature control parameter κ also seems to have an interaction with P_{min}. The only runs for which $P_{min} = 0.001$ that have high T values are ones in which κ is also at its lowest level.

The above observations all point to the idea that *increasing the diversity of the simulated annealing search increases the chances of selecting the true model.* Search diversity refers to the extent to which the search covers the model space. Note that a higher value of P_{min} will result in more bad moves being accepted, and that a lower value of κ will cause the search to spend less time in locally-good areas. Both of these will increase search diversity.

Increasing the value of n_{gen} increases search diversity directly, by extending the duration of the search. The results in Figure 5.1 suggest that most combinations of parameter choices will perform well, but one should avoid making P_{min} small, n_{gen} small, and κ large at the same time.

Two final observations are in order from the sensitivity results. The first is that the model size, p, does not appear to strongly influence the outcome over the range of values tested. This is a reassuring result, as it is difficult to come up with a formal basis for the choice of p. The present results suggest that it should be sufficient to allow users to choose p on a case-by-case basis, guided by effect sparsity.

The second observation to make is that the effect of parameter choices on T and O should be even less when model selection is done by a human with the aid of the raster plot and link plot. The results of the simulation study necessarily depend on the automatic model extraction algorithm, and it is likely that this algorithm can be fooled more easily than a human operator. The fact that the sum T+O is nearly constant suggests that the link plots, for example, will look similar across the different runs.

5.3 Study 2: Performance

The second simulation compares the performance of different model selection methods. In each iteration of this simulation, a true model was generated from the process described in Section 5.1, and a realization of $\boldsymbol{Y}$ was generated from the true model. Model selection was then performed using different methods. For each method's chosen model, two values were recorded: the number of correct variables (n_c) and the number of wrong variables (n_w) selected. From n_c and n_w, each selected model was assigned membership in one of the possible outcomes: underfitted ($\mathcal{U}$), true ($\mathcal{T}$), overfitted ($\mathcal{O}$), partial-truth ($\mathcal{P}$), and wrong ($\mathcal{W}$). This process was repeated 5000 times each for the PB_{12} and PB_{20} designs. The maximum acceptable truth size, τ, was set to four for the PB_{12} case and six for the PB_{20} case. Four model selection methods were compared:

Oracle. This "method" requires that the true model be known in advance (the name is taken after Fan and Li 2001). Coefficients of the truly-active variables were tested for significance, using Bonferroni-corrected t-tests with a groupwise significance level of 0.05. The variables deemed significant by this test were returned as the chosen model.

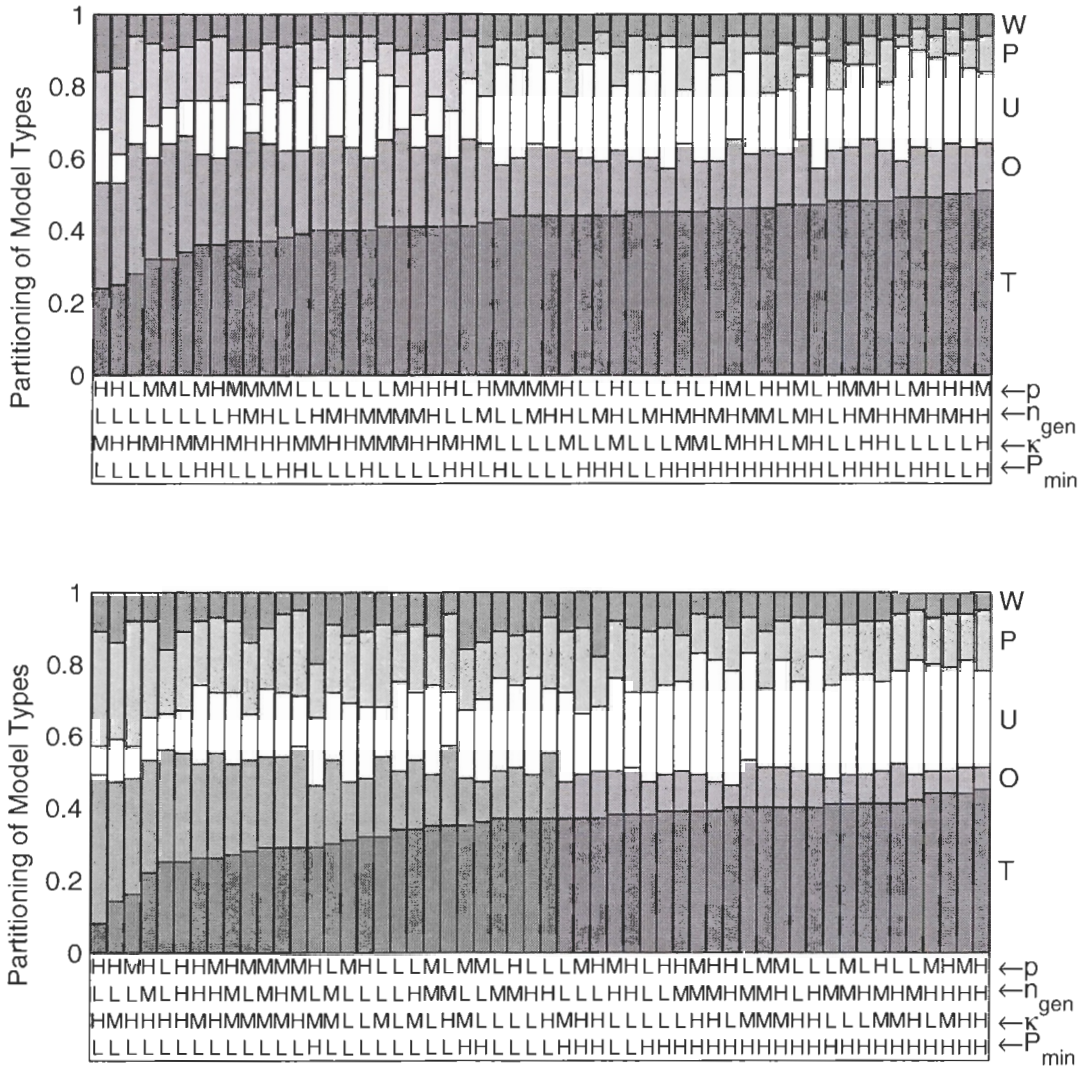

Figure 5.1: Results of the parameter sensitivity simulation, for the PB_{12} design (top) and the PB_{20} design (bottom). Each bar gives the proportion of chosen models that fell into the sets $\mathcal{T}, \mathcal{O}, \mathcal{U}, \mathcal{P}, \mathcal{W}$, for the same 100 truth scenarios. The experimental runs have been sorted in increasing order of true-model-selection frequency. Factor settings are indicated by L, M, and H for low, medium, and high levels.

Note that this method can only choose the true model or an underfitted model, since only the truly-active variables are considered.

SAMS. The new oversized-model sets method is abbreviated SAMS, for Simulated Annealing Model Search. The method was carried out as described in Section 3. The automatic model-extraction method using branch-and-bound and the entropy measure was used to choose a single best model. The parameters were set to $m = 10000$, $\rho = 0.95$, $\kappa = 4$, and $P_{min} = 0.01$. The model size p was set to seven for the PB_{12} design and eight for the PB_{20} design.

Stepwise. The hybrid stepwise approach of Wu and Hamada (Pseudocode 2) was used as a representative testing-based method. The default stepwise control parameters ($P_e = P_d = 0.05$) were used.

AICc. A search-plus-criterion method based on AIC_c was included in the study. Model search was done differently for the two designs. For the PB_{12} simulation, exhaustive search was used. For the PB_{20} simulation, exhaustive search was not practical, so two search heuristics were employed. First, the two-stage search of Pseudocode 3 was run. To improve the chances of finding the true minimum-AIC_c model, output from the SAMS algorithm was also used. Recall that the branch-and-bound search in the good model set returns the five most frequent models of size one through τ in the model set. These models all tend to have good fit for their size, and they are already available from the SAMS output. So the AIC_c criterion was also evaluated for these models, and compared to the output of the two-stage search. The model with minimum AIC_c was returned as the chosen model.

The most comprehensive way to study the simulation results is to review tables of the distribution of (n_c, n_w) for each selection method and each true model size. Such tables are included in Appendix C; only selected summary tables are shown below to illustrate the main results of the study.

Oracle performance

The oracle method provides a reasonable limit on how well a model selection method can be expected to perform. The method starts with the active variables already known; the true model is selected as long as all of its coefficients are deemed statistically significant

Table 5.4: Distribution of number of correct variables chosen by the oracle method, for each size of true model. Values are in percent.

PB_{12} Design					
True	n_c				
Size	0	1	2	3	4
1	1	99			
2	1	12	87		
3	5	14	29	52	
4	18	20	22	23	17

PB_{20} Design							
True	n_c						
Size	0	1	2	3	4	5	6
1	0	100					
2	0	5	95				
3	0	2	17	81			
4	0	2	10	33	55		
5	1	5	11	22	30	31	
6	3	7	13	20	23	21	13

for the given $\boldsymbol{y}$. The method employs a Bonferroni correction, to limit the probability of erroneously declaring an inactive coefficient active. In the present case all variables are known to be active, so the correction actually makes oracle performance worse on larger models. This problem is exacerbated by the fact that larger models are constrained to have smaller coefficients to satisfy the R^2 requirement.

The performance of the oracle method is displayed in Table 5.4. The table shows, for each true model size s, the percent of cases where $1, \ldots, s$ of the variables were declared active. The results confirm that performance is very good for small true models, but steadily decreases with model size, becoming very poor for the largest models considered. For both experimental designs studied, essentially all of the single-variable models were correctly identified. The success rate dropped to only 17% for models of size four in the PB_{12} case, and 13% for models of size six in the PB_{20} case.

The performance of the oracle method confirms that the chosen model-generating scheme provides a challenging test for the other methods. If even the oracle underfits, one can expect considerable model selection uncertainty to exist when the other predictors are added to the problem.

Average performance, all model sizes

The other model selection methods can be compared to each other and to the oracle by pooling results for all different true model sizes and tabulating the average performance. Table 5.5 summarizes performance by giving the percentage of cases where each method

chose a model in each of the five outcome categories $\mathcal{T}, \mathcal{O}, \mathcal{U}, \mathcal{P}, \mathcal{W}$. These percentages are given for three divisions of the data: first, for all 5000 cases; second, for only those cases in which the oracle successfully chose the true model; and third, for only those cases in which the oracle did *not* choose the true model.

Considering first the all-models case, one observation is immediately clear from Table 5.5: the stepwise and AIC_c methods predominantly overfit, while the SAMS method is much more likely to choose the true model or an underfitted one. SAMS chooses the true model 43.4% of the time for the PB_{12} design, and 34.7% of the time for the PB_{20} design. These numbers are quite high, considering the oracle method chooses correctly under 65% of the time for each design. Both stepwise and AIC_c methods have much lower rates of picking the truth, particularly for the larger PB_{20} design, for which both methods choose correctly under 2% of the time. The low rates of picking the truth for stepwise and AIC_c are explained by the fact that both methods are very prone to the inclusion of spurious variables. Both methods have $\mathcal{O}$ and $\mathcal{P}$ frequencies much higher than SAMS. It should be noted that although the AIC_c and stepwise methods show similar patterns in their results, AIC_c has generally better performance than stepwise. In particular it is much less likely to choose a completely wrong model.

Selection of an overfitted model may be considered an acceptable result in some situations, since all of the truly-active variables will be identified among the chosen subset. It is reasonable, then, to compare the totals in the $\mathcal{T}$ and $\mathcal{O}$ columns of Table 5.5. Over the PB_{12} runs, this total is 59.7% for SAMS; 52.7% for stepwise; and 62.3% for AIC_c. Over the PB_{20} runs, the corresponding totals are 45.8%, 45.0%, and 53.3% for the three methods. So if overfitted models are considered equally desirable as identification of the true model, then the three methods are quite similar, with AIC_c having a slight edge (though this edge may be primarily due to the automatic model selection process used in the simulation; this will be discussed further in Section 5.5).

The second and third sets of sub-tables in Table 5.5 compare the performance of the three competing methods conditional on the success or failure of the oracle method. The motivation for dividing the data in this way is that oracle-failed cases may be too challenging to expect any model selection method to work well; conversely, oracle-successful cases should have less model selection uncertainty and thus the methods should perform better. The table shows that there is indeed a major difference between these two scenarios. When the oracle is not successful, the rates of picking the true model, or even an overfitted one, are very

Table 5.5: Percentages of selected models falling into the sets $\mathcal{U}, \mathcal{T}, \mathcal{O}, \mathcal{P}, \mathcal{W}$, for both designs. Results are given for all methods: oracle (Or), stepwise (St), SAMS (SA), and AIC_c (A).

PB_{12} Design						PB_{20} Design					
All 5000 Models						All 5000 Models					
	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$		$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$
Or	36.4	63.6	0.0	0.0	0.0	Or	37.9	62.1	0.0	0.0	0.0
St	4.0	10.6	42.1	16.5	26.8	St	0.5	1.9	43.1	36.9	17.7
SA	15.9	43.4	16.3	15.1	9.3	SA	30.1	34.7	11.1	17.2	6.9
A	0.7	7.4	54.9	25.0	12.0	A	0.0	0.6	52.7	40.3	6.5
3180 Oracle-Successful Models						3107 Oracle-Successful Models					
	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$		$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$
St	1.0	15.7	62.7	4.8	15.8	St	0.1	3.0	67.1	19.5	10.3
SA	4.3	66.7	24.0	2.5	2.5	SA	18.4	55.1	17.5	6.9	2.2
A	0.0	10.1	81.4	4.5	4.0	A	0.0	0.8	79.8	17.0	2.4
1820 Oracle-Failed Models						1893 Oracle-Failed Models					
	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$		$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$
St	9.2	1.8	6.2	37.0	45.9	St	1.2	0.1	3.6	65.4	29.7
SA	36.2	2.8	2.7	37.1	21.2	SA	49.3	1.3	0.5	34.2	14.7
A	1.9	2.7	8.7	60.7	26.0	A	0.0	0.3	8.2	78.4	13.2

low---the large majority of choices by all methods are underfitted, partial-truth, or wrong models. When the oracle method is able to identify the truth, all of the methods perform well. The new SAMS method is able to identify the true model in two thirds of the oracle-successful cases for the PB_{12} design, and 55% of the cases for the PB_{20} design. The other two methods see little improvement in their ability to identify the truth, but the proportion of times an overfitted model was found did increase significantly.

Performance for different model sizes

Table 5.5 is a useful summary that illustrates the main differences among the methods, but it omits certain important information. It is an average across all model sizes in the experiment, so that no model size effects are observed. It also combines all over- and under-fitted models together, treating, for example, a model overfitted by three variables the same as a model overfitted by only one variable.

The information in Table 5.6 helps to understand the simulation output in more detail. The table contains the same distribution of $\mathcal{T}, \mathcal{O}, \mathcal{U}, \mathcal{P}, \mathcal{W}$ outcomes as before, but now with

separate sub-tables for each true model size. Each design case has tables for size one up to its maximum true-model size, τ. Each sub-table also contains two extra columns, one for the average number of correct variables selected ($\overline{n_c}$) and one for the average number of wrong variables selected ($\overline{n_w}$).

Inspection of Table 5.6 reveals some details that are not evident in the marginal tables of the previous section. The first observation is that the number of wrong variables selected by AIC_c is high. This agrees with the discussion of Section 2.2.2, which suggested that criterion-based methods will become more prone to overfitting when the model set gets larger. For the PB_{12} design, with $\tau = 4$, the AIC_c-best model found is size four 80% of the time, and size three 19% of the time. The chosen model has size one or two in only 60 of 5000 trials. For the PB_{20} case, with $\tau = 6$, the situation is even worse; a model of the maximum size, six, is selected in over 97% of the cases. There would be no practical difference between using the criterion and simply searching for the lowest-RSS model of size τ.

The stepwise method also has high values for $\overline{n_w}$, indicative of a strong tendency to include spurious variables. By contrast, the SAMS method has much lower average rates of selecting incorrect variables. Taking true models of size three as an example, SAMS includes an average of 0.7 false variables for the PB_{12} iterations, while AIC_c included 1.7 false variables on average. For the PB_{20} case these numbers were 0.2 variables for SAMS and 3.4 for AIC_c.

The tendency of SAMS to avoid spurious variables results in very high rates of selecting the true model when the true model is small, say up to size three (the 75% rate of finding a true model of size two for the PB_{20} design is particularly notable). As the true models get larger, however, the new method becomes worse at detecting the truth and tends to underfit or choose partial-truth models quite often. For example, when the design is PB_{12} and the truth is size four, SAMS chooses the true model only 5% of the time, and chooses models in $\mathcal{U}$ and $\mathcal{P}$ 33% and 39% of the time, respectively. For PB_{20} models, the method chooses an underfitted model about 50% of the time when the truth has size four, five, or six.

Cases with larger true models provide the only situations where AIC_c or stepwise appear to have an advantage over the new method. Because these methods will always pick well-fitting large models, they are well suited to catch more of the active variables when the truth is indeed large. Considering models of size five in the PB_{20} simulation, for example, SAMS

chooses an average of 1.9 correct variables, while AIC_c has an average of 2.4—which is better, but still quite poor. The usefulness of SAMS in such difficult cases will be discussed further at the end of this chapter.

5.4 Additional Results

The performance and sensitivity simulations have addressed a number of the biggest questions about the new model selection method. A few remaining points, not covered previously, are addressed below.

Performance on the null model

The performance simulation only considered true models of sizes 1–τ. The null model (with only an intercept, size zero) was intentionally left out of the simulations. It is generally considered desirable for a statistical method to have controlled type I error for the null case—so that, for example, the null model is correctly identified $100(1-\alpha)\%$ of the time. This requirement will be very difficult to achieve when the model set is huge, however. Regardless of the $\boldsymbol{y}$-vector that happens to be observed, it is virtually certain that at least one variable combination will fit the data well enough to reject the null-model hypothesis. Though this issue has not been thoroughly explored, it is felt that tuning a variable selection procedure to control type I error under the null (for example, by using a selection criterion with a large complexity penalty) would result in extremely low power to detect non-null models when they do exist.

A more pragmatic reason not to be concerned about the null case is that in applications, the null model is almost never the best approximation of the truth. In industrial experiments, candidate variables are usually carefully selected using considerable process knowledge. It is very rare, therefore, for none of the variables in a screening experiment to have any effect.

A very small simulation was run to illustrate the difficulty in identifying the null model. The PB_{20} performance simulation code was run for 250 additional iterations, but with all realizations of $\boldsymbol{Y}$ generated from the null model. The distribution of chosen-model sizes is shown in Table 5.7 for stepwise, AIC_c, and SAMS. Stepwise was the only method that ever chose the null model, doing so for only four out of 250 cases. Stepwise and AIC_c both chose a model of the largest admissible size most of the time—96.8% of the time, in the case of

Table 5.6: Percentages of selected models falling into the sets $\mathcal{U}, \mathcal{T}, \mathcal{O}, \mathcal{P}, \mathcal{W}$, for both designs. Results are stratified by true model size. The average number of correct variable choices ($\overline{n_c}$) and the average number of wrong choices ($\overline{n_w}$) are also given.

PB_{12} Design

1240 models of size 1

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	1.4	98.6	0.0	0.0	0.0	1.0	0.0
St	0.0	22.7	74.8	0.0	2.6	1.0	1.9
SA	0.0	56.6	42.3	0.0	1.1	1.0	0.7
A	0.0	0.1	99.0	0.0	1.0	1.0	2.8

1283 models of size 2

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	13.5	86.5	0.0	0.0	0.0	1.9	0.0
St	1.8	11.6	67.4	5.8	13.3	1.7	1.7
SA	6.5	69.3	18.7	4.0	1.5	1.9	0.3
A	0.0	2.4	85.3	8.9	3.4	1.8	1.9

1219 models of size 3

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	47.7	52.3	0.0	0.0	0.0	2.3	0.0
St	4.4	5.0	25.8	24.9	39.9	1.3	1.9
SA	24.0	42.5	4.1	18.1	11.2	2.1	0.7
A	0.2	16.7	34.8	32.4	15.8	2.0	1.7

1258 models of size 4

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	83.4	16.6	0.0	0.0	0.0	2.0	0.0
St	9.6	3.2	0.0	35.5	51.7	0.8	2.1
SA	33.1	5.0	0.0	38.5	23.4	1.4	1.3
A	2.5	10.8	0.0	58.7	28.0	1.4	2.4

PB_{20} Design

833 models of size 1

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	0.4	99.6	0.0	0.0	0.0	1.0	0.0
St	0.0	6.5	91.6	0.0	1.9	1.0	4.1
SA	0.0	52.9	46.3	0.0	0.7	1.0	0.7
A	0.0	0.0	98.7	0.0	1.3	1.0	5.0

819 models of size 2

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	5.3	94.7	0.0	0.0	0.0	1.9	0.0
St	0.2	2.6	86.0	7.4	3.8	1.8	3.7
SA	4.9	75.1	15.9	3.7	0.5	1.9	0.3
A	0.0	0.0	90.2	8.2	1.6	1.9	4.1

852 models of size 3

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	19.5	80.5	0.0	0.0	0.0	2.8	0.0
St	0.2	0.9	55.8	27.8	15.3	2.1	3.6
SA	25.4	58.0	4.1	10.4	2.1	2.5	0.2
A	0.0	0.0	69.8	27.5	2.7	2.6	3.4

831 models of size 4

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	45.5	54.5	0.0	0.0	0.0	3.4	0.0
St	0.5	0.8	21.4	55.2	22.0	1.9	3.8
SA	52.3	21.2	0.4	19.9	6.3	2.5	0.4
A	0.0	0.0	44.4	49.3	6.3	2.9	3.1

815 models of size 5

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	68.8	31.2	0.0	0.0	0.0	3.7	0.0
St	1.0	0.0	4.0	64.4	30.6	1.4	4.2
SA	52.9	1.2	0.0	31.7	14.2	1.9	0.8
A	0.0	0.6	13.4	73.7	12.3	2.4	3.6

850 models of size 6

	$\mathcal{U}$	$\mathcal{T}$	$\mathcal{O}$	$\mathcal{P}$	$\mathcal{W}$	$\overline{n_c}$	$\overline{n_w}$
Or	87.3	12.7	0.0	0.0	0.0	3.7	0.0
St	1.2	0.5	0.0	66.1	32.2	1.3	4.2
SA	45.1	0.0	0.0	37.4	17.5	1.5	1.0
A	0.0	2.8	0.0	82.5	14.7	2.1	3.8

Table 5.7: Distribution of chosen model sizes under the null model, PB_{20} case. Values are in percent.

Size	Stepwise	AIC_c	SAMS
0	1.6	0.0	0.0
1	2.0	0.0	8.8
2	6.8	0.0	38.0
3	6.4	0.0	34.4
4	5.2	0.0	16.0
5	4.4	3.2	2.8
6	73.6	96.8	0.0

AIC_c. As in the non-null cases, SAMS overfitted less than the other methods, commonly choosing models of sizes two through four.

The results here suggest that if correct identification of the null model is important, none of the methods considered is particularly useful. For the new SAMS method as implemented here, it is actually impossible to choose a model of size zero. For the stepwise algorithm and AIC_c it is technically possible to choose the null model, though in practice these methods almost always overfit the null model even more than SAMS does. Controlled error rate under the null model is a desirable property, but one that is hard to achieve in the screening case, where n is small and the number of candidate models is very large.

Repeatability

The SAMS algorithm involves a stochastic search heuristic, so repetitions of the model selection process will not result in identical good-model sets. Ideally, the repeatability of the method would be sufficiently good that raster plots and link plots look the same run-to-run, and that the maximum-entropy model selected does not change.

To assess the repeatability of the method, the SAMS method was carried out twice for the first 1000 runs in the PB_{12} and PB_{20} performance simulations. Two outputs were recorded for each design, with the results shown in Table 5.8. The first output, % same, is the percent of cases where the same model was selected in both trials. The second output, % top 5, is the percent of cases where the best model from the first trial was in the top five models in the second trial. As before, results are shown for all cases, and then conditional on the outcome of the oracle method.

Table 5.8: Results of repetitions of the SAMS method.

	PB_{12} Design		PB_{20} Design	
	% same	% top 5	% same	% top 5
All models	85	94	77	89
Oracle-successful	89	95	82	91
Oracle-failed	78	93	69	85

The table shows that repeatability is good, with the same model being chosen about 90% of the time. The number predictably drops when the model space is bigger (the PB_{20} design) and when model selection uncertainty is higher (the oracle-failed cases). Experience has shown that for typical cases, the raster plot and link plot do not change their qualitative appearance very much on repetition.

Run time

Computer run times for the SAMS algorithm, with the default parameter settings, are very reasonable. For the PB_{12} design, typical run times are 10-15 seconds for the simulated annealing search, and less than two seconds for the branch-and-bound algorithm. Run times are somewhat longer for the PB_{20} design, with the model search taking approximately 20 seconds, and another 20 seconds for the branch-and-bound search. The branch-and-bound method is very sensitive to the problem size and to the quality of the initial guess solutions. For considerably larger problems where good models are not clearly evident, the branch-and-bound time could become prohibitive.

The relatively fast run time for SAMS means that the user has options in applying the method. For example, the size of the good-model set could be increased considerably, or the search could be repeated with alternative parameter settings. When there is strong evidence for the best model, such changes should have little effect on the results; but if model selection uncertainty is high, some differences may be observed. The method lends itself well to an interactive approach.

5.5 Comments on Underfitting and Overfitting in the New Method

The results of the parameter sensitivity study and the performance study seem to point to some areas of concern regarding underfitting and overfitting. The sensitivity study suggests that changing the simulated annealing paramters can strongly influence the balance between under- and over-fitting; and the performance study suggests that SAMS tends to underfit when the true model is large. These concerns originate primarily in the entropy criterion and the simulation study, however. It is important to remember that the new method is not a model selection criterion. The necessity of picking a single model forced criterion-like properties onto the method in the simulation studies.

Many traditional model selection methods are based in the notion of optimality—the idea that if only an appropriate metric or algorithm could be found, a single optimal model could be identified in every situation. The new method abandons this idea at the outset, in the face of the huge model set. It is much more similar in spirit to Bayesian variable selection, which tries primarily to report what the data says, including the possibility of model selection uncertainty. Both Bayesian variable selection and the oversized-models approach can be thought of as decision support tools. When the best model is obvious, the method finds it; the SAMS method was much more proficient at choosing the true model than its competitors in the performance simulation. When there is uncertainty about the best model, the method returns information that is useful in deciding what to do next. It should be noted that criterion-based methods generally do *not* provide this extra information.

Inspection of the complete output of the oversized-models method is necessary to fully appreciate its usefulness in cases of model selection uncertainty. The method provides (through the branch-and-bound search) a list of good candidate models of each size, but it also provides graphical summaries in the form of the raster and link plots.

An example can help to illustrate how the new method provides more information than just the best-guess model. A true model of size six was randomly generated from the PB_{20} design:

$$E[Y] = 0.93 - 0.67B - 0.76F + 0.70H - 0.96S - 0.86BI - 0.69FJ.$$

This model has $E[R^2]$ of 0.934, but because of several small coefficients, the power for the oracle method to successfully detect all six active variables is only 0.31. This is a challenging example for model selection.

A response vector was generated from the chosen model, and the SAMS method with automatic best-model choice selected the three-variable model (B, S, BI). All three variables are in the true model: the selected model is underfitted by three variables. For the same data, the AIC_c-best model found was (B, F, G, H, S, BI). This model contains five of the six truly-active variables plus one spurious main effect, G. So by strictly comparing the two best-guess models, the AIC_c method appears to do better.

The link plot of the good-model set is shown in 5.2. It is clear from the plot how (B, S, BI) was chosen as the best-guess model; but the other two variables found by AIC_c, F and H, are also visible as possibly important. The only variable not detected by the plot is the FJ interaction, which was not found by either method.

The AIC_c-best model (B, F, G, H, S, BI) was also the most frequent combination of six variables in the SAMS good-model set. In this and many other cases, the best AIC_c model was found by searching the SAMS most-frequent model list, *not* by using the two-stage search method of 2.2.2. This suggests that, if one wanted to replicate the performance of AIC_c using the new method, one could just take the most frequent model of the largest size found by branch-and-bound.

This example illustrates that as long as there is uncertainty in the data, the data analyst can never totally escape the problem of model size. The oversized-models method suggests a best model, but also provides the user with information they can use to select a larger or smaller model, depending on the particular situation.

An additional example can be extracted from the sensitivity simulation. In the PB_{12} experiment, run 12 $(p = 7, m = 1000, \kappa = 8, P_{min} = 0.01)$ selected overfitted models particularly frequently. One data set for which this run selected an overfitted model was randomly chosen. The true model for this case was (G, J); run 12 erroneously selected model (G, J, FJ). A different set of parameter values was then chosen at random from those runs that correctly identified the true model. The chosen run was run 47, which happened to differ from run 12 only by having a larger model set $(p = 7, m = 5000, \kappa = 8, P_{min} = 0.01)$. The raster plots, link plots, and entropy calculations were then generated for these two sets of parameter values, using the same $\boldsymbol{y}$-vector used in the sensitivity study. The results of this test are shown in Figure 5.3.

The figure shows that there is little difference between the results produced by the two sets of parameter choices. Run 12 chooses the overfitted model including FJ because it has slightly higher entropy: $H(G, J, FJ) = 2.38$, while $H(G, J) = 2.25$. Run 47, on the other

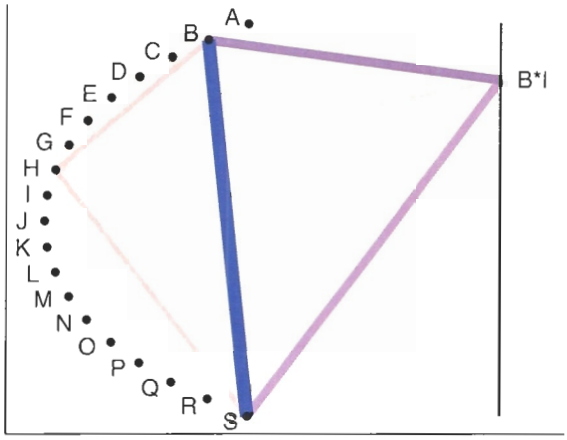

Figure 5.2: Link plot for an example where the new method chooses an underfitted model.

hand, has the situation reversed: $H(G, J) = 1.83$, and $H(G, J, FJ) = 1.79$. The simulation is forced to choose the single highest-entropy model as output, but in fact the two cases are practically equivalent. The equivalence is easily seen in the link plots, but cannot be translated into the entropy criterion.

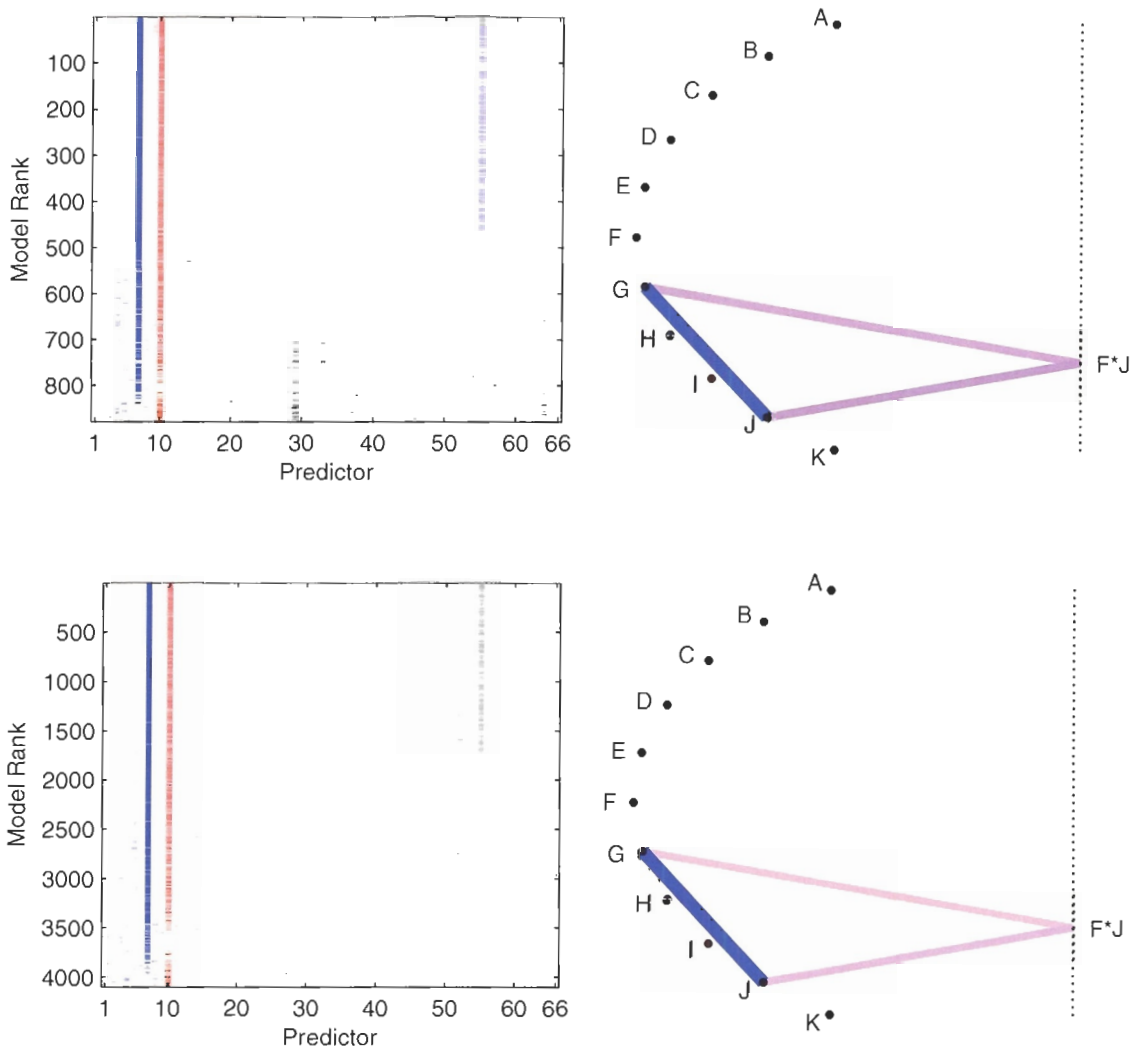

Figure 5.3: Raster plots and link plots for two runs in the sensitivity study. Run 12 (top) selected the overfitted model (G, J, FJ), while run 47 (bottom) selected the true model (G, J). A human user would likely consider the two cases equivalent from a decision making standpoint.

Chapter 6

Conclusions

A new model selection method has been presented. The proposed method takes an approach to the model selection problem that is motivated by the particular challenges of screening designs, and is somewhat different from traditional techniques.

6.1 Summary of the Method

The new method, called the method of oversized-model sets, is particularly suited to problems where the set of candidate models is too large to search exhaustively, and where a small heredity-respecting model is desired.

The method proceeds by first building a large set of well-fitting models of a fixed size p, larger than the anticipated best-model size. This good-model set is constructed using a stochastic search heuristic called simulated annealing model search (SAMS). The SAMS algorithm has been designed to provide a non-convergent search through only heredity-respecting models.

The premise of the method is that good smaller models can be found by inspecting the oversized-model set for common variable combinations. The raster plot, clustered raster plot, and link plot are suggested as graphical tools to aid this inspection. A branch-and-bound algorithm has also been employed to find the r most common variable combinations of each size in the good-model set.

Graphical outputs and a list of most common models should be sufficient for most real applications. As a final component, an entropy criterion was developed for cases where an objective, "hands-off" model selection is required. The criterion aims to find the most

over-represented model in the good set, based on a random-sampling reference distribution.

Simulation studies and examples showed that the new approach compares favourably to alternative methods. The key differences and advantages of the oversized-models approach are listed below.

Effect heredity is built into the procedure. The procedure is built on the premise that only models respecting (weak) heredity will be considered reasonable in screening experiments. By enforcing heredity throughout, model search is made more efficient, and the end results become easier to interpret.

Statistical testing and selection criteria are avoided. Methods based on significance testing or on model selection criteria will always suffer when the set of candidate models gets very large. The new method handles the huge model sets more gracefully by using information across a large number of well-fitting models.

Goodness-of-fit and the choice of model size are treated separately. In most traditional (frequentist) model selection methods, the search for a well-fitting model is coupled to the choice of model size. Indeed, the trade-off between fit and complexity is the central problem of model selection. The new method separates the two issues by searching for good models of only one fixed size. Having constructed the good-model set based on RSS, the fit of the models is never considered again. In subsequent steps, only the *structure* of the models is used to extract a best combination of variables.

Graphical tools are provided. The method of oversized-model sets is based on a simple concept, and its results can be gathered from graphical displays. This feature should greatly enhance ease-of-use and the quality of decision making. A very important feature of the graphics is that they also communicate the extent of model selection uncertainty much better than, say, a list of the five best-fitting models of each size.

6.2 Future Work

The oversized-models concept and the SAMS algorithm have been shown to work well in many situations. The work to date has pointed out several areas of room for improvement of the algorithms, and has also suggested some more general research questions.

Improvements to the existing algorithms

Considering first the SAMS algorithm itself, future work will focus on making it as robust and easy to use as possible, and making it applicable to a wider range of scenarios. There are several areas for further development:

1. Gain an improved understanding of the workings of the SAMS algorithm, specifically the roles of the control parameters n_{gen}, ρ, κ, P_{min}, and p. Simplify the parameter choices for the algorithm and make the search as robust and repeatable as possible.

2. Generalize the method to allow arbitrary specification of heredity relationships, including multiple-level relationships. Generalize the entropy criterion to match.

3. Investigate the applicability of the method across a wider range of cases. Examples could include: cases where the sparsity or heredity assumptions do not hold; cases with active three-way interactions; regression cases exhibiting strong multicollinearity; or cases involving hundreds of variables.

The first point above is worthy of further discussion. The parameter sensitivity study (Section 5.2) indicated that the method is fairly robust to different choices of the user-adjustable quantities. The results suggested, however, that there may be some redundancy among the parameters. Specifically, the P_{min} parameter can probably be eliminated through appropriate choice of the other parameters and some algorithmic changes. A deeper analysis of the temperature traces from different SAMS runs should help to understand the process better, and suggest simpler ways to control the algorithm.

One promising idea is to track the frequency with which the search is re-visiting recent models, and to use this information to help control the search. For example, random restarts could be triggered if the frequency of visiting duplicates is too high. Random restarts should have the same effect as including P_{min}, but in a way that is transparent to the user.

Additional research questions

The present work has also suggested some higher-level questions that could be addressed in future research. Three such questions are listed, and then discussed briefly, below.

1. How serious is the effect of model set size on the utility of model selection criteria?

Can model selection criteria be modified to account for the size of the candidate model set?

2. What is the best way to structure a model selection simulation study? What is the best way if one is not willing to accept coefficients exactly equal to zero? How can disparate methods such as SAMS, stepwise regression, and Bayesian variable selection be fairly compared?

3. Can heuristic search strategies be proposed as general optimization methods for searching model spaces subject to heredity constraints?

The fact that the size of the model set does indeed affect the performance of criteria appears to be little represented in the literature. The extent of this problem, ways of measuring it, and ways of correcting for it, could be subjects of ongoing research.

Attempts to create a meaningful simulation study suggested that the question of study design and analysis could itself be a problem for future research. Section 5.1 discussed the difficulty in creating a distribution of true models that yields realistic scenarios across a range of true model sizes. The common device of setting a few coefficients to large values while the rest are exactly zero is also somewhat questionable, from the standpoint of remaining true to the real world. There may be opportunities to create a better simulation framework, where small coefficients are allowed to exist. Alternative measures of model selection performance could be investigated, and model selection performance could be measured relative to some fair standard.

Combinatorial optimization methods such as simulated annealing, genetic algorithms, or tabu search have not frequently been applied to model selection problems, despite their good performance in other domains. Preference in the statistical literature has been for more simplistic algorithms such as stepwise regression or sequential replacement.

Implementation of the more advanced combinatorial optimization heuristics for model selection is generally straightforward—until constraints such as effect heredity are enforced. Dealing with heredity constraints requires modifications to the standard algorithms, as in the case of SAMS. The fact that SAMS works well suggests that the modified simulated annealing algorithm could be proposed as a generic tool for search through huge model sets. A first attempt at a heredity-respecting genetic algorithm has also been tried, with some success. Further work in this area could provide model search tools that would be useful regardless of the analyst's preferred model selection approach.

Appendix A

Counting Hereditary Models

Equation 2.3 (the number of hereditary models of a given size) and equation 2.5 (the number of hereditary models containing a particular subset) are derived below. These formulas are used in determining the reference probability for the entropy criterion of Section 3.4.

Note that both of the equations derived here apply only when the full matrix consists of w main effects and all of their 2-way interactions. Cases with additional predictor columns, or alternative heredity specifications, will need modifications to these formulas.

Number of Hereditary Models of a Given Size

> **Goal:** for a problem involving w main effects, determine $N_p(w)$, the number of heredity-respecting models of size p.

A heredity-respecting model must contain at least one main effect, and may contain up to p main effects. Let the number of hereditary models of size p that contain γ main effects be $N_p^\gamma(w)$.

Consider a model of size p with γ main effects. There are $\binom{w}{\gamma}$ ways to choose the main effects. The remaining $p-\gamma$ variables in the model are chosen from the admissible interactions. Each main effect in the model is involved in $w-1$ interaction variables. But for γ main effects, they will share $\binom{\gamma}{2}$ interactions with each other. So the number of admissible interactions is $\gamma(w-1)-\binom{\gamma}{2}$. The number of ways to form a model of size p with γ main effects is then

$$N_q^\gamma(w) = \binom{w}{\gamma}\binom{\gamma(w-1)-\binom{\gamma}{2}}{q-\gamma}.$$

Summing this over $\gamma = 1, \ldots, p$ gives the desired result:

$$N_p(w) = \sum_{\gamma=1}^{p} \binom{w}{\gamma} \binom{\gamma(w-1) - \binom{\gamma}{2}}{q-\gamma}.$$

The above equation is only valid when $p < w$, that is, when the model size is less than the number of main effects. The formula also assumes that when $x < y$, $\binom{x}{y} = 0$.

Number of Overfitted Models with a Given Structure

> **Goal:** Given a design matrix with w main effects, and a submodel consisting of a main effects and b interactions, determine $N_p(w, a, b)$, the number of heredity-respecting models of size p that contain the given submodel.

The given submodel contains $a + b$ variables; so $p - a - b$ variables must be added to this model to yield a hereditary model of size p. The goal is to determine the number of ways this can be done.

To build the submodel up to size p, one may add $0, 1, \ldots, p-a-b$ main effects, and the complementary number of interactions. Consider the case where γ main effects are added. There are already a main effects in the submodel, leaving $\binom{w-a}{\gamma}$ ways to select the added main effects. The remaining $p - a - b - \gamma$ added variables must be interactions, selected from the set of admissible interactions. Define the main effects in the original submodel as *old*, and the main effects added as *new*. Then the admissible interactions can be divided into two groups:

1. Interactions involving at least one of the a old main effects. All such interactions, other than the b that are in the given submodel, are admissible:

   Number of such interactions $= a(w-1) - \binom{a}{2} - b$.

2. Interactions involving only the γ new main effects, and not the old ones. There are $w - a - 1$ such interactions involving each new main effect:

   Number of such interactions $= \gamma(w-a-1) - \binom{\gamma}{2}$.

So the total number of admissible interactions is $a(w-1) - \binom{a}{2} - b + \gamma(w-a-1) - \binom{\gamma}{2}$. The $p - a - b - \gamma$ interactions are selected from these, along with the γ main effects, so:

$$N_p^{\gamma}(w, a, b) = \binom{w-a}{\gamma} \binom{a(w-1) - \binom{a}{2} - b + \gamma(w-a-1) - \binom{\gamma}{2}}{p-a-b-\gamma}$$

This equation can be summed over $\gamma = 0, 1, \ldots, p - a - b$ to give the desired result:

$$N_p(w, a, b) = \sum_{\gamma=0}^{p-a-b} \binom{w-a}{\gamma} \binom{a(w-1) - \binom{a}{2} - b + \gamma(w-a-1) - \binom{\gamma}{2}}{p-a-b-\gamma}$$

The above equation is only valid when $p > a + b$, $a \geq 1$, and $p < w$. The formula assumes that when $x < y$, $\binom{x}{y} = 0$.

Appendix B

Branch-and-Bound Algorithm

Branch and bound (hereafter, B&B) is a well-established combinatorial optimization algorithm. A good introduction can be found in Clausen (1999). It has the advantage of being a global optimization method that does not require the evaluation of every alternative. The problem in question must have two characteristics for B&B to be applicable:

1. It must be possible to exhaustively divide the solution space into a series of smaller and smaller subsets (branches).

2. At any point in the subdivision of the model space, it must be possible to calculate an upper bound (for maximization) or a lower bound (for minimization) for all solutions in the current branch.

When these two conditions are satisfied, the B&B method can search very large solution spaces efficiently. The speed of the search can also be greatly improved when good initial guesses can be found.

The discussion that follows relates the particular implementation of B&B for the purpose of extracting the most frequent submodels in the SAMS good-model set.

> **Goal:** given the m models in the good set $\mathcal{G}$, find the r most frequent submodels of size q.

Definitions:

Branch. At any point in the search, the current branch is defined by a particular submodel of size $1 \ldots q-1$. For example, the current branch may be (A, B). Then further subbranches would be (A, B, C), (A, B, D), and so on.

Leaf. A leaf is reached when branching has produced a model of the maximum size, q. Every leaf is a candidate solution to the problem.

Currently proposed branch, P. Let P be the branch that is currently under consideration at a given point in the search.

Occurrence frequency, $f(P)$. For a submodel (branch) P, let $f(P)$ be its frequency of occurrence in $\mathcal{G}$. The occurrence frequency is easy to evaluate for any given model.

Frequency to beat, f^*. Let M^* be the r^{th} most frequent leaf model found at the current point in the search. Then $f^* = f(M^*)$. Any future leaf found with higher occurrence frequency constitutes an improvement in the solution.

The set of live branches, $\mathcal{L}$. This is the set of branches that have not yet been explored or eliminated from the search. The algorithm moves through $\mathcal{L}$ until all branches are eliminated. Let $\mathcal{L}_i$ be the i^{th} element of $\mathcal{L}$.

Ordering of variables and branches:

A particular ordering of variables within a branch, and branches in $\mathcal{L}$ is required to avoid visiting any branches more than once. At any given time, $\mathcal{L}$ may be contain a mixture of models of different sizes. The following order is imposed:

- Consider all variables in the full matrix to be assigned a letter.
- Within a branch, all variables are in alphabetical order.
- Order the models in $\mathcal{L}$ in alphabetical order, with "space" being ordered before A.
- E.g. if there are only four variables and $q = 3$, the fourteen models in sorted order are (A), (A, B), (A, B, C), (A, B, D), (A, C), (A, C, D), (A, D), (B), (B, C), (B, C, D), (B, D), (C), (C, D), (D).

Choosing initial solutions

The nature of $\mathcal{G}$ makes it particularly suited to B&B since it is easy to get good initial solutions. There are typically only a small number of variables that have high frequency, so good models can be found by sequentially taking the most-frequent variables in the set. Intial solutions are obtained as follows:

1. Select the most frequently-occurring main effect.

2. Select the variable that respects heredity and occurs most frequently *among all models that contain the previously chosen variable(s).*

3. Repeat step 2, each time adding the conditionally-most-frequent variable, until the selected model has size q.

4. Disregard all of the models containing the previously chosen model(s).

5. Repeat steps 1–4 until r models have been thus chosen.

The B&B algorithm

The essence of the method, in the present case, is as follows. At any point in the search, the frequency of the r^{th} most frequent model is the current score to beat. For example, say that halfway through the search, the r^{th} most frequent model found occurs 1000 times in $\mathcal{G}$. A branch is defined by a particular subset of variables; for example, the current branch may be "all models containing (A, AB)." The occurrence frequency of the model defining the branch sets an upper bound for the frequency of any larger model. So if (A, AB) occurs, say, 500 times, then it is not necessary to search any other models containing (A, AB), as they cannot be more frequent than the current cutoff of 1000. This idea can be implemented systematically through the algorithm in Pseudocode 8.

Pseudocode 8 *The branch and bound algorithm.*

1. Choose r good initial guesses by the method described above. Let the lowest occurrence frequency among these initial models be f^*.
2. Set any variables with marginal frequencies $< f^*$ to inadmissible.
3. Set $\mathcal{L} = \{\text{All admissible main effects}\}$. Ensure that $\mathcal{L}$ is sorted.
4. While $\mathcal{L}$ is not empty:
   (a) Set P $= \mathcal{L}_1$
   (b) Set $\mathcal{L} = \mathcal{L} \setminus P$
   (c) If $f(P) < f^*$ then go to (a)
   (d) Else
   If P is a leaf
   Add P to the top r solutions and recalculate f^*
   Else
   Find $\mathcal{V} = \left\{\begin{matrix}\text{Variables that are admissible, respect heredity with} \\ \text{P, and are ordered lower than the last variable in P}\end{matrix}\right\}$
   Set $\mathcal{L} = \mathcal{L} \cup \mathcal{V}$. Ensure that $\mathcal{L}$ is sorted.
   End If
   (e) End If
5. End While

Appendix C

Detailed Output from Performance Simulations

A detailed picture of variable selection performance can be obtained by considering the performance as a bivariate outcome. For a true model of size s, when models up to size τ are considered, the structure of the selected model can be described by two variables: n_c, the number of correct variable choices, and n_w, the number of wrong variable choices. Different combinations of n_c and n_w correspond to the different outcome categories $\mathcal{T}, \mathcal{O}, \mathcal{U}, \mathcal{P}, \mathcal{W}$ described in Section 2.1.1.

The tables that follow contain sub-tables giving the empirical distribution of (n_c, n_w) for each method and for each true model size. Interpretation of the sub-tables is described below in Figure C.1. The figure shows the structure of the sub-table for a case where the true model has size four.

Table C.1: Distribution of the number of correct and incorrect variables selected in the PB_{12} performance simulation, for each method and each truth size. Values are in percent. The upper-right corner of each sub-table gives the percent of times the true model was chosen.

True Size = 1

Oracle

Incorrect \ Correct	0	1
0	1	99
1		
2		
3		
4		

Stepwise

	0	1
0		23
1		16
2		12
3	1	47
4	1	

SAMS

	0	1
0		57
1		24
2		14
3		4
4		

AICc

	0	1
0		
1		1
2		16
3		82
4	1	

True Size = 2

Oracle

	0	1	2
0	1	12	87
1			
2			
3			
4			

Stepwise

	0	1	2
0		2	12
1	1	1	13
2	3	1	55
3	2	3	
4	6		

SAMS

	0	1	2
0		7	69
1		2	16
2		1	2
3	1		
4	1		

AICc

	0	1	2
0			2
1			21
2		1	65
3		8	
4	3		

True Size = 3

Oracle

	0	1	2	3
0	5	14	29	52
1				
2				
3				
4				

Stepwise

	0	1	2	3
0	3	3	1	5
1	5	3	2	26
2	10	2	8	
3	7	10		
4	15			

SAMS

	0	1	2	3
0		6	18	42
1		5	5	4
2	3	5	1	
3	5	2		
4	3			

AICc

	0	1	2	3
0				17
1			3	35
2		2	13	
3	3	15		
4	13			

True Size = 4

Oracle

	0	1	2	3	4
0	18	20	22	23	17
1					
2					
3					
4					

Stepwise

	0	1	2	3	4
0	4	7	3	1	3
1	7	5	2	4	
2	11	4	8		
3	9	12			
4	21				

SAMS

	0	1	2	3	4
0		9	14	10	5
1	1	9	6	1	
2	8	12	2		
3	10	8			
4	5				

AICc

	0	1	2	3	4
0				2	11
1			3	8	
2		5	16		
3	5	27			
4	23				

Table C.2: Distribution of the number of correct and incorrect variables selected, PB_{20} performance simulation. Values are in percent.

True Size = 1

Oracle

Incorrect \ Correct	0	1
0		100
1		
2		
3		
4		
5		
6		

Stepwise

	0	1
0		6
1		8
2		5
3		5
4		3
5		71
6	2	

SAMS

	0	1
0		53
1		29
2		14
3		3
4		
5		
6		

AICc

	0	1
0		
1		
2		
3		
4		2
5		96
6	1	

True Size = 2

	0	1	2
0		5	95
1			
2			
3			
4			
5			
6			

	0	1	2
0			3
1			5
2			4
3			4
4			73
5		6	
6	3		

	0	1	2
0		5	75
1		2	13
2		1	3
3			
4			
5			
6			

	0	1	2
0			
1			
2			
3			3
4			87
5		8	
6	1		

True Size = 3

	0	1	2	3
0		2	17	81
1				
2				
3				
4				
5				
6				

	0	1	2	3
0				1
1				2
2				2
3	1		1	52
4	1		14	
5	2	11		
6	10			

	0	1	2	3
0		3	22	58
1		3	5	4
2	1	1	1	
3	1			
4				
5				
6				

	0	1	2	3
0				
1				
2				1
3			1	69
4			20	
5		7		
6	2			

True Size = 4

	0	1	2	3	4
0		2	10	33	55
1					
2					
3					
4					
5					
6					

	0	1	2	3	4
0					1
1				1	1
2	1	1	1		21
3	1	1		14	
4	1	2	15		
5	2	19			
6	17				

	0	1	2	3	4
0		5	21	26	21
1		4	6	3	
2	3	4	2		
3	3	1			
4	1				
5					
6					

	0	1	2	3	4
0					
1					
2					44
3			1	22	
4			14		
5		11			
6	6				

True Size = 5

	0	1	2	3	4	5
0	1	5	11	22	30	31
1						
2						
3						
4						
5						
6						

	0	1	2	3	4	5
0						
1	1	1	1			4
2	1	1	1		4	
3	1	2		10		
4	1	1	17			
5	3	26				
6	22					

	0	1	2	3	4	5
0		8	20	17	8	1
1	2	9	8	3	1	
2	6	7	1			
3	5	3				
4	2					
5						
6						

	0	1	2	3	4	5
0						1
1						13
2					14	
3				18		
4		1	19			
5	1	22				
6	11					

True Size = 6

	0	1	2	3	4	5	6
0	3	7	13	20	23	21	13
1							
2							
3							
4							
5							
6							

	0	1	2	3	4	5	6
0					1		
1	1	1	1				1
2	1	1	1		4		
3	3	2	1	10			
4	2	1	20				
5	2	23					
6	23						

	0	1	2	3	4	5	6
0		8	23	11	3		
1	1	12	10	1			
2	7	8	4				
3	7	3					
4	2						
5							
6							

	0	1	2	3	4	5	6
0							3
1						5	
2					10		
3			1	18			
4		2	25				
5	1	21					
6	14						

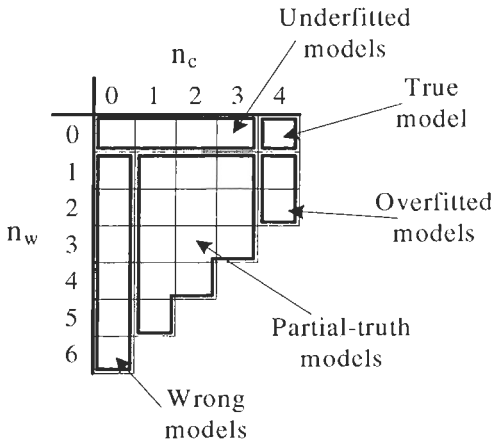

Figure C.1: Schematic of the distribution of n_c and n_w for chosen models.

Bibliography

Box, G. E. P., W. G. Hunter, and J. S. Hunter (1978). *Statistics for Experimenters*. John Wiley and Sons.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics 37*(4), 373–84.

Burnham, K. P. and D. R. Anderson (2002). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer.

Chipman, H. (1998). Fast model search for designed experiments with complex aliasing. In B. Abraham (Ed.), *Quality Improvement Through Statistical Methods*. Birkhauser.

Chipman, H., M. Hamada, and C. F. J. Wu (1997). A bayesian variable-selection approach for analyzing designed experiments with complex aliasing. *Technometrics 39*(4), 372–381.

Clausen, J. (1999). Branch and bound algorithms: Principles and examples. Unpublished manuscript.

Dowsland, K. A. (1993). Some experiments with simulated annealing techniques for packing problems. *European Journal of Operational Research 68*, 389–399.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association 96*(456), 1348–1360.

George, E. I. and R. E. McCulloch (1993). Variable selection via gibbs sampling. *Journal of the American Statistical Association 88*(423), 881–889.

Hamada, M. and N. Balakrishnan (1998). Analyzing unreplicated factorial experiments: a review with some new proposals. *Statistica Sinica 8*, 1–41.

Hamming, R. W. (1986). *Coding and Information Theory* (2 ed.). Prentice Hall.

Harrell, Jr., F. E. (2001). *Regression Modeling Strategies*. Springer.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Hurvich, C. M. and C.-L. Tsai (1989). Regression and time series model selection in small samples. *Biometrika 76*(2), 297-307.

Jessop, A. (1994). *Informed Assessment: An Introduction to Information, Entropy, and Statistics*. Prentice Hall.

Johnson, R. A. and D. W. Wichern (2002). *Applied Multivariate Statistical Analysis* (5 ed.). Prentice Hall.

Lenth, R. V. (1989). Quick and easy analysis of unreplicated factorials. *Technometrics 37*(4), 469–473.

Loughin, T. M. and W. Noble (1997). A permutation test for effects in an unreplicated factorial design. *Technometrics 39*(2), 180–190.

Miller, A. (2002). *Subset Selection in Regression* (2 ed.). Number 95 in Monographs on Statistics and Probability. Chapman & Hall/CRC.

Miller, A. and R. R. Sitter (2001). Using the folded-over 12-run plackett-burman design to consider interactions. *Technometrics 43*(1), 44–55.

Montgomery, D. C., E. A. Peck, and G. G. Vining (2001). *Introduction to Linear Regression Analysis* (3 ed.). John Wiley and Sons.

Reeves, C. R. (Ed.) (1993). *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific.

Rencher, A. C. (2000). *Linear Models in Statistics*. John Wiley and Sons.

Royall, R. (1997). *Statistical Evidence: A Likelihood Paradigm*. Number 71 in Monographs on Statistics and Probability. Chapman & Hall/CRC.

Taper, M. L. and S. R. Lele (Eds.) (2004). *The Nature of Scientific Evidence: Statistical, Philosophical, and Empirical Considerations*. University of Chicago Press.

Wu, C. F. J. and M. Hamada (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization*. John Wiley and Sons.

Zhu, M. and H. A. Chipman (2006). Darwinian evolution in parallel universes: A parellel genetic algorithm for variable selection. *Technometrics 48*(4), 491- 502.