THE GROUPING PROBLEM IN DISTRIBUTION-FREE GENERAL LINEAR REGRESSION

by

Marthinus J. Pella, Drs.

in the Department of Mathematics

submitted in fulfilment
of the requirements for the degree of
Master of Science.

University of Tasmania
Hobart
November, 1990

Except as stated herein, this thesis contains no material which has been accepted for the award of any other degree or diploma in any university, and to the best of my knowledge and belief, contains no copy or paraphrase of material previously published or written by another person except where duly acknowledged.

Marthinus J. Pella

ABSTRACT

An exact distribution–free method is proposed for solving general linear regression problems, which have identically distributed errors and one of the slope parameters of interest. The method reduces the model to simple linear regression form through grouping of observations, and then uses an exact distribution–free method for slope in simple linear regression to test or estimate the parameter of interest. Of course reducing the model involves a loss of efficiency. The choice of an optimal grouping to minimize efficiency loss is discussed.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. B. M. Brown for supervising this work. His ideas, experience and guidance were invaluable in the preparation of this thesis.

My thanks also go to Mr. Tim Stokes for giving his time in assisting with proof reading and English corrections, and to Mr Glen McPherson, Mrs. Betty Golding and Mr. Kym Hill for their help in the typing of the manuscript.

I am grateful to the Australian International Development Assistance Bureau for giving an award to allow me to study as a research student at the University of Tasmania.

Finally, I also wish to thanks my wife Wieke and my daughters Stephanie and Anita for their patience and understanding during the time I have spent in this research.

CONTENTS

CHAPTER 1

INTRODUCTION

Statistical inferences are based only in part upon the observations. An equally important base is formed by prior assumptions about the underlying situation. Even in the simplest cases, there are explicit or implicit assumptions about randomness and independence, about distributional models, perhaps prior distributions for some unknown parameters and so on. Thus we can say briefly that each statistical method is based on special assumptions about the population from which the sample was obtained.

The usual method of solving general linear regression problems is the least squares (LS) method. This method has the nice property of providing *best linear unbiased* estimates for the unknown parameters; however this method is vulnerable to gross errors in the data and is also inefficient for distributions with heavy tails (e.g., Cauchy–type distribution functions). In such cases we need alternative methods which rely on some broader and weaker assumptions about underlying distributional forms such as symmetry or identical error distributions; namely *distribution–free methods.*

In simple linear regression (SLR), numerous distribution–free (DF) tests and the corresponding estimates can be developed. The SLR model is $y_i = \alpha + \beta x_i + \epsilon_i$, $i = 1, 2, \ldots, n$ with $\{\epsilon_i\}$ being random errors. Mood and Brown (1950) have proposed a DF test, based on their median estimates. Parameters α and β can be estimated simultaneously from the two equations, median $(y_i - \hat{\alpha} - \hat{\beta}x_i) = 0$ for $x_i \leq x_M$, and median $(y_i - \hat{\alpha} - \hat{\beta}x_i) = 0$ for $x_i > x_M$, where x_M is the median of $x_1, x_2, \ldots, x_n$. The point estimate $(\hat{\alpha}, \hat{\beta})$ is obtained by trial and error. Theil (1950) developed a simple point estimator

of slope β, the median of $\binom{n}{2}$ slopes $(y_j - y_i)/(x_j - x_i)$, $1 \leq i < j \leq n$, with assumptions that the errors are independent, identically distributed and all x_i are distinct. He also obtained corresponding confidence intervals for β . Adichie (1967) considered a class of rank score tests for the hypothesis $\alpha = \beta = 0$, with the basic assumption that $F(y) = F(y - \hat{\alpha} - \hat{\beta}x)$ is an absolutely continuous, symmetric distribution with square integrable density function. Moreover his point estimators of β required trial and error solutions and also Adichie gave no confidence interval for β. Sen's (1968) estimate is quite analogous to Theil's (1950) but is based on weaker assumptions and does not require all of the $x_1, x_2, \ldots, x_n$ to be distinct. If N is the number of non zero differences $x_j - x_i$, $(1 \leq i < j \leq n)$, the proposed point estimator is the median of N slopes $(y_j - y_i)/(x_j - x_i)$ for which $x_i \neq x_j$. The confidence interval for β is also obtained in terms of two order statistics of this set of N slopes. Brown and Maritz (1982) made a modification to the LS estimating equations in SLR, leading to exact DF inference about slope. Exact inference for intercept is developed by Maritz (1979), based on work of Theil (1950).

The planar regression model is $y_i = \mu + \alpha x_i + \beta z_i + \epsilon_i$, $i = 1, 2, \ldots n$, where $\{\epsilon_i\}$ are random errors, $\{x_i, z_i\}$ are known and α, β are unknown parameters. Suppose β is of interest, μ and α are nuisance parameters and $\{\epsilon_i\}$ are identically distributed. Brown and Maritz (1982) showed how a suitable $\{x_i, z_i\}$ design, coupled with a restricted permutation or restricted randomization scheme, enables exact procedures to be developed. Brown (1985) extended the Maritz/Theil ideas to general regression schemes, through grouping of observations to eliminate the nuisance parameters. By pairing observations , taking differences to eliminate μ and giving symmetric errors, then dividing through by the α coefficient, the model is reduced to SLR with

symmetric errors, and β the slope parameter. The Maritz/Theil scheme then is applicable, but since that involves *further* pairing, the overall problem is one of finding groups of four observations, which through two pairing operations yield one observation distributed symmetrically about β. Exact DF methods for the symmetric location parameter problem are then used.

This thesis outlines another approach to exact DF regression methods in the presence of nuisance parameters through grouping of observations to eliminate the nuisance parameters. The number of groups depends on the number of observations and also on the number of independent design variables. For instance in planar regression, the observations are grouped into k groups, where $k = [n^{1/2}]$, the integer part of $n^{1/2}$; in regression with three independent design variables $k = [(n/2)^{1/2}]$. After grouping and eliminating the nuisance parameters, the model is reduced to simple linear regression form, allowing exact DF methods for slope to be employed.

Of course grouping and reducing the model as described seems to involve a loss of efficiency. A question of interest is the extent of efficiency loss suffered through grouping and reducing the model. How can the groups be chosen to minimize the loss of efficiency? This optimal grouping task is a very difficult combinatorial optimization problem, without convexity or other regular structure leading to efficient unique solution methods. Three methods will be discussed for finding approximate solutions. The methods are : a *Monte Carlo* method which is suitable for small or medium–size designs , a *search for better neighbours* method which is easy to program and implement but can get stuck in local optima, and a general technique known as *simulated annealing* which

has proved to be very successful in diverse Operations Research applications over recent years.

To illustrate how the proposed method in this thesis works, and to demonstrate that the necessary computer programming is relatively straightforward, numerical examples will be given.

To summarize what we have discussed, the *aim* of this thesis is to show how the general linear model (GLM)

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_p x_{pi} + \epsilon_i \ , \ i = 1, 2, \ldots , n \qquad (1.1)$$

where β_j, $j = 0, 1, \ldots , p$ are unknown parameters, $x_{1i}, x_{2i}, \ldots , x_{pi}$ are design constants, $\{\epsilon_i\}$ are independent errors and identically distributed, with *one* of the β_j $(j \neq 0)$ of interest, can be reduced to the simple linear regression model, through grouping of observations to eliminate the nuisance parameters, allowing exact distribution–free methods for slope in simple linear regression to be employed.

The content of the thesis is as follows : Chapter 1 gives the background, problem and aim of the thesis. Chapter 2 contains several concepts about estimation and a brief resume of two documented regression methods, which will be used in the following chapters. The methods are least–squares methods and an exact distribution– free method for slope in simple linear regression (Brown and Maritz, 1982). The least squares method will be involved in the topic of efficiency–loss, and exact distribution–free tests for slope in simple

linear regression will be used for solving the general linear regression problem after reducing the model to simple linear form. In Chapter 3 it is shown how to solve the planar regression problems by using the proposed method. This chapter outlines the method of parameter elimination, a method of calculating the efficiency loss as a consequence of grouping of observations and reducing the model, methods of minimizing efficiency–loss, the basic steps of the computer programs of the approximation method, and a numerical example. In Chapter 4, we extend the ideas presented in Chapter 3 to the case of more than two design independent variables. The modifications are needed just in the section concerning the method of parameter elimination. A numerical example also is given in the last section of this chapter. A discussion about the benefits and limitations of the proposed method will be presented in Chapter 5. A complete computer program of the three approximation methods for solving the planar regression problems will be put in an Appendix.

CHAPTER 2

SOME BASIC CONCEPTS AND DOCUMENTED REGRESSION METHODS

This chapter is presented as a basis for the following chapters, so it contains some concepts and documented regression methods which will be used to develop the proposed method. As stated in Chapter 1, the thesis is concerned with efficiency loss due to grouping and reducing the model, so a concept of efficiency is needed. Here Pitman's asymptotic relative efficiency (ARE) will be used and may be obtained by considering the ratio of efficiencies of least squares analyses for grouped and ungrouped cases, so Pitman's ARE and a short summary of the least squares method in the general linear model (GLM) will be discussed. One of the methods for finding the best grouping to minimize the loss of efficiency given in Chapter 3 will use pooling of estimate and variance, so a method of pooling estimates and variance also will be outlined briefly. The last section will outline an exact DF method for slope in SLR (Brown and Maritz, 1982) which will be used after reducing the GLM form to SLR form.

The contents of this chapter are as follows : pooling estimates, Pitman's asymptotic relative efficiency, the least squares method and an exact DF method for slope in SLR.

2.1. Pooled Estimates.

Let $\hat{\theta}_1$ and $\hat{\theta}_2$ be two independent estimates of an unknown parameter θ. Assume that $\hat{\theta}_1$ and $\hat{\theta}_2$ are unbiased, so for i = 1,2, we have, for minimum variance

$$E(\hat{\theta}_i) = \theta,$$

and

$$\text{var}(\hat{\theta}_i) = \sigma_i^2.$$

The pooled estimate $\hat{\theta}$ of estimators $\hat{\theta}_1$ and $\hat{\theta}_2$ is

$$\hat{\theta} = \frac{\hat{\theta}_1/\sigma_1^2 + \hat{\theta}_2/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2}, \tag{2.1}$$

and the pooled variance of $\text{var}(\hat{\theta}_1)$ and $\text{var}(\hat{\theta}_2)$ is

$$\text{var}(\hat{\theta}) = \left[\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right]^{-1}. \tag{2.2}$$

2.2 Pitman's Asymptotic Relative Efficiency.

When two or more statistics are available for testing a given hypothesis, one statistic is considered more efficient if it is more powerful than other statistics, using the same level of significance, at the same fixed alternative. Such a comparison of powers for two statistics based on the same data is usually dependent on the level of significance α, the sample size n (or some measure of sample sizes with several samples), and the fixed alternative at which the powers are compared. In order to define a suitable measure of efficiency, an alternative approach is adopted comparing the corresponding sample sizes necessary to attain an equal power, say β, at the same alternative for two tests using the same level α. A limit argument is usually needed for this measure to be independent of particular values

α, n; furthermore one needs to use then a sequence of alternatives converging to the null hypothesis at a suitable rate in order to come up with a meaningful definition.

Pitman (1979) defines asymptotic relative efficiency as follows :

Let $\hat{\theta}_1$ and $\hat{\theta}_2$ be two unbiased estimators of an unknown parameter θ, and n the sample size. For $n \longrightarrow \infty$, the efficiency of $\hat{\theta}_1$ relative to $\hat{\theta}_2$ is

$$e = \lim_{n \longrightarrow \infty} \frac{\mathrm{var}(\hat{\theta}_1)}{\mathrm{var}(\hat{\theta}_2)} \tag{2.3}$$

where $\mathrm{var}(\hat{\theta}_1)$ and $\mathrm{var}(\hat{\theta}_2)$ are the variance of $\hat{\theta}_1$ and $\hat{\theta}_2$ respectively.

2.3. The Least Squares Method.

This section will give a short summary of the least squares method in the GLM. If the GLM equation (1.1) is written in a matrix notation, we have

$$\underline{y} = X\underline{\beta} + \underline{\epsilon} \tag{2.4}$$

where $\underline{y}^T = (y_1, y_2, \ldots, y_n)$, $\underline{\beta}^T = (\beta_0, \beta_1, \ldots, \beta_p)$, $\underline{\epsilon}^T = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$ and

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1p} \\ 1 & x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n2} & x_{n3} & \ldots & x_{np} \end{bmatrix}.$$

The least squares assumptions for the error terms are

(i) $\{\epsilon_i\}$ are random variables with mean zero and variance σ^2 (unknown) , that is $E(\epsilon_i) = 0$, $\text{var}(\epsilon_i) = \sigma^2$.

(ii) $\{\epsilon_i\}$ are uncorrelated, that is if $i \neq j$, $\text{cov}(\epsilon_i,\epsilon_j) = 0$.

(iii) $\{\epsilon_i\}$ are normally distributed random variables, that is $\epsilon_i \sim N(0,\sigma^2)$.

The problem is solved by minimizing the sum of squares

$$\begin{aligned} S &= (\underline{y} - X\underline{\beta})^T(\underline{y} - X\underline{\beta}) \\ &= \underline{y}^T\underline{y} - 2\underline{\beta}^T X^T\underline{y} + \underline{\beta}^T X^T X\underline{\beta} \end{aligned}$$

by differentiating S with respect to β , equating to zero, so obtaining the normal equations

$$X^T X\underline{\beta} = X^T\underline{y} \tag{2.5}$$

We assume that X has full rank (p+1) , so $\hat{\underline{\beta}}$ (the LS estimate of $\underline{\beta}$) is

$$\hat{\underline{\beta}} = (X^T X)^{-1} X^T\underline{y} . \tag{2.6}$$

From the least squares assumption $E(\epsilon_i) = 0$, it can be shown that $\hat{\underline{\beta}}$ is an unbiased estimate of $\underline{\beta}$, and by assumptions $\text{cov}(\epsilon_i, \epsilon_j) = \delta_{ij}\sigma^2$, where δ_{ij} is 0 or 1 according to whether $i \neq j$ or $i = j$, we get

$$\text{Cov}(\hat{\underline{\beta}}) = \sigma^2(X^T X)^{-1}, \tag{2.7}$$

i.e. the covariance matrix of the elements of $\hat{\beta}$, so that $\text{var}(\hat{\beta}_{i-1}) = \sigma^2 a_{ii}$, $i = 1, 2, \ldots, (p+1)$, where $\{a_{ii}\}$ are diagonal elements of the matrix $(X^T X)^{-1}$.

2.4 An Exact Distribution–Free Methods for Slope in Simple Linear Regression.

Brown and Maritz (1982) made a modification to the least squares estimating equations, leading to exact distribution–free inference for slope. Instead of the least squares assumptions (i), (ii), and (iii) in Section 2.3., their method relies merely on two broader and weaker assumptions about underlying distributional forms, i.e. independent and identically distributed errors. These assumptions enable the basic permutation argument to be applied to obtain an exact permutation procedure for the slope parameter.

The model used is $y_i = \alpha + \beta x_i + \epsilon_i$, $i = 1, 2, \ldots, n$ where α, β are unknown constants, $\{x_i\}$ are design constants, $\{\epsilon_i\}$ are independent errors, identically distributed. The least squares estimating equations

$$\sum_{i=1}^{n} r_i = 0$$

and

$$\sum_{i=1}^{n} x_i r_i = 0$$

where $r_i = y - \alpha - \beta x$, are modified to the general form

$$\sum_{i=1}^{n} \psi(r_i) = 0$$

and (2.8)

$$\sum_{i=1}^{n} h(x_i)\psi(r_i) = 0$$

Both h and ψ should preserve the original orderings of $\{x_i\}$ and $\{r_i\}$. In addition, ψ should be suitably centred so that the first equation of (2.8) provides consistent estimates of α when β is known. Though the approach is general, the various exact tests and confidence intervals are worked out in detail only for three specific cases of the general residual transformation ψ; the cases considered are ψ equal to sign, rank and $\psi(x) = x$ itself. Modified designs of h are used throughout and also for h = sign, rank and identity.

Of course, as a consequence of the transformation described, an efficiency loss will be involved. The asymptotic efficiency of the estimate of slope β relative to least squares is

$$\rho_h^2 e_\psi \qquad (2.9)$$

where ρ_h is the limit correlation coefficient between $\{h(x_i)\}$ and $\{x_i\}$, and e_ψ is an efficiency factor associated with ψ.

A different choice of h and ψ will give different efficiency loss results. For each of the choices of ψ given, the associated procedures are less than optimal under at least one criterion; the choice sign suffers loss of efficiency, the choice rank is difficult to compute, and the choice of identity is not robust (the least squares choice).

This thesis is concerned with loss of efficiency due to grouping which will be additional to, and independent of, whatever choices of h and ψ are used when the model is reduced to SLR form. So studying grouping efficiency loss will not need to refer to any particular choices of h and ψ (see the next chapter). However, to illustrate how all the exact DF methods are used, with examples, some specific

choices of h and ψ will be needed. For illustrative purposes the choices $h(x) = x$ and $\psi(r) = \text{rank}(r)$ will be used and will be outlined in this section.

Inference about only slope β is based on the estimating equations (2.8), for $h(x) = x$, $\psi(r_i) = \text{rank}(r_i) - (n+1)/2$, and because $\text{rank}(r_i)$ is independent of α

$$S(\beta) = \sum_{i=1}^{n} x_i\{\text{rank}(y_i - \beta x_i) - (n+1)/2\}. \tag{2.10}$$

S is a monotone function of β which decreases only in downward jumps at certain β values, i.e. at $\beta = \beta_{ij} = (y_i - y_j)/(x_i - x_j)$ for all pairs i, j.

We now discuss the usual statistical inference problems, i.e. the problem of point estimation, confidence intervals and hypothesis testing.

Suppose β_0 is the true value of the slope β. The estimated value of β_0 is the weighted median of β_{ij} with weights $|x_i - x_j|$ for all pairs i, j, so

$$\hat{\beta}_0 = \begin{matrix}\text{Weighted median} \\ \text{Weights } |x_i - x_j|\end{matrix} \left[\frac{y_i - y_j}{x_i - x_j} \right] \tag{2.11}$$

where $\hat{\beta}_0$ is the estimate value of β_0.

Tests $H_0 : \beta = \beta_0$ are rejected for large or small values of $S(\beta_0)$. Because of the assumption of identical error distributions, the exact null distribution of $T = S(\beta_0) + n(n+1)\bar{x}/2$ is enumerated by calculating all the values $\sum_{i=1}^{n} x_i \rho_i$, where $\rho_1, \rho_2, \ldots \rho_n$ is a permutation of 1, 2, ... , n. All n! such values are equi–probable, and the number of these permutations obviously becomes excessive as n increases.

In most cases it is convenient to use the normal approximation. According to Wald and Wolfowitz (1944), if the sequences $(x_1, x_2, \ldots, x_n)$ and $(\rho_1, \rho_2, \ldots, \rho_n)$ satisfy condition W, that is for all integral $r > 2$

$$\frac{n^{-1}\sum_{i=1}^{n}(x_i - \bar{x})^r}{\left[n^{-1}\sum_{i=1}^{n}(x_i - \bar{x})^2\right]^{r/2}} = O(1) \qquad (2.12)$$

and

$$\frac{n^{-1}\sum_{i=1}^{n}(\rho_i - \bar{\rho})^r}{\left[n^{-1}\sum_{i=1}^{n}(\rho_i - \bar{\rho})^2\right]^{r/2}} = O(1) \qquad (2.13)$$

then the distribution of

$$\tau = \frac{T - E(T)}{Var(T)}$$

approaches the normal distribution with mean 0 and variance 1 as $n \longrightarrow \infty$, where $E(T) = n\,\bar{x}\,\bar{\rho}$, $Var(T) = (S_{xx}S_{\rho\rho})/(n-1)$, $\bar{x} = (\sum_{i=1}^{n} x_i)/n$, $\bar{\rho} = (\sum_{i=1}^{n} \rho_i)/n$, $S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2$ and $S_{\rho\rho} = \sum_{i=1}^{n}(\rho_i - \bar{\rho})^2$, and therefore the approximate distribution of T is

$$T = \sum_{i=1}^{n} x_i\rho_i \sim N\left(n\,\bar{x}\,\bar{\rho},\ \frac{S_{xx}S_{\rho\rho}}{(n-1)}\right) \qquad (2.14)$$

Because $(\rho_1, \rho_2, \ldots, \rho_n)$ is a permutation of $(1, 2, \ldots, n)$, the condition (2.13) is satisfied, so whether (2.14) is satisfied or not merely depends on the condition (2.12).

If the rank scores are chosen centred, that is $\sum_{i=1}^{n} \rho_i = 0$, or $\rho_i = i - (n+1)/2$, we have

$$T = \sum_{i=1}^{n} x_i \rho_i \sim N(0, \frac{S_{xx} S_{\rho\rho}}{(n-1)})$$

and

$$S_{\rho\rho} = \sum_{i=1}^{n} (\rho_i - \bar{\rho})^2 = \frac{n(n^2-1)}{12}.$$

Thus the null distribution of T is

$$N(0, \frac{n(n+1)}{12} S_{xx}). \qquad (2.15)$$

For confidence intervals, the behaviour of S has to be examined. From (2.10) we obtain

$$S(-\infty) = \sum_{i=1}^{n} x_i \{ \text{rank}(x_i) - (n+1)/2 \} \qquad (2.16)$$

and

$$S(\infty) = \sum_{i=1}^{n} x_i \{ \text{rank}(-x_i) - (n+1)/2 \}.$$

Thus

$$S(\infty) = \sum_{i=1}^{n} x_i \{ (n+1) - \text{rank}(x_i) - (n+1)/2 \}$$

$$= -\sum_{i=1}^{n} x_i \{ \text{rank}(x_i) - (n+1)/2 \} \qquad (2.17)$$

S decreases only in downwards jumps of size $|x_i - x_j|$ at $\beta = \beta_{ij}$ for all pairs i, j. From (2.16) and (2.17) we see that $S(+\infty) = -S(-\infty)$, so the structure of S may therefore be enumerated systematically; the confidence intervals for the slope β must have some β_{ij} as their end points, and the approximate confidence level of any such interval may be found.

CHAPTER 3

PLANAR REGRESSION

This chapter outlines the proposed method of solving the planar regression problems when only one of the slope parameters is of interest. The basic steps are as follows. Firstly, eliminate the other slope parameter through grouping of observations such that the planar regression can be reduced to SLR form; the group is chosen to approximately minimize the efficiency loss. Secondly, estimate or test the slope of interest by using an exact distribution–free method for slope in SLR.

This chapter is concerned with the first step and outlines the method of parameter elimination, the method of calculating the efficiency loss, the method of minimizing efficiency loss, the computer programs and a numerical example to illustrate how the method works.

3.1. Method of Parameter Elimination

In planar regression, the usual model for fitting a straight line to data is :

$$y_i = \mu + \alpha x_i + \beta z_i + \epsilon_i, \quad i = 1, 2, \ldots, n \qquad (3.1)$$

where n is the number of observations, $\{x_i, z_i\}$ are design constants, $\{\epsilon_i\}$ are independent errors, identically distributed with finite variance, and μ, α, β are unknown parameters. Suppose β is of interest, and that μ and α are nuisance parameters.

There are many possibilities for grouping of observations to eliminate the parameter α and reduce the equation (3.1) to SLR form, but this section outlines just one simple method of grouping. Other methods of grouping will be discussed later (see Chapter 5). The method is said to be simple because it is straightforward compared with other methods; moreover after reducing the model to simple linear regression form, the slope parameter of interest can be estimated or tested directly by employing an exact DF method for slope in simple linear regression.

To eliminate the parameter α, the observations are placed into k groups where

$$k = [n^{1/2}].$$

the integer part of $n^{1/2}$. Let $\lambda_1, \lambda_2, \ldots, \lambda_k$ be constants, and for $i = 1, 2, \ldots, k$, define

$$y_i^* = \sum_{j=1}^{k} \lambda_j y_{j+k(i-1)} = \mu \sum_{j=1}^{k} \lambda_j + \alpha \sum_{j=1}^{k} \lambda_j x_{j+k(i-1)} + \beta \sum_{j=1}^{k} \lambda_j z_{j+k(i-1)} + \sum_{j=1}^{k} \lambda_j \epsilon_{j+k(i-1)} . \qquad (3.2)$$

Suppose that $\{\lambda_j\}$ are chosen so that for all $i = 1, 2, \ldots, k$,

$$\sum_{j=1}^{k} \lambda_j x_{j+k(i-1)} = \text{constant}, \quad x^* \text{ say.} \qquad (3.3)$$

Then from (3.2) and (3.3)

$$y_i^* = \mu^* + \beta z_i^* + \epsilon_i^* \qquad (3.4)$$

where $y_i^* = \sum_{j=1}^{k} \lambda_j y_{j+k(i-1)}$, $\mu^* = \mu c + \alpha x^*$, $c = \sum_{j=1}^{k} \lambda_j$, $z_i^* = \sum_{j=1}^{k} \lambda_j z_{j+k(i-1)}$ and $\epsilon_i^* = \sum_{j=1}^{k} \lambda_j \epsilon_{j+k(i-1)}$. The equation (3.4) is of SLR form because $\{\epsilon_i^*\}$ are independent, identically distributed, and so can be treated by exact DF methods.

How can $\{\lambda_j\}$ be chosen to assume that (3.3) holds? Let $\underline{\lambda}^T = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ and a $k \times k$ matrix X be such that $(X)_{i,j} = x_{j+k(i-1)}$. We need to find $\underline{\lambda}$ to satisfy (3.3) , i.e so that

$$X \underline{\lambda} = x^* \underline{1}$$

where $\underline{1}^T = (1, 1, \ldots, 1)$. Since the value of x^* is immaterial, the choice

$$\underline{\lambda} = X^{-1} \underline{1} \qquad (3.5)$$

will always suffice. That is, the row–sums of X^{-1} provide the multipliers $\{\lambda_j\}$.

The k pairs of values $\{z_i^*, y_i^*\}$ for the SLR model (3.4) can be calculated more easily using a matrix notation. Let $k \times k$ matrices Z and Y be such that $(Z)_{i,j} = z_{j+k(i-1)}$ and $(Y)_{i,j} = y_{j+k(i-1)}$. If $\underline{z}^{*T} = (z_1^*, z_2^*, \ldots, z_k^*)$,

---

Of course, X needs to be non–singular, and only non–singular X should be accepted in the random search methods soon to be described. See also Brown, B.M. and Pella, M.J. (1991), The grouping problem in distribution–free planar regression, Austral. J. Statist. 33, to appear.

and $\underline{y}^{*T} = (y_1^*, y_2^*, \ldots, y_k^*)$ then

$$\begin{aligned} \underline{z}^* &= Z\,\underline{\lambda} \\ \underline{y}^* &= Y\,\underline{\lambda} \end{aligned} \tag{3.6}$$

We have reduced the planar regression (3.1) with n observations to SLR equations (3.4) with k observations.

3.2. Efficiency Loss due to Grouping.

As stated in Chapter 1, grouping and reducing a model will result in a loss of efficiency. The extent of efficiency loss thereby suffered is a question of natural interest.

By an analysis similar to that in Brown (1985) , it can be shown that the asymptotic efficiency of exact DF methods applied after grouping is $e_G \cdot e_{DF}$, where e_{DF} is the characteristic efficiency of the particular DF method used (for example as in the symmetric location problem or simple linear regression), and where e_G is a factor attributable to grouping. The factor e_G is of interest here, and it may be obtained by considering the ratio of efficiencies of least–squares analyses for grouped and ungrouped cases.

Such a ratio of efficiencies is just a ratio of variances of estimates of β (see Section 2.2). If equation (3.1) is written in a matrix form, we have $\underline{y} = A\,\underline{\theta} + \underline{\epsilon}$, where

$$A = \begin{bmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & z_n \end{bmatrix},$$

$\underline{y}^T = (y_1, y_2, \cdots, y_n)$, $\underline{\theta}^T = (\mu, \alpha, \beta)$ and $\underline{\epsilon}^T = (\epsilon_1, \epsilon_2, \cdots, \epsilon_n)$. Suppose that LS assumptions are valid, and if $\hat{\beta}_{LS}$ is the ungrouped LS estimate of β, then from (2.7)

$$\text{var}(\hat{\beta}_{LS}) = \sigma^2 \{(A^T A)^{-1}\}_{33}, \qquad (3.7)$$

where σ^2 is the observational error variance.

After grouping

$$\underline{y}^* = A^* \underline{\theta}^* + \underline{\epsilon}^*$$

where $\underline{y}^{*T} = (y_1^*, y_2^*, \cdots, y_k^*)$,

$$A^* = \begin{bmatrix} 1 & z_1^* \\ 1 & z_2^* \\ \vdots & \vdots \\ 1 & z_k^* \end{bmatrix},$$

$\underline{\theta}^{*T} = (\mu^*, \beta)$ and $\underline{\epsilon}^{*T} = (\epsilon_1^*, \epsilon_2^*, \cdots, \epsilon_k^*)$. The LS assumptions also hold for the grouped data because $\{\epsilon_i^*\}$ are independent and normally distributed and if

$\hat{\beta}_{GLS}$ is the least squares estimate of β after grouping, then from (2.7)

$$\text{var}(\hat{\beta}_{GLS}) = \sigma^{*2}\{(A^{*T}A^{*})^{-1}\}_{22}, \tag{3.8}$$

where $\sigma^{*2} = \text{var}(\epsilon_1^{*}) = \sigma^2 \sum_{j=1}^{k} \lambda_j^2$. Thus from (2.3), (3.7) and (3.8) we obtain

$$e_G = \frac{\text{var}(\hat{\beta}_{LS})}{\text{var}(\hat{\beta}_{GLS})} = \frac{\{(A^{T}A)^{-1}\}_{33}}{\sum_{j=1}^{k} \lambda_j^2 \{(A^{*T}A^{*})^{-1}\}_{22}} \tag{3.9}$$

The formula (3.9) is the asymtotic efficiency of estimates of slope β for the grouped LS relative to ungrouped LS.

3.3. Minimizing the Loss of Efficiency

To maximize the grouping efficiency e_G , it follows from (3.9) that $\text{Var}(\hat{\beta}_{GLS})$ must be minimized, i.e. groups are to be chosen to minimize

$$\{(A^{*T}A^{*})^{-1}\}_{22} \sum_{j=1}^{k} \lambda_j^2 = \frac{k \sum_{j=1}^{k} \lambda_j^2}{k \sum_{j=1}^{k} z_j^{*2} - (\sum_{j=1}^{k} z_j^{*})^2}. \tag{3.10}$$

This optimal–groupings task is a very difficult combinatorial optimization problem, without convexity or other regular structure leading to

efficient unique solution methods. Three methods for finding approximate solutions will be described.

(i) A Monte Carlo method, estimating the probability of having the optimal solution, suitable for small or medium–sized designs;

(ii) a search method which seeks improved neighbours of any current solution, which is easy to program and implement but which can get stuck in local optima; and

(iii) a general technique known as *simulated annealing*, of great recent popularity in operations research circles.

The two large–sample methods (ii) and (iii) will then be illustrated and compared via an example.

3.3.1. A Monte Carlo Method.

When n is not too large, it can be surprisingly efficient just to generate completely random groupings, evaluate e_G for each, and repeat a large number (say N) times. Although it may appear that there are a large number (n!) of possible groupings, many will share the same value of e_G, as can be observed by noting that e_G is unchanged by swapping rows and/or columns in the matrices X, Z and Y. Therefore if N is large, the probability that the current maximum e_G is in fact the overall maximum can be surprisingly high.

First, generate random groupings for a short time, then count and list the number of distinct values of e_G encountered. Let there be m distinct values. Then randomly generate N further groupings, where N is now very

much larger. Count the number of times a value e_G occurs which was among the initial set of m values.

Let U be the number of such occurences, so that U ~ Bi(N, m/M) , where M is the total possible number of different e_G values. When U is observed, a confidence interval for m/M and hence for M may be evaluated. Possibly the use of approximations Binomial ——> Poisson ——> Normal is the easiest path to take in getting the confidence interval.

To illustrate, for the design with n = 9 points (using 9 observations selected randomly from the data of the numerical example of Section 3.4), an initial set of m = 83 distinct e_G values followed by N = 40,000 further random groupings gave U = u = 459 . The resulting approximate 95 % confidence interval for the Poisson parameter mN/M is (418.89 , 502.96) and the corresponding interval for M is (6601 , 7926).

What then is the probability that among 40,000 values of e_G , the overall maximum is already present? Taking a conservatively large estimate of M as 8,000 , the probability of missing the maximum is $e^{-5} = 0.0067$; the calculations are as for the famous " birthdays paradox " . Thus in this case we can be over 99 % certain of having already found the overall solution.

3.3.2. A search for better neighbours.

Any "better–neighbour" search depends firstly on having a concept of "neighbour". In the present situation, a neighbour of a given grouping will be any other grouping produced by the smallest possible change, i.e. the interchange of a single pair of corresponding elements within X and within Z . Note that within–row or within–column changes create different groupings and different e_G values.

A simple neighbours search is to start with any randomly chosen grouping, and generate a neighbouring grouping by interchanging a randomly chosen pair. Evaluate e_G for the new grouping, and move to the new grouping if the e_G value exceeds that for the original grouping. Keep re–generating random pairs and moving to better neighbours indefinitely, or until the procedure appears to terminate at a local maximum with no better neighbours. Repeat the whole procedure several times to see if improved finishing groupings can be obtained.

Because every grouping has $n(n-1)/2$ neighbours, it can be more efficient to generate s neighbours at random at each step, where s is a fixed integer possibly greater than 1. Choose the neighbour with best value of e_G and move to it if the new e_G exceeds the old. For $s > 1$, this refinement can lead to more rapid improvement.

A variation of the neighbour search is to divide the data into two (or more) separate sets, and carry out the grouping operation separately within each set. Each set yields an independent estimate of β, say $\hat{\beta}_1$ and $\hat{\beta}_2$, with different expressions $\mathrm{var}(\hat{\beta}_1)$ and $\mathrm{var}(\hat{\beta}_2)$. The final combined estimate will use weight proportional to $(\text{variance})^{-1}$ (see Section 2.1), and the final expression for grouped efficiency is

$$e_G = \{(A^TA)^{-1}\}_{33} \sum{}' [\{(A^{*T}A^*)^{-1}\}_{22} \Sigma \lambda_j^2]^{-1} \qquad (3.11)$$

where $\sum'$ refers to a sum over the two (or more) sets.

As in the neighbour search, randomly chosen pairs are swapped, either within or between the sets, and improved combinations of sets/groups are accepted.

All these variations on the neighbour search theme end at local maxima which, while usually having good efficiency, might not be the overall optimum.

3.3.3. Simulated annealing.

The method of simulated annealing stems from Kirkpatrick et. al (1983), who applied to general optimization problems a method of Metropolis et. al (1953), which mimicked the passage to crystalline states of cooling high temperature material. The technique has proved to be very succsessful in diverse Operation Research applications over recent years.

When applied to the optimal–grouping problem, the method is as follows.

(i) Generate a new grouping G_1 and evaluate its grouped efficiency e_{G_1}. The new grouping can be generated in any fashion, either totally at random or by some interchanges of pairs within the existing grouping G_o. However, generating totally random new groupings would be computationally wasteful, and some method based on small changes to G_o is preferable.

(ii) Accept G_1 if $e_{G_1} > e_{G_o}$. Otherwise, accept G_1 with probability

$$p = \exp\{-(e_{G_o} - e_{G_1})/T)\}, \qquad (3.12)$$

where T is a " temperature " parameter which is decreased in some slow manner during the course of iterations.

(iii) Continue the iteration process for as long as is possible.

Note that *worse* G_1 can be accepted, with a relatively high probability at early stages when T is high, but with much lower probability later on. Thus G_1 can escape from local maxima, but it is more difficult to escape from later local maxima which are more likely to be " close " to the overall maxima.

3.3.4. Computer Programs.

This section outlines just *the basic steps* of the computer programs for finding an approximation of the best grouping, whereas the complete program can be seen in Appendix 5.

Because only k^2 observations of the n observations available are used for estimating the parameter of interest, firstly we have to choose the k^2 observations from the n observations randomly. Our aim is to allocate the k^2 observations into k groups such that e_G will be maximum; however, to simplify the flow–charts we just need to find an order of the k^2 observations, because when we allocate the k^2 observations into k groups, the first k observations becomes the members of the first group, the second k observations becomes the members of the second group and so on. So our aim now is to find an ordering of the k^2 observations which minimizes the loss of efficiency.

The Turbo Pascal Language Version 5.0 is used to write the programs. Suppose that the original ordering of the observations is x_i, z_i, y_i, where $i = 1, 2, \ldots, n$. Some *arrays* will be used to store the (order of) data. We now discuss the flow–charts of the three approximation methods one by one. The

complete flow–chart will be very complicated, so to simplify the flow–charts, here we just show how to get the approximation of the best order of data without describing the flow–chart of the *procedures* of choosing the k^2 observations, calculating e_G , and so on. Those procedures can be seen directly in the computer programs.

Monte Carlo Method.

The arrays used to store the data are as follows :

(i). XOrig[i], ZOrig[i], YOrig[i], i = 1, 2, ... , n , are used to store the original ordering of the n observations.

(ii). XSelectBest[i], ZSelectBest[i], YSelectBest[i], i = 1, 2, ... , k^2 are first used to store the k^2 observations selected, and later for keeping an ordering of the data which gives improved efficiency.

(iii). XCalc[i], ZCalc[i], YCalc[i] are used to store an ordering of the data used for calculating e_G .

Some other variables used in the flow–chart are:

(i) BestEfficiency is used to store the best e_G so far.

(ii) NuOfRepeat is the number of repeats of the procedure of calculating e_G to get an approximation of the best efficiency.

The flow–chart is as follows :

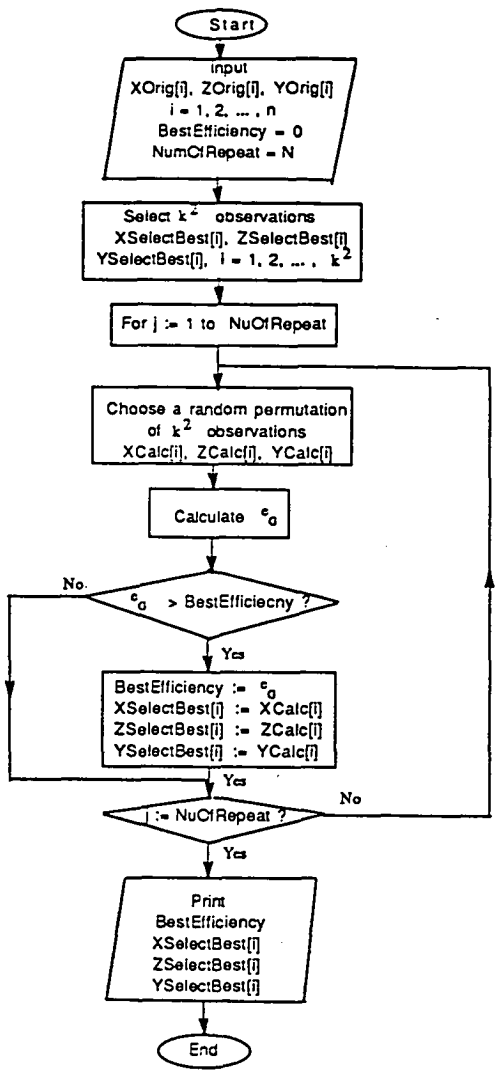

Figure 3.1.
The flow–chart of the Monte Carlo method.

A Search for better neighbour.

The arrays needed to store the data are the same as the Monte Carlo method, except that XSelectBest[i], ZSelectBest[i] and YSelectBest[i] have a slightly different meaning, in that the arrays are first used to store the k^2 observations selected, and then for storing a neighbour which gives improved efficiency. The variables BestEfficiency and NuOfRepeat have the same meaning as for the Monte Carlo Method. The flow–chart is as follows :

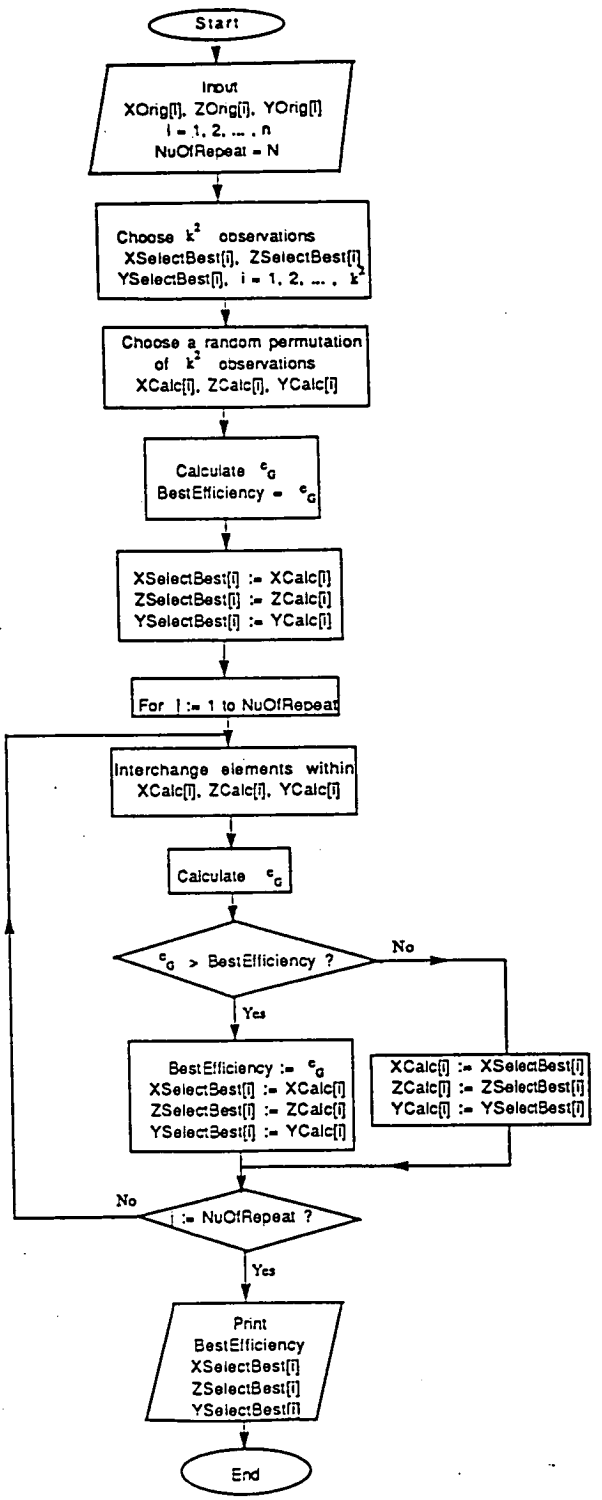

Figure 3.2.
The flow–chart of a search for better neighbour

Simulated Annealing Method.

The arrays needed to store the data are as follows:

(i) (i) XOrig[i], ZOrig[i], YOrig[i], i = 1, 2, ... , n are used to store the original ordering of the data.

(ii) XSelectMove[i], ZSelectMove[i], YSelectMove[i], i = 1, 1, ... , k^2 are first used to store the k^2 observations selected, and then for storing an ordering of the k^2 observations if it gives improved efficiency, or worse efficiency with probability as presented in (3.12).

(iii) XCalc[i], ZCalc[i], YCalc[i] are first used to store a random permutation of data and then to store an ordering of the data used for calculating e_G .

(iv). XSelectBest[i], ZSelectBest[i], YSelectBest[i] are used to store the best ordering of the observations so far.

Some other variables used are :

(i) Temp is used for temperature.

(ii) TFactor is the temperature multiplying factor.

(iii) NuOfTempUsed is the number of temperatures used in the process of getting the best efficiency. The temperature will decrease with the factor TFactor.

(iv) NuOfCalcEachTemp is the number of repeats of the procedure of calculating e_G at each temperature.

(v) ProbMove is the probability of moving from one ordering of the data to others.

$$\text{ProbMove} < \begin{cases} = 1 & \text{if } e_G \geq \text{AnnealEff} \\ = \exp\{-(\text{AnnealEff} - e_G)/\text{Temp}\} & \text{if } e_G \leq \text{AnnealEff} \end{cases}.$$

(vi) AnnealEff is used to record a new e_G each time we move to a new ordering of the data.

(vii) BestEfficiency is used for recording the best efficiency so far.

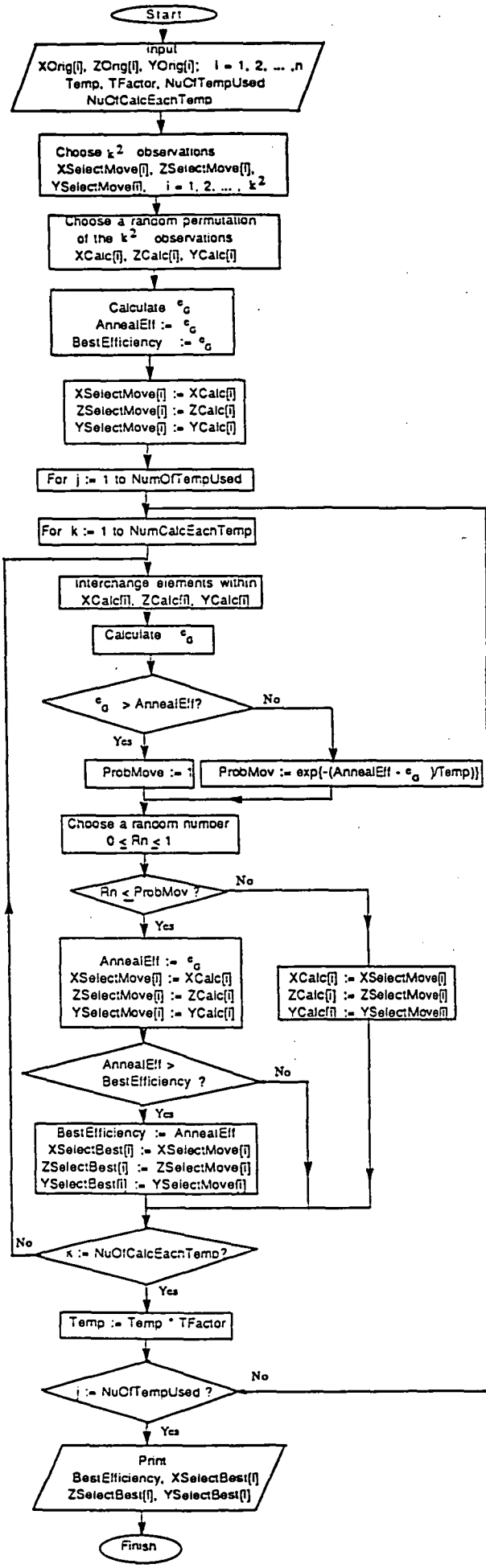

Figure 3.3
The flow–chart of the simulated annealing method

3.4. A Numerical Example.

To illustrate how the method works, and to demonstrate that the necessary computer programming is relatively straightforward, the data from Maritz (1981) page 194 was used, where observations were modulus of rigidity of timber specimens, and design variables were

x air dried density
z modulus of elasticity, and
y modulus of rigidity.

There were $n = 50$ observations, which is convenient for neighbour–searching with either

a single set and $k = 7 < 50^{1/2}$, or
two sets each with $k = 5 = 25^{1/2}$.

Table 1 shows the best e_G values obtained in several runs of the various methods. For neighbour searches, the number of evaluations of e_G was $N = 9{,}000$ and for simulated annealing, $N = 30{,}000$.

Table 3.1.
Efficiency of GLS to LS method
using Neighbour–searches and simulated annealing method.

(i) Neighbour–search with single set, k = 7

s = 1	s = 10	s = 25	s = 100	s = 1096
0.9526	0.9453	0.9453	0.9125	0.9136
0.9304	0.9453	0.9453	0.9310	0.9426
0.9125	0.9316	0.9371	0.9375	0.9125

(ii) Neighbour search with two sets , k = 5

s = 1
0.9332
0.9318
0.9521

(iii) Simulated annealing

0.9702
0.9730
0.9762

For this example, the best performance is by the simulated annealing method.

We now continue to give a complete solution for this example. Suppose we choose the grouping which gave efficiency $e_G = 0.9762$. The groups of observations are presented in matrix form as follows

$$X = \begin{bmatrix} 40.1 & 30.7 & 42.5 & 36.8 & 61.3 & 50.2 & 40.3 \\ 68.1 & 58.6 & 56.9 & 63.2 & 31.4 & 59.5 & 25.3 \\ 42.3 & 40.3 & 29.1 & 32.5 & 54.9 & 43.0 & 50.3 \\ 39.6 & 51.3 & 51.7 & 58.7 & 42.4 & 68.9 & 53.8 \\ 68.9 & 43.0 & 53.9 & 55.3 & 28.6 & 52.8 & 28.2 \\ 63.3 & 55.1 & 58.3 & 39.0 & 55.2 & 46.7 & 40.6 \\ 37.1 & 38.3 & 60.8 & 49.0 & 57.3 & 50.3 & 61.5 \end{bmatrix},$$

$$Z = \begin{bmatrix} 203 & 146 & 189 & 194 & 272 & 228 & 193 \\ 205 & 264 & 276 & 196 & 91 & 223 & 99 \\ 238 & 167 & 133 & 188 & 252 & 213 & 240 \\ 110 & 248 & 261 & 189 & 130 & 246 & 186 \\ 346 & 165 & 188 & 274 & 188 & 245 & 173 \\ 268 & 222 & 238 & 182 & 244 & 210 & 188 \\ 195 & 177 & 245 & 224 & 254 & 209 & 264 \end{bmatrix} \text{ and}$$

$$Y = \begin{bmatrix} 1587 & 1069 & 1492 & 1306 & 2054 & 1728 & 1145 \\ 1767 & 2036 & 1916 & 1746 & 925 & 1474 & 1000 \\ 1595 & 1438 & 1087 & 1306 & 1990 & 1605 & 1897 \\ 1254 & 1822 & 2129 & 2570 & 1129 & 2159 & 1676 \\ 2649 & 1647 & 1621 & 2086 & 1033 & 2053 & 1112 \\ 2604 & 1764 & 1870 & 1332 & 1909 & 1539 & 1281 \\ 1323 & 1379 & 2116 & 1706 & 1889 & 1703 & 1994 \end{bmatrix}$$

where the observations in the same row are in the same group. By using (3.5) we obtain

$$\underline{\lambda}^T = 10^{-3}\,(8.5919, -3.5517, -4.8352, 6.4375, 6.8459, 2.8427, 4.2365)$$

and therefore, by (3.6),

$\underline{z}^{*T} = (4.8885,\ 2.4273,\ 5.3664,\ 2.3963,\ 5.9581,\ 4.5989,\ 4.7556)$ and
$\underline{y}^{*T} = (34.8566,\ 24.6856,\ 37.9711,\ 31.5204,\ 40.1202,\ 38.5121,\ 34.4414)$.

In the model

$$y_i = \mu + \alpha x_i + \beta z_i + \epsilon_i$$

with independent errors having zero expectations and common variance σ^2, from Appendix 1 we see that the full least squares estimate of β is

3.319 with estimated standard error 0.81

After grouping and reducing the model to

$$y_i^* = \mu^* + \beta z_i^* + \epsilon_i^*,$$

the least squares estimate of β is

3.263 with standard error 0.85

The discrepancy between the full least squares and the grouped least squares point estimate of β is 0.056, with pooled standard error ≥ 0.81, and this is almost certainly due to sampling variation. From the two standard errors above, we can calculate an emperical relative efficiency of grouped to ungrouped least squares, that is $e = (0.81)^2/(0.85)^2 = 0.9081$, which is different to

$e_G = 0.9762$. The discrepancy between the two efficiencies can be explained as follows. From Appendix 1 , we see that estimates of variance from ANOVA error mean squares are

$$\text{var}(\epsilon_i) = \hat{\sigma}^2 = 34{,}963.286 ,$$

and for the grouped case

$$\text{var}(\epsilon^*_i) = \hat{\sigma}^{*2} = 8.419 .$$

Also we can calculate $\sum_{i=1}^{7} \lambda_i^2 = 2.24152 \times 10^{-4}$. The two efficiencies would be the same if the assumption–estimate $\hat{\sigma}^{*2} = \hat{\sigma}^2 \sum_{i=1}^{7} \lambda_i^2$ was satisfied. Here we have $\hat{\sigma}^2 \sum_{i=1}^{7} \lambda_i^2 = 7.8371 < \hat{\sigma}^{*2} = 8.419$, which explains why $e_G > e$.

Any exact method of inference for slope β in the simple linear regression model with y^* regressed z^* could now be used, and for this example we use the Brown and Maritz method (see Section 2.4). Since there are 7 observations, we have $\binom{7}{2} = 21$ values of β_{ij}. The point estimate and the confidence interval $\hat{\beta}$ can be obtained more easily by using the graph of $S(\beta)$. The values of β_{ij} and their weights can be seen in Table 3.2.

Table 3.2.
The values of β_{ij} in ascending order and their weights.

β_{ij}	Weight
−220.5673	0.0310
−32.3527	0.1567
−12.6198	0.2897
−0.7049	0.7675
0.8142	2.3593
1.1831	1.3592
1.3387	2.4923
2.1719	2.9701
2.4145	3.5618
3.1743	2.2026
3.6321	0.5917
3.7605	2.3283
4.1324	2.4613
4.3714	3.5308
4.5203	2.9371
4.9124	1.0695
5.5543	1.2025
6.3669	2.1716
6.5179	0.4778
7.4166	0.6108
10.6470	0.1329

The formula (2.16) and (2.17) gives $S(-\infty) = 16.8532$ and $S(\infty) = -16.8532$ and the graph is shown in Figure 3.1. The graph shows that the solution for $S = 0$ is $\hat{\beta} = 3.761$. From (2.15), we obtain $\text{var}(S) = 54.361$ and therefore the standard deviation is 7.373. The 90 percent confidence interval of β can be calculated as follows

$$P(-1.645 < \frac{S}{7.373} < 1.645) = 0.90$$

or

$$P(-12.129 < S < 12.129) = 0.90$$

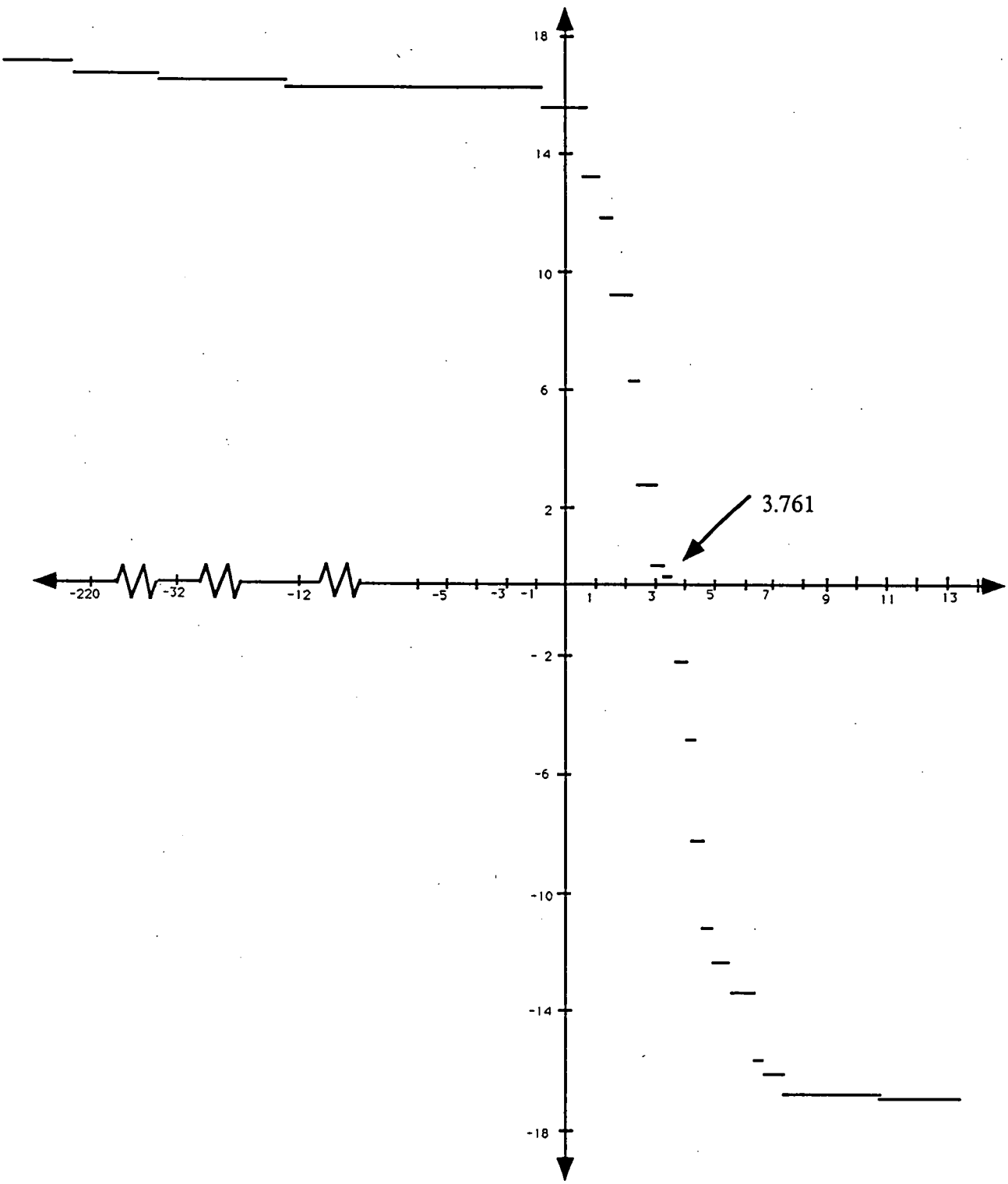

Figure 3.1

The graph of the funtion $S(\beta)$.

and from the figure 3.1, we can obtain the 90 percent confidence interval of β,

$$1.1831 < \beta < 5.9214.$$

By the same method if the slope parameter α is of interest, with $e_G = 0.9764$ (the grouping matrices can be seen in Appendix 2), we obtain a point estimate of $\alpha = 16.764$ and the 90 percent confidence interval of α is (13.367, 28.756).

The next table displays the estimate of slope parameters α and β and their confidence interval by using the full least squares, grouped least squares and the proposed method.

Table 3.3.
The estimation of slope parameters α and β by using full least squares, grouped least squares and the proposed method

Parameter	Method	Point Estimate	Confidence interval (90 %)
α	Full LS	20.305	(14.474, 26.136)
	Grouped LS	20.026	(9.329, 30.722)
	Proposed *)	16.764	(13.367, 28.756)
β	Full LS	3.319	(1.959, 4.680)
	Grouped LS	3.263	(1.550, 4.977)
	Proposed **)	3.761	(1.183, 4.921)

Note : *) $e_G = 0.9764$ and **) $e_G = 0.9762$

Table 3.3 shows that each 90 percent confidence interval of each slope parameter contains the three point estimates for that parameter. If the least squares assumptions hold, by using the *t test* with the same level of significance there is no reason to say that the three point estimates are different.

CHAPTER 4

GENERAL LINEAR REGRESSION

In this chapter the method in Chapter 3 will be extended to situations where more than two independent design variables are taken into account. In order to avoid any confusion in notation when the observations are grouped, the general linear regression (1.1) is written again using superscript notation for the independent design variables as follows

$$y_i = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \ldots + \beta_p x_i^p + \epsilon_i, \quad i = 1, 2, \ldots, n \tag{4.1}$$

where n is the number of observations, $\beta_0, \beta_1, \ldots, \beta_p$ are unknown parameters, $x_i^1, x_i^2, \ldots, x_i^p$ are design constants, $\{\epsilon_i\}$ are independent errors and identically distributed. Without loss of generality suppose that β_1 is of interest and the other β_i $(i \neq 1)$ are the nuisance parameters. The next step is to eliminate the nuisance parameters such that the general linear equation (4.1) can be reduced to a simple linear form.

2.1. Method of Parameter Elimination.

To eliminate the nuisance parameters β_i $(i \neq 0 \text{ or } 1)$ and reduce the model to SLR form, the observations are placed into

$$k = [\{n/(p-1)\}^{1/2}] \tag{4.2}$$

groups, where n is the number of observations and p is the number of

independent design variables. Thus the number of observations allocated in each group is $m = (p-1)k$.

Let $\lambda_1, \lambda_2, \ldots, \lambda_m$ be constants, and for $i = 1, 2, \ldots, k$ define

$$y_i^* = \sum_{j=1}^{m} \lambda_j y_{j+m(i-1)} = \beta_0 \sum_{j=1}^{m} \lambda_j + \beta_1 \sum_{j=1}^{m} \lambda_j x^1_{j+m(i-1)} + \beta_2 \sum_{j=1}^{m} \lambda_j x^2_{j+m(i-1)} + \ldots + \beta_p \sum_{j=1}^{m} \lambda_j x^p_{j+m(i-1)} + \sum_{j=1}^{m} \lambda_j \epsilon_{j+m(i-1)} \qquad (4.3)$$

Suppose that $\{\lambda_j\}$ are chosen so that for all $r = 2, 3, \ldots, p$

$$\sum_{j=1}^{m} \lambda_j x^r_j = \sum_{j=1}^{m} \lambda_j x^r_{j+m} = \sum_{j=1}^{m} \lambda_j x^r_{j+2m} = \ldots = \text{constant}, \; x^*_r \text{ say.} \qquad (4.4)$$

Then from (4.1), (4.3) and (4.4)

$$y_i^* = \beta_0^* + \beta z_i^* + \epsilon_i^* \qquad (4.5)$$

where $y_i^* = \sum_{j=1}^{m} \lambda_j y_{j+m(i-1)}$, $\beta_0^* = \beta_0 c + \sum_{j=2}^{m} \beta_j x_j^*$, $c = \sum_{j=1}^{m} \lambda_j$, $z_i^* = \sum_{j=1}^{m} \lambda_j x^1_{j+m(i-1)}$ and $\epsilon_i^* = \sum_{j=1}^{m} \lambda_j \epsilon_{j+m(i-1)}$. The equation (4.5) is of SLR form because $\{\epsilon_i^*\}$ are independent and identically distributed, and so can be treated by exact DF methods.

How can $\{\lambda_j\}$ and $\{x_r^*\}$ be chosen so that (4.4) holds? Let $k \times m$ matrices $X_2, X_3, \ldots, X_p$ be such that

$$X_2 = (X_2)_{i,j} = x^2_{j+(i-1)k}$$

$$X_3 = (X_3)_{i,j} = x^3_{j+(i-1)k}$$

...

$$X_p = (X_p)_{i,j} = x^p_{j+(i-1)k}$$

If $\underline{\lambda}^T = (\lambda_1, \lambda_2, \ldots, \lambda_m)$, $\underline{u}^T = (x^*_2, x^*_2 \ldots, x^*_2, \ldots, x^*_p, x^*_p, \ldots x^*_p)$ and X is an m x m matrix such that

$$X = \begin{bmatrix} X_2 \\ \hline X_3 \\ \hline \vdots \\ \hline X_p \end{bmatrix}$$

then we need to find $\underline{\lambda}$ and $\underline{u}$ satisfying (4.4), i.e.

$$X\underline{\lambda} = \underline{u},$$

and if X is a non–singular matrix we get

$$\underline{\lambda} = X^{-1}\underline{u}. \qquad (4.6)$$

Note that there is some choice available in finding a suitable vector of multipliers $\underline{\lambda}$. From (4.6), any choice of the vector $\underline{u}$ will suffice.

After defining a suitable $\underline{\lambda}$, the k pairs of values $\{z_i^*, y_i^*\}$ for the SLR model (4.5) can be obtained more easily by using matrix notation.

Let $k \times m$ matrices Z and Y be such that $(Z)_{i,j} = x^1_{j+m(i-1)}$ and $(Y)_{i,j} = y_{j+m(i-1)}$. If $\underline{z}^{*T} = (z_1^*, z_2^*, \ldots, z_k^*)$ and $\underline{y}^{*T} = (y_1^*, y_2^*, \ldots, y_k^*)$ then

$$\begin{aligned} \underline{z}^* &= Z\,\underline{\lambda} \\ \underline{y}^* &= Y\,\underline{\lambda} \end{aligned} \tag{4.7}$$

We have reduced the general linear model (4.1) with n observations to the simple linear regression model (4.5) with k observations.

4.2 Minimizing the Loss of Efficiency.

After reducing the model, the further steps are similar to the steps which we discussed in Chapter 3. Our aim is to maximize the grouping efficiency e_G, and to that end the approximation methods for finding the best grouping as described in Section 3.3 can be used.

If equation (4.1) is written in a matrix notation, we have $\underline{y} = A\,\underline{\theta} + \underline{\epsilon}$, where

$$A = \begin{bmatrix} 1 & x_1^1 & x_1^2 & \ldots & x_1^P \\ 1 & x_2^1 & x_2^2 & \ldots & x_2^P \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & x_n^1 & x_n^2 & \ldots & x_n^P \end{bmatrix},$$

$\underline{y}^T = (y_1, y_2, \ldots, y_n)$, $\underline{\theta}^T = (\beta_0, \beta_1, \ldots, \beta_p)$ and $\underline{\epsilon}^T = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$. If $(\hat{\beta}_1)_{LS}$ is the ungrouped least squares estimate of β_1, then from (2.7)

$$\text{var}\{(\hat{\beta}_1)_{LS}\} = \sigma^2\{A^TA)^{-1}\}_{22}$$

where σ^2 is the observational error variance. To get $\text{var}\{(\hat{\beta}_1)_{GLS}\}$, where $(\hat{\beta}_1)_{GLS}$ is the least squares estimate of β_1 after grouping, we write

$$\underline{y}^* = A^* \underline{\theta} + \underline{\epsilon}$$

where $\underline{y}^{*T} = (y_1^*, y_2^*, \ldots, y_k^*)$,

$$A^* = \begin{bmatrix} 1 & z_1^* \\ 1 & z_2^* \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & z_k^* \end{bmatrix},$$

$\underline{\theta}^{*T} = (\beta_0^*, \beta_1)$ and $\underline{\epsilon}^{*T} = (\epsilon_1^*, \epsilon_2^*, \ldots, \epsilon_k^*)$, and by (2.7)

$$\text{var}\{(\hat{\beta}_1)_{GLS}\} = \sigma^{*2}\{(A^{*T}A^*)^{-1}\}_{22}$$

where $\sigma^{*2} = \text{var}(\epsilon_i^*) = \sigma^2 \sum_{j=1}^{m} \lambda_j^2$. Thus the relative efficiency of grouped to ungrouped least squares is

$$e_G = \frac{\{(A^TA)^{-1}\}_{22}}{\sum_{j=1}^{m} \lambda_j^2 \{(A^{*T}A^*)^{-1}\}_{22}} . \qquad (4.8)$$

To maximize e_G , $\text{var}\{(\hat{\beta}_1)_{GLS}\}$ must be minimized, that is groups are to be chosen to minimize

$$\{A^{*T}A^*\}^{-1}\}_{22} \sum_{j=1}^{m} \lambda_j^2 = \frac{\sum_{j=1}^{m} \lambda_j^2}{\sum_{j=1}^{m} z_j^{*2} - (\sum_{j=1}^{m} z_j^*)^2/k} . \qquad (4.9)$$

The values of (4.9) depend on the choice of $\underline{\lambda}$ and $\underline{u}$ corresponding to allocation of the observations into the k groups.

The vector $\underline{u}^T = (x_2^*, x_2^*, \ldots , x_2^*, \ldots . , x_p^*, x_p^*, \ldots . x_p^*)$ can be written as

$$\underline{u} = V \underline{u}^* \qquad (4.10)$$

where $\underline{u}^{*T} = (x_2^*, x_3^*, \ldots , x_p^*)$, and V is an m x (p–1) matrix such that if $\underline{1}_k^T = (1, 1, \ldots , 1)$ and $\underline{0}_k^T = (0, 0, \ldots , 0)$ then

$$V = \begin{bmatrix} \underline{1}_k & \underline{0}_k & \cdots & \underline{0}_k \\ \underline{0}_k & \underline{1}_k & \cdots & \underline{0}_k \\ \vdots & \vdots & & \vdots \\ \underline{0}_k & \underline{0}_k & \cdots & \underline{1}_k \end{bmatrix}.$$

Now (4.6) can be written as

$$\underline{\lambda} = X^{-1} V \underline{u}^* = M \underline{u}^* \qquad (4.11)$$

where $M = X^{-1} V$. By (4.7) and (4.11) each term of the the denominator of (4.9) can be writen as

$$\sum_{j=1}^{m} z_j^{*2} = \underline{z}^{*T} \underline{z}^* = \underline{u}^{*T} M^T Z^T Z M \underline{u}^*$$

and

$$(\sum_{j=1}^{m} z_j^*)^2/k = (\underline{z}^{*T} \underline{1}\,\underline{1}^T \underline{z}^*)/k = (\underline{u}^{*T} M^T Z^T \underline{1}\,\underline{1}^T Z M \underline{u}^*)/k$$

where $\underline{1}^T = (1, 1, \ldots, 1)$. Now we can write (4.9) as

$$\{(A^{*T}A^*)^{-1}\}_{22} \sum_{j=1}^{m} \lambda_j^2 = \frac{\underline{u}^{*T} M^T M \underline{u}^*}{\underline{u}^{*T} M^T Z^T P Z M \underline{u}^*} \qquad (4.12)$$

where $P = I - (\underline{1}\,\underline{1}^T)/k$ and I is the k x k identity matrix. To get the minimum solution of (4.12), we minimize

$$\rho = \frac{\underline{u}^{*T}\ C\ \underline{u}^*}{\underline{u}^{*T}\ B\ \underline{u}^*} \tag{4.13}$$

where $C = M^T M$, $B = M^T Z^T P Z M$. By differentiating ρ with respect to the elements of $\underline{u}^*$, and equating to zero, we obtain

$$(\underline{u}^{*T} B \underline{u}^*) C \underline{u}^* = (\underline{u}^{*T} C \underline{u}^*) B \underline{u}^*$$

that is

$$(C - \rho B) \underline{u}^* = \underline{0} . \tag{4.14}$$

In general C and B are non–singular, so there are two possibilities for solving (4.14); that is

(i) $$B (B^{-1}C - \rho I) \underline{u}^* = \underline{0} \tag{4.15}$$

where $\underline{u}^*$ is an eigenvector of $B^{-1} C$ corresponding to the eigenvalue ρ , and since our aim is to minimize ρ , ρ must be the minimum eigenvalue of $B^{-1} C$;

and

(ii) $$\rho C (\rho^{-1} I - C^{-1} B) \underline{u}^* = \underline{0} \tag{4.16}$$

where $\underline{u}^*$ is an eigenvector of $C^{-1} B$ corresponding to the eigenvalue ρ^{-1}, so to minimize ρ, ρ^{-1} must be the maximum eigenvalue of $C^{-1} B$.

So to choose e_G optimally,

(i) find the minimum eigenvalue ρ of

$$B^{-1} C = (M^T Z^T P Z M)^{-1} M^T M,$$

or the maximum eigenvalue ρ^{-1} of

$$C^{-1} B = (M^T M)^{-1} M^T Z^T P Z M.$$

(ii) set $\underline{u}^*$ = the corresponding eigenvector, and the resulting maximal value of e_G, from (4.12), (4.9) and (4.8), is

$$e_G = \rho^{-1} \{(A^T A)^{-1}\}_{22}. \qquad (4.17)$$

Now we can maximize the grouping efficiency e_G by using one of the three methods of finding approximate solutions as oulined in Section 3.3.

4.3. A Numerical Example.

This example is taken from Feldman et al (1986) page 31 (demonstration data) where the observations are the cholesterol (y), cholesterol at 1 years old (x_i^1), weight (x_i^2) and the tryglicerides (x_i^3). With model $y_i = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i$, suppose β_1 is of interest. Here m = 2,

so the 50 observations are divided into $k = [50/2]^{1/2} = 5$ groups, each containing 10 observations. Because β_1 is of interest, let Y, Z, X_2, and X_3 be 5×10 matrices such that

$$Y = (Y)_{i,j} = (y)_{j+5(i-1)}$$

$$Z = (Z)_{i,j} = (x^1_i)_{j+5(i-1)}$$

$$X_2 = (X_2)_{i,j} = (x^2_i)_{j+5(i-1)}$$

$$X_3 = (X_3)_{i,j} = (x^3_i)_{j+5(i-1)}$$

and then form the 10×10 matrix

$$X = \begin{bmatrix} X_2 \\ \hline X_3 \end{bmatrix}.$$

After using the simulated annealing method, we obtained an approximation of the best grouping which gave efficiency $e_G = 0.9374$ with

(i) $\rho^{-1} = 32832.50622848$,

(ii) $C^{-1} B = \begin{bmatrix} 97614.6279807 & -116468.3150964 \\ 54101.6117438 & -64433.8906319 \end{bmatrix}$, and

(iii) the grouping matrices as follows

$$X = \begin{bmatrix} 139 & 149 & 134 & 168 & 162 & 110 & 152 & 172 & 170 & 177 \\ 183 & 156 & 116 & 191 & 170 & 138 & 146 & 154 & 201 & 122 \\ 123 & 178 & 155 & 168 & 165 & 175 & 154 & 167 & 187 & 173 \\ 125 & 205 & 160 & 158 & 165 & 162 & 177 & 153 & 201 & 166 \\ 163 & 150 & 192 & 208 & 187 & 115 & 150 & 121 & 136 & 154 \\ 148 & 61 & 53 & 98 & 79 & 65 & 60 & 100 & 105 & 135 \\ 92 & 81 & 118 & 57 & 88 & 71 & 68 & 64 & 85 & 79 \\ 59 & 85 & 69 & 96 & 91 & 118 & 69 & 105 & 95 & 82 \\ 85 & 53 & 103 & 57 & 116 & 98 & 32 & 96 & 79 & 70 \\ 59 & 167 & 47 & 65 & 85 & 184 & 73 & 105 & 145 & 89 \end{bmatrix},$$

$$Z = \begin{bmatrix} 173 & 142 & 135 & 178 & 203 & 176 & 185 & 134 & 191 & 229 \\ 179 & 171 & 180 & 167 & 185 & 176 & 172 & 148 & 175 & 185 \\ 187 & 209 & 223 & 210 & 137 & 186 & 149 & 273 & 182 & 160 \\ 165 & 139 & 159 & 145 & 228 & 189 & 179 & 136 & 177 & 200 \\ 190 & 249 & 172 & 182 & 162 & 224 & 167 & 244 & 200 & 175 \end{bmatrix},$$

$$Y = \begin{bmatrix} 172 & 142 & 133 & 158 & 192 & 169 & 178 & 134 & 184 & 219 \\ 172 & 151 & 170 & 167 & 180 & 171 & 161 & 148 & 162 & 180 \\ 177 & 189 & 201 & 190 & 137 & 166 & 145 & 253 & 182 & 168 \\ 164 & 129 & 152 & 135 & 208 & 182 & 159 & 130 & 188 & 221 \\ 188 & 222 & 167 & 172 & 155 & 219 & 167 & 224 & 191 & 145 \end{bmatrix}.$$

Note that all of the notations and definitions are the same as in Section 4.2.

Thus an eigenvector of $C^{-1} B$ corresponding to ρ^{-1} is

$$\underline{u}^{*T} = (1, 0.5562210).$$

By using (4.11) we obtain

$$\underline{\lambda}^T = 10^{-3}(1.541370, 2.562477, 3.256696, 1.348090, -2.749399, -1.049215,$$
$$-1.332395, 5.295262, -0.5206508, -1.795876)$$

and then by (4.7)

$$\underline{z}^{*T} = (0.519725,\ 0.963300,\ 2.126295,\ 0.528960\ ,1.707120)$$
$$\underline{y}^{*T} = (0.553226,\ 0.918092,\ 1.867065,\ 0.479284,\ 1.582235)\ .$$

In the model $y_i = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i$ with independent errors having zero expectation and common variance σ^2 , from Appendix 3 the full least squares estimate of β is

0.853 with estimated standard error 0.047 .

After grouping and reducing the model to $y_i^* = \beta_0^* + \beta_1 z^* + \epsilon_i^*$, the least squares estimate of β becomes

0.858 with estimated standard error 0.033 .

As in the numerical example of Section 3.4 , the discrepancy between the full least squares and the grouped least squares point estimate of β is 0.005 with pooled standard error > 0.033 , and is almost certainly due to sampling variation. The discrepancy between $e_G = 0.9365$, and an empirical relative efficiency calculated from the two standard errors above, that is $e = 2.0285$, is related to the fact that the assumption–estimate $\hat{\sigma}^{*2} = \hat{\sigma}^2 \sum_{j=1}^{5} \lambda_j^2$ was not satisfied . From Appendix 3 we see that estimates of variance from ANOVA error mean squares are

$$\text{var}(\epsilon_i) = 78.113 ,$$

and for the grouped case

$$\text{var}(\epsilon_i^*) = 0.002 .$$

Also we can calculate $\sum_{i=1}^{5} \lambda_i^2 = 6.333690 \times 10^{-5}$, so

$\hat{\sigma}^2 \sum_{i=1}^{5} \lambda_i^2 = 0.005 > \hat{\sigma}^{*2} = 0.002$ which explains why $e_G < e$.

The Brown and Maritz method for slope in simple linear regression now can be employed to estimate β_1. Since there are 5 pairs of new observations (z_i^*, y_i^*) , we have $\binom{5}{2} = 10$ values of β_{ij}. The values of β_{ij} in ascending order and their weights are shown in Table 4.1.

Table 4.1.
The values of β_{ij} in ascending order
and their weights.

β_{ij}	Weight
−8.0067	0.0092
0.6795	0.4192
0.8160	1.1630
0.8178	1.6066
0.8226	0.4436
0.8666	1.1874
0.8688	1.5973
0.8929	0.7438
0.9362	1.1782
1.0103	0.4343

The point estimate and the confidence interval of β_1 can be obtained using

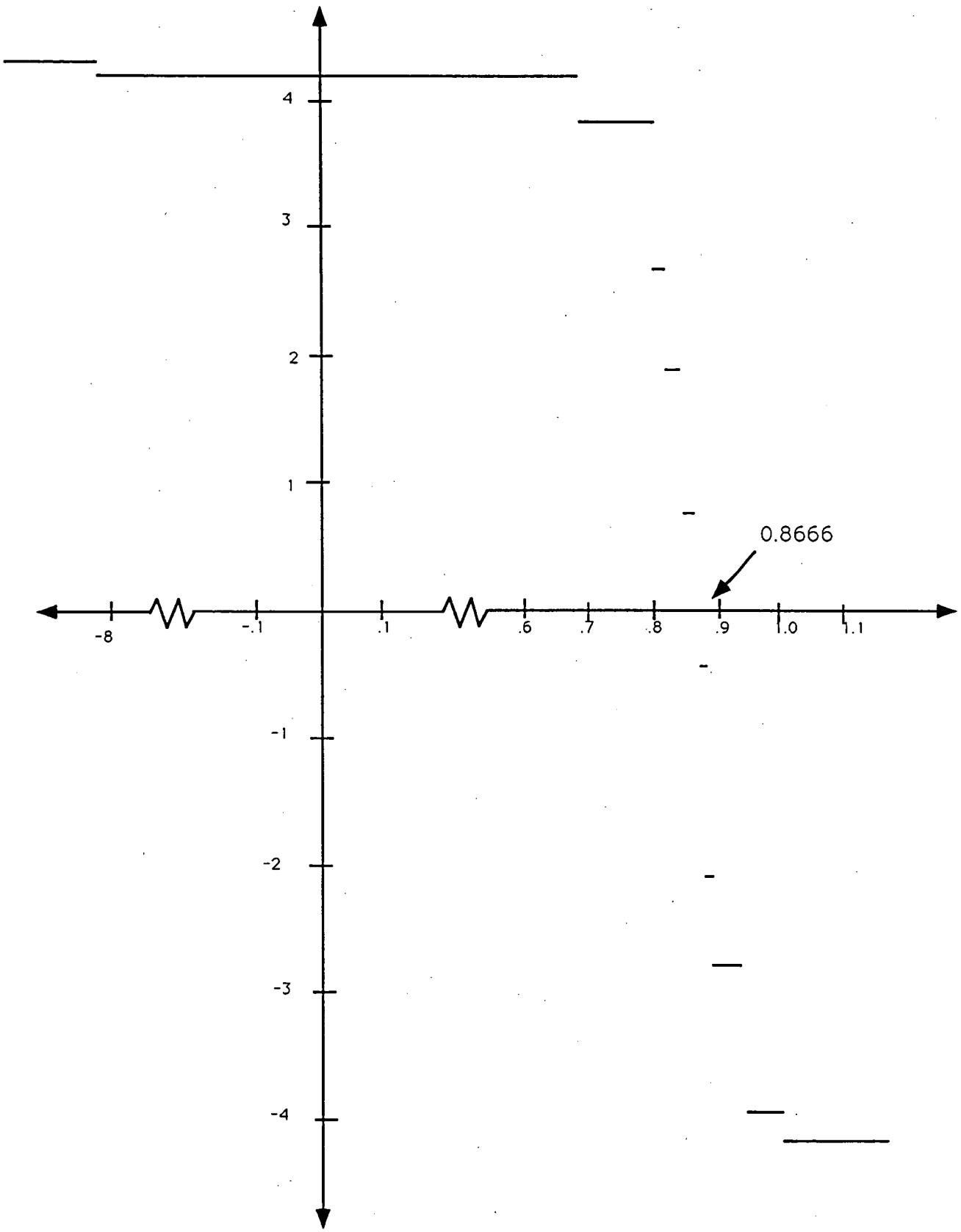

Figure 4.1

The graph of the funtion $S(\beta_1)$.

the graph of the function $S(\beta_1)$. The formula (2.16) and (2.17) give $S(-\infty) = 4.3913$ and $S(\infty) = -4.3913$ respectively, and the graph of $S(\beta_1)$ is shown in Figure 4.1. The graphs shows that the solution for $S = 0$ is $\hat{\beta}_1 = 0.8666$. From (2.15), we obtain $var(S) = 5.1988$ and therefore the standard deviation 2.2801 . The 90 percent confidence interval of β can be calculated as follows

$$P(-1.645 < \frac{S}{2.2801} < 1.6445) = 0.90$$

or

$$P(-3.7507 < S < 3.7507) = 0.90$$

and from Figure 4.1., we obtain the 90 percent confidence interval of β ,

$$0.680 < \beta < 0.893 .$$

The following table shows the estimate of slope parameter β_1 and its confidence interval using the full least squares, grouped least squares and the proposed method.

Table 4.1.
The estimation of the slope parameter β_1 using
full LS, grouped LS and the proposed method.

Method	Point estimate	Conf. Interval(90 %)
Full LS	0.853	(0.774, 0.933)
Grouped LS	0.858	(0.781, 0.934)
Proposed	0.867	(0.680, 0.893)

The above table shows that each confidence interval contains the three point estimates of β_1, and if the least squares assumptions hold, with the same level of significance, then there is no reason to say that the three point estimates are different.

CHAPTER 5

DISCUSSION AND CONCLUSION

5.1. Discussion.

In this section we discuss another possible way of grouping the observations in planar regression (see Section 3.1), a possible way to sidestep the difficulties of obtaining the eigenvector ρ for example when both of the matrices $M^T Z^T P Z A$ and $M^T M$ are "nearly" singular (see Section 4.2), and a possible method to speed up the the simulated annealing convergence for a relatively large sample size.

As an effect of grouping of observations, usually all of the observations cannot be used to estimate the parameter of interest, and therefore the observations used to estimate the parameter of interest must be chosen randomly from the observations available. The numerical example of Section 3.4 shows that just 49 of 50 data available were used. If the number of data used is less than the number of observations available, then there is a loss of information needed to estimate the slope parameter of interest. For planar regression, to maximize the number of observations used (and also the efficiency of grouping), we can combine a variation of neighbours search method (i.e. divide data into two or more separate sets and carry out the grouping operations separately within each set, see Section 3.3.2.) and the simulated annealing method. The variation of neighbours search method is used to maximize the number of data used and the simulated annealing method to maximize groupings efficiency. If observations are divided into m sets, k is the number of groups in each set, and n is the number of observations, then

$$k = \left[\frac{n}{m} \right]^{1/2}$$

The combined method helps us to maximize the number of data used, but implies extra work to estimate the parameter of interest. Because the m sets of observations give m independent estimates of the parameter of interest, then the parameter of interest can be estimated using a linear combination of the m estimators. In the numerical example of Section 3.5 , if we choose $m = 2$, all of the 50 observations can be used to estimate β.

In Section 4.2 , we maximize the grouping efficiency e_G by minimizing $\text{var}\{(\hat{\beta}_1)_{GLS}\}$, and in so doing need an approximation to the minimum eigenvalue ρ of $(M^T Z^T P Z A)^{-1} M^T M$ or the maximum eigenvalue ρ^{-1} of $(M^T M)^{-1} M^T Z^T P Z A$. After obtaining ρ or ρ^{-1} we calculate $\underline{u}^*$ as the corresponding eigenvector, but can encounter problems for example in case both of the matrices $M^T Z^T P Z A$ and $M^T M$ are nearly singular. To sidestep such problems we choose $\underline{u}^* = \underline{1}$. With this choice from (4.6) we obtain

$$\underline{\lambda} = X^{-1} \underline{1} .$$

Using the choice of $\underline{u}^* = 1$ to solve the numerical example of Section 4.3 , gave an approximation of the best efficiency $e_G = 0.9365$ (the grouping matrices can be seen in Appendix 4). This result is close to the result in Section 4.3 , that is $e_G = 0.9374$.

The proposed method involves solving a difficult combinatorial optimization problem with the simulated annealing method showing the best performance. The method is quite computer intensive and to give good results, lengthly computer runs may be necessary. Because the method of grouping used implies n! possible groupings (many will share the same value of e_G), the computing time needed to get an approximate solution will depend on sample size, the number of independent design variables and which approximation method is used.

For a relatively large sample size, to speed up the the simulated annealing convergence, instead of interchanging one or more pairs as described in Section 3.3.3 , we choose r observations at random (r also is chosen randomly, $2 \leq r \leq k^2$), and then permute the order of the r observations selected randomly. Preliminary experience shows that if r is restricted so that it is not too large (say $2 \leq r \leq 10$), then there is a considerable reduction in computing time.

5.2. Conclusion.

In this thesis we have proposed an exact distribution–free method of solving general linear regression problems, where one of the slope parameters is of interest, through grouping of observations to eliminate the nuisance parameters and reducing the model to simple linear regression form, and then using an exact distribution–free method for slope in simple linear regression.

Because the method merely relies on two broader and weaker assumptions about underlying distribution forms, that is, independent and identically distributed errors, the application of the proposed method therefore

involves all of the general linear regression problems with error terms satisfying the above assumptions.

The method is a simpler alternative to the Maritz–Theil approach, and also gives satisfactory efficiency, especially for the planar regression. The efficiency will decrease if the number of independent design variables increases. There are two factors causing a loss of efficiency, that is grouping and reducing the model (eliminating the nuisance parameters). Grouping eliminates the individual character of data in that it ignores the variation of data within each group and then replaces them with a new value, and reducing the model eliminates the individual effect of each nuisance parameter. The above information explains why the grouping efficiency will decrease if the number of independent design variable increases.

Three approximation methods for finding the best grouping to minimize efficiency loss are discussed. The methods are a *Monte Carlo* method, a *search for better neighbours,* and a *simulated annealing* method. The Monte Carlo method is easy to program and suitable for small and medium sized design, the search for better neighbours can get stuck in local optima but needs relatively less computing time compare to the simulated annealing method, and the simulated annealing method shows the best performance.

Appendix 1
The least squares solution of the numerical example of Section 3.4.

1. The full least squares.

Multiple Regression Y 1 :var y 2 X variables

Count:	R:	R-squared:	Adj. R-squared:	RMS Residual:
49	.9	.81	.802	186.985

Analysis of Variance Table

Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	2	6857621.081	3428810.54	98.069
RESIDUAL	46	1608311.164	34963.286	p = .0001
TOTAL	48	8465932.245		

No Residual Statistics Computed

Multiple Regression Y 1 :var y 2 X variables

Beta Coefficient Table

Variable:	Coefficient:	Std. Err.:	Std. Coeff.:	t-Value:	Probability:
INTERCEPT	-13.607				
var x	20.305	3.473	.565	5.846	.0001
var z	3.319	.81	.396	4.096	.0002

Multiple Regression Y 1 :var y 2 X variables

Confidence Intervals and Partial F Table

Variable:	95% Lower:	95% Upper:	90% Lower:	90% Upper:	Partial F:
INTERCEPT					
var x	13.312	27.297	14.474	26.136	34.175
var z	1.688	4.951	1.959	4.68	16.776

2. The grouped least squares.

Simple Regression X 1 : variable z Y 1 : variable y

Count:	R:	R-squared:	Adj. R-squared:	RMS Residual:
7	.864	.747	.696	2.902

Analysis of Variance Table

Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	1	124.061	124.061	14.735
RESIDUAL	5	42.097	8.419	p = .0121
TOTAL	6	166.159		

No Residual Statistics Computed

Simple Regression X 1 : variable z Y 1 : variable y

Beta Coefficient Table

Variable:	Coefficient:	Std. Err.:	Std. Coeff.:	t-Value:	Probability:
INTERCEPT	20.275				
SLOPE	3.263	.85	.864	3.839	.0121

Confidence Intervals Table

Variable:	95% Lower:	95% Upper:	90% Lower:	90% Upper:
MEAN (X,Y)	31.624	37.264	32.234	36.654
SLOPE	1.078	5.449	1.55	4.977

Appendix 2
The grouping matrices for slope parameter α
for the numerical example of Section 3.4

$$X = \begin{bmatrix} 146 & 223 & 213 & 130 & 205 & 196 & 186 \\ 222 & 346 & 209 & 188 & 188 & 194 & 133 \\ 245 & 264 & 210 & 252 & 167 & 272 & 91 \\ 165 & 248 & 264 & 99 & 188 & 195 & 261 \\ 182 & 110 & 276 & 245 & 246 & 189 & 188 \\ 244 & 224 & 254 & 228 & 193 & 189 & 240 \\ 268 & 274 & 238 & 177 & 173 & 238 & 203 \end{bmatrix}$$

$$Z = \begin{bmatrix} 30.7 & 59.5 & 43.0 & 42.4 & 68.1 & 63.2 & 53.8 \\ 55.1 & 68.9 & 50.3 & 40.6 & 28.6 & 36.8 & 29.1 \\ 52.8 & 58.6 & 46.7 & 54.9 & 40.3 & 61.3 & 31.4 \\ 43.0 & 51.3 & 61.5 & 25.3 & 32.5 & 37.1 & 51.7 \\ 39.0 & 39.6 & 56.9 & 60.8 & 68.9 & 58.7 & 53.9 \\ 55.2 & 49.0 & 57.3 & 50.2 & 40.3 & 42.5 & 50.3 \\ 63.3 & 55.3 & 58.3 & 38.3 & 28.2 & 42.3 & 40.1 \end{bmatrix}$$

$$Y = \begin{bmatrix} 1069 & 1474 & 1605 & 1129 & 1767 & 1746 & 1676 \\ 1764 & 2649 & 1703 & 1281 & 1033 & 1306 & 1087 \\ 2053 & 2036 & 1539 & 1990 & 1438 & 2054 & 925 \\ 1647 & 1822 & 1994 & 1000 & 1306 & 1323 & 2129 \\ 1332 & 1254 & 1916 & 2116 & 2159 & 2570 & 1621 \\ 1909 & 1706 & 1889 & 1728 & 1145 & 1492 & 1897 \\ 2604 & 2086 & 1870 & 1379 & 1112 & 1595 & 1587 \end{bmatrix}$$

Appendix 3
The least squares solution of the numerical example of Section 4.3.

1. The full least squares.

Multiple Regression Y 1 :Cholestrol 3 X variables

Count:	R:	R-squared:	Adj. R-squared:	RMS Residual:
50	.95	.903	.897	8.838

Analysis of Variance Table

Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	3	33478.596	11159.532	142.865
RESIDUAL	46	3593.184	78.113	p = .0001
TOTAL	49	37071.78		

No Residual Statistics Computed

Multiple Regression Y 1 :Cholestrol 3 X variables

Beta Coefficient Table

Variable:	Coefficient:	Std. Err.:	Std. Coeff.:	t-Value:	Probability:
INTERCEPT	17.65				
Chol-1yr	.853	.047	.948	18.068	.0001
Weight	.002	.054	.002	.036	.9716
Triglycerides	.005	.049	.006	.103	.9186

Multiple Regression Y 1 :Cholestrol 3 X variables

Confidence Intervals and Partial F Table

Variable:	95% Lower:	95% Upper:	90% Lower:	90% Upper:	Partial F:
INTERCEPT					
Chol-1yr	.758	.948	.774	.933	326.451
Weight	-.106	.11	-.088	.092	.001
Triglycerides	-.093	.103	-.077	.087	.011

2. The grouped least squares

Simple Regression X 1 : variable z* Y 1 : variable y*

Count:	R:	R-squared:	Adj. R-squared:	RMS Residual:
5	.998	.996	.994	.047

Analysis of Variance Table

Source	DF:	Sum Squares:	Mean Square:	F-test:
REGRESSION	1	1.53	1.53	694.48
RESIDUAL	3	.007	.002	p = .0001
TOTAL	4	1.536		

No Residual Statistics Computed

Simple Regression X 1 : variable z* Y 1 : variable y*

Beta Coefficient Table

Variable:	Coefficient:	Std. Err.:	Std. Coeff.:	t-Value:	Probability:
INTERCEPT	.077				
SLOPE	.858	.033	.998	26.353	.0001

Confidence Intervals Table

Variable:	95% Lower:	95% Upper:	90% Lower:	90% Upper:
MEAN (X,Y)	1.013	1.147	1.031	1.129
SLOPE	.754	.961	.781	.934

Appendix 4
The grouping matrices for the numerical example of Section 4.3 with the choice of $\underline{u}^* = \underline{1}$.

$$X = \begin{bmatrix} 183 & 163 & 149 & 138 & 116 & 150 & 155 & 175 & 136 & 154 \\ 122 & 156 & 162 & 150 & 177 & 173 & 172 & 177 & 192 & 160 \\ 125 & 178 & 165 & 170 & 134 & 121 & 167 & 158 & 168 & 139 \\ 191 & 115 & 208 & 154 & 170 & 187 & 205 & 205 & 146 & 152 \\ 201 & 154 & 165 & 187 & 123 & 160 & 153 & 110 & 162 & 166 \\ 92 & 59 & 61 & 71 & 118 & 167 & 69 & 118 & 145 & 69 \\ 79 & 81 & 79 & 73 & 32 & 82 & 100 & 135 & 47 & 96 \\ 85 & 85 & 91 & 105 & 53 & 105 & 105 & 57 & 98 & 148 \\ 57 & 184 & 65 & 89 & 88 & 85 & 53 & 85 & 68 & 60 \\ 79 & 64 & 116 & 95 & 59 & 103 & 96 & 65 & 98 & 70 \end{bmatrix}$$

$$Z = \begin{bmatrix} 179 & 190 & 142 & 176 & 180 & 249 & 223 & 186 & 200 & 149 \\ 185 & 171 & 203 & 167 & 179 & 160 & 134 & 229 & 172 & 210 \\ 165 & 209 & 137 & 191 & 135 & 244 & 273 & 145 & 178 & 173 \\ 167 & 224 & 182 & 175 & 185 & 162 & 139 & 175 & 172 & 185 \\ 177 & 148 & 228 & 182 & 187 & 159 & 136 & 176 & 189 & 200 \end{bmatrix}$$

$$Y = \begin{bmatrix} 172 & 188 & 142 & 171 & 170 & 222 & 201 & 166 & 191 & 145 \\ 180 & 151 & 192 & 167 & 159 & 168 & 134 & 219 & 167 & 190 \\ 164 & 189 & 137 & 184 & 133 & 224 & 253 & 135 & 158 & 172 \\ 167 & 219 & 172 & 145 & 180 & 155 & 129 & 162 & 161 & 178 \\ 188 & 148 & 208 & 182 & 177 & 152 & 130 & 169 & 182 & 221 \end{bmatrix}$$

Appendix 5
The computer program for solving
the planar regression problems

```
Program Maximize_Efficiency_Of_Grouping;
USES CRT, PRINTER;
LABEL 10;

Const
   NuOfdataAvailable          :   50;
   NuOfDataUsed               :   49;
   NuOfTemp                   :  100;
   NuOfRepeat                 :  250;
   NLimit                     :   20;
   T                          : 1000;
   k                          :    0.0001;
   Col                        :    3;
   Const1                     :    1;
   Const2                     :    2;
   NuOfGroup                  :    7;

Type
   DataType       = ARRAY[1..NuOfDataAvailable] Of Integer;
   DataType1      = ARRAY[1..NuOfDataAvailable] Of Real;
   MatrixType1    = ARRAY[1..NuOfDataUsed,1..Col] Of Real;
   MatrixType2    = ARRAY[1..Col,1..NuOfDataUsed] Of Real;
   MatrixType3    = ARRAY[1..Col,1..Col] Of Real;
   MatrixType4    = ARRAY[1..NuOfGroup,1..NuOfGroup] Of Real;
   MatrixType5    = ARRAY[1..NuOfDataUsed,1..Const1] Of Real;
   MatrixType6    = ARRAY[1..Const,1..NuOfGroup] Of Real;

Const
   XOrig : DataType1 =
                    (99,173,188,133,146,240,248,261,245,186,
                     91,188,194,195,177,188,252,222,244,274,
                     182,110,203,193,167,276,254,238,264,189,
                     188,238,130,189,213,223,245,272,264,196,
                     165,210,224,228,209,268,205,346,246,237.5);
   ZOrig : DataType1 =
                    (25.3,28.2,28.6,29.1,30.7,50.3,51.3,51.7,52.8,53.8,
                     31.4,32.5,36.8,37.1,38.3,53.9,54.9,55.1,55.2,55.3,
                     39.0,39.6,40.1,40.3,40.3,56.9,57.3,58.3,58.6,58.7,
                     40.6,42.3,42.4,42.5,43.0,59.5,60.8,61.3,61.5,63.2,
                     43.0,46.7,49.0,50.2,50.3,63.3,68.1,68.9,68.9,70.8);
   YOrig : DataType1 =
                    (1000,1112,1033,1087,1069,1897,1822,2129,2053,1676,
                     925,1306,1306,1323,1379,1621,1990,1764,1909,2086,
                     1332,1254,1587,1145,1438,1916,1889,1870,2036,2570,
                     1281,1595,1129,1492,1605,1474,2116,2054,1994,1746,
                     1647,1539,1706,1728,1703,2604,1767,2649,2159,2078);
```

```
Var
   XSelectMove,ZSelectMove,YSelectMove          : DataType1;
   XSelectBest,ZSelectBest,YSelectBest          : DataType1;
   XCalc,ZCalc,YCalc                            : DataType1;
   Identity3                                    : MatrixType3;
   X,Y,Z,IdentNuOfGroup                         : MatrixType4;
   Vect1                                        : MatrixType5;
   VarLeastSq,VarGLS,Temper,eG                  : Real;
   AnnealEff,Rn1,BestEfficiency,Tfactor         : Real;
   I,J,NSucc                                    : Integer;
   Select                                       : Char;

Procedure SwapInteger( Var p,q : Integer);
Var
   Temp : Integer;
Begin
   Temp := p; p := q; q := Temp;
End

Procedure SwapReal( Var p,q : Real);
Var
   Temp : Real;
Begin
   Temp := p; p := q; q := Temp;
End

Procedure SelectOfObservations;
Var
   I,Counter,Rand       : Integer;
   Temp                 : DataType;
Begin
   If NuOfDataUsed < NuOfDataAvailable Then
      Begin
         Rand := Trunc(Rand*NuOfDataAvailable)+1;
         Temp[1] := Rand;
         XSelectMove[1] := XOrig[Rand];
         ZSelectMove[1] := ZOrig[Rand];
         YSelectMove[1] := YOrig[Rand];
         Counter := 2;
         FOR I := 2 TO NuOfDataUsed DO
            Begin
               Rand := Trunc(Random*NuOfDataAvailable)+1;
               WHILE Counter < = I  DO
                  Begin
                     If Rand = Temp[Counter–1] Then
                        Begin
                           Rand := Trunc(Random*NuOfDataAvailable)+1;
                           Counter := 1;
                        End;
                     Counter := Counter + 1;
                  End;
```

```
                Temp[I] := Rand;
                XSelectMove[I] := XOrig[Rand];
                ZSelectMove[I] := ZOrig[Rand];
                YSelectMove[I] := YOrig[Rand];
                Counter := 2;
             End
          Else
             Begin
                XSelectMove := XOrig;
                ZSelectMove := ZOrig;
                YSelectMove := YOrig;
             End;
End;

Procedure RandomPermutation;
Var
   I.J,K    : Integer;
Begin
   XCalc := XSelectMove;
   ZCalc := ZSelectMove;
   YCalc := YSelectMove;
   FOR I := 2 TO NuOfDataUsed Do
      Begin
         K := Trunc(Random*I) + 1;
         SwapReal(XCalc[I],XCalc[K]);
         SwapReal(ZCalc[I],ZCalc[K]);
         SwapReal(YCalc[I],YCalc[K]);
      End;
End;

Procedure Linked;
Var
   I,J      : Integer;
Begin
   FOR I := 1 TO NuOfGroup DO
      Begin
         FOR J := 1 TO NuOfGroup DO
            Begin
               X[I,J] := XCalc[J+(I-1)*NuOfGroup];
               Z[I,J] := ZCalc[J+(I-1)*NuOfGroup];
               Y[I,J] := YCalc[J+(I-1)*NuOfGroup];
            End;
      End;
End;
```

```
Procedure IdentityMatrixAndVector1;
Var
   I,J      : Integer;
Begin
   FOR I := 1 TO Col DO
      Begin
         FOR J := 1 TO Col DO
            Begin
               IF I = J THEN Identity3[I,J] :=1
               Else Identity3 := 0;
            End;
      End;
   FOR I := 1 TO NuOfGroup DO
      Begin
         FOR J := 1 TO NuOfGroup DO
            Begin
               IF I = J THEN IdentNuOfGroup[I,J] :=1
               Else IdentNuOfGroup := 0;
            End;
      End;
   FOR I := 1 TO NuOfGroup DO
      Begin
         Vect1[I,Const1] := 1;
      End;
End;

Procedure CalcVarianceLeastSquares;
Var
   I,J,K                : Integer;
   Temp,V,R             : Real;
   A                    : MatrixType1;
   ATranpose            : MatrixType2;
   AtA,AtAInvers        : MatrixType3;
Begin
   FOR I := 1 TO NuOfDataUsed DO  {Form matrix A}
      Begin
         A[I,1] := 1;
         A[I,2] := XSelectMove[I];
         A[I,3] := ZSelectMove[I];
      End;
   FOR I := 1 TO NuOfDataUsed DO  {Form A transpose}
      Begin
         FOR J := 1 TO Col DO
            Begin
               ATranspose[J,I] := A[I,J];
            End;
      End;
```

```
   FOR I := 1 TO Col DO  {Calc  AtA = ATranspose * A}
      Begin
         FOR J := 1 TO Col DO
            Begin
               Temp := 0;
               FOR K := 1 TO NuOfDataUsed;
                  Begin
                     Temp := Temp+(ATranspose[I,K]*A[K,J]);
                     AtA := Temp;
                  End;
            End;
      End;
   AtAInvers := Identity3;  {Calc. AtAInvers}
   FOR I := 1 TO Col DO
      Begin
         IF AtA[I,I] < > 0 Then V := 1/AtA[I,I];
         FOR J := 1 TO Col DO
            Begin
               AtA[I,J] := V*AtA[I,J];
               AtAInvers[I,J] := V*AtAInvers[I,J];
            End;
         FOR J := 1 TO Col DO
            Begin
               IF J < > I Then
                  Begin
                     R := –AtA[J,I];
                     FOR K := 1 TO Col DO
                        Begin
                           AtA[J,K] := AtA[J,K] + R*AtA[I,K];
                           AtAInvers[J,K] := AtAInvers[J,K]+R*AtAInvers[I,K];
                        End;
                  End;
            End;
      End;
      VarLeastSq := AtAInvers[3,3];
End;

Procedure CalcVarianceGroupedLeastSquares;
Var
   I,J,K                              : Integer;
   V,R,m1,m2,Temp,SigmaLdSq           : Real;
   XInvers                            : MatrixType4;
   Vector1,Lamda,ZStar                : MatrixType5;
   ZStarTransp,LamdaTransp            : MatrixType6;
Begin
   XInvers := IdentNuOfGroup;  {Calc. XInvers}
   FOR I := 1 TO NuOfGroup DO
      Begin
         IF X[I,I] < > 0 Then V := 1/X[I,I];
```

```
      FOR J := 1 TO NuOfGroup DO
        Begin
          X[I,J] := V*X[I,J];
          XInvers[I,J] := V*XInvers[I,J];
        End;
      FOR J := 1 TO NuOfGroup DO
        Begin
          IF J < > I Then
            Begin
              R := –X[J,I];
              FOR K := 1 TO NuOfGroup DO
                Begin
                  X[J,K] := X[J,K] + R*X[I,K];
                  XInvers[J,K] := XInvers[J,K]+R*XInvers[I,K];
                End;
            End;
        End;
  End;
Vector1 : = Vect1; {Calc. Lamda}
FOR I:= 1 TO NuOfGroup DO
  Begin
    Temp := 0;
    FOR K := 1 TO NuOfGroup DO
      Begin
        Temp := Temp+(XInvers[I,K]*Vector1[K,Const1]);
        Lamda[I,Const1] := Temp;
      End;
  End;
FOR I := 1 NuOfGroup DO {Form Lamda Transpose}
  Begin
    LamdaTransp[Const1,I] := Lamda[I,Const1];
  End;
SigmaLdSq := 0; {Calc. SigmaLdSq = LamdaTramsp*Lamda}
FOR I := 1 TO NuOfGroup DO
  Begin
    SigmaLdSq := SigmaLdSq+(LamdaTransp[Const1,I]*Lamda[I,Const1]);
  End;
FOR I := 1 TO NuOfGroup DO { Calc. ZStar = Z * Lamda}
  Begin
    Temp := 0;
    FOR K := 1 TO NuOfGroup DO
      Begin
        Temp := Temp + (Z[I,K]*Lamda[K,Const1]);
        ZStar[I,Const1] := Temp;
      End;
  End;
FOR J := 1 TO NuOfGroup DO {Form ZStar Transpose}
  Begin
    ZStarTransp[Const1,J] := ZStar[J,Const1];
  End;
```

```
   m1 := 0;          {Calc. m1 = ZstarTransp * ZStar}
   FOR K := 1 TO NuOfGroup DO
      Begin
         m1 := m1+(ZStarTransp[Const1,K]*ZStar[K,Cost1]);
      End;
   m2 := 0; Vector1 := Vect1;  {Calc m2 = ZstarTransp * Vector1}
      FOR K := 1 TO NuOfGroup DO
      Begin
         m2 := m2+(ZStarTransp[Const1,K]*Vector1[K,Cost1]);
      End;
   VarGLS := SigmaLdSq/(m1-(m2*m2)/k);
End;

Procedure CalcEfficiency;
Begin
   eG := VarLeastSq/VarGLS;
End;

Procedure InterchangeOnePair;
Var
   I,a1,a2 : Integer;
Begin
   a1 := 1+Trunc(Random*NuOfDataUsed);
   a2 := 1+Trunc(Random*NuOfDataUsed);
   SwapReal(XCalc[a1],XCalc[a2]);
   SwapReal(ZCalc[a1],ZCalc[a2]);
   SwapReal(YCalc[a1],YCalc[a2]);
end;

Procedure RandomPermMElements;
Var
   I,J,K,MElem,Rand,Counter          : Integer;
   Temp1                             : Real;
Begin
   MElem := 0;
   WHILE (MElem < = 0) OR (MElem > 10) DO
   Begin
      MElem := Trunc(Random*NuOfDataUsed)+1;
   End;
   Rand := Trunc(Random*NuOfDataUsed)+1;
   Temp1 := Rand; Counter := 2;
   FOR I := 2 TO MElem DO
      Begin
         Rand := Trunc(Random*NuOfDataUsed)+1;
         WHILE Counter < = I  DO
            Begin
               IF Rand = Temp1[Counter-1] Then
                  Begin
                     Rand := Trunc(Random*NuOfDataUsed)+1;
                     Counter := 1;
                  End;
               Counter := Counter+1;
            End;
```

```
            Temp1[I] := Rand; Counter := 2;
          End;
    FOR I := 2 TO MElem DO
      Begin
        K : = Trunc(Random*I)+1;
        SwapReal(XCalc[Temp1[I]],XCalc[Temp1[K]]);
        SwapReal(ZCalc[Temp1[I]],ZCalc[Temp1[K]]);
        SwapReal(YCalc[Temp1[I]],YCalc[Temp1[K]]);
      End;
    End;

Procedure SelectOfEfficiency;
Var
   Rn2,DE,ProbMov      : Real;
Begin
   Rn2 := Random;
   DE := (eG – AnnealEf)/(k*Temper);
   IF DE < –50 Then ProbMov := 0
   ELSE IF (DE > = –50) AND (DE < 0) Then ProbMov := Exp(DE)
   ELSE ProbMove := 1;
   IF Rn2 < ProbMov Then
      Begin
        AnnealEff := eG;
        NSucc := NSucc+1;
        XSelectMove := XCalc;
        ZSelectMove := ZCalc;
        YSelectMove := YCalc
      End
   ELSE
      Begin
        XCalc := XSelectMove;
        ZCalc := ZSelectMove;
        YCalc := YSelectMove;
      End;
   IF AnnealEff > BestEfficiency Then
      Begin
        BestEfficiency := AnnealEff;
        XSelectBest := XSelectMove;
        ZSelectBest := ZSelectMove;
        YSelectBest := YSelectMove;
      End;
End;

Procedure PrintResult;
Var
   I,J  : Integer;
Begin
   Writeln(lst,'Best Efficiency = ',BestEfficiency:12:8);
   Writeln(lst,'Matrix  X = ');
```

```
   FOR I := 1 TO NuOfGroup DO
      Begin
         FOR J := 1 TO NuOfGroup DO
            Begin
               Write(lst,X[I,J]:8:2);
            End;
         Writeln(lst);
      End;
   Writeln(lst);
   Writeln(lst,'Matrix  Z = ');
   FOR I := 1 TO NuOfGroup DO
      Begin
         FOR J := 1 TO NuOfGroup DO
            Begin
               Write(lst,Z[I,J]:8:2);
            End;
         Writeln(lst);
      End;
Writeln(lst);
   Writeln(lst,'Matrix  Y = ');
    FOR I := 1 TO NuOfGroup DO
      Begin
         FOR J := 1 TO NuOfGroup DO
            Begin
               Write(lst,Y[I,J]:8:2);
            End;
         Writeln(lst);
      End;
End;


Begin {Main Program}
   ClrScr;
   Writeln('Program Menu '); Writeln;
   Writeln(' 1. The Monte Carlo method '); Writeln;
   Writeln(' 2. The Search for better neighbours '); Writeln;
   Writeln(' 3. The Simulated Annealing method '); Writeln;
   Write(' Select ? '); Read(Select);
   CASE Select OF
   '1' :
      Begin
         SelectOFObservations;
         IdentityMatrixAndVector1;
         CalcVarianceLeastSquares;
         BestEfficiency := 0;
         FOR I := 1 TO NuOfRepeat DO
            Begin
               RandomPermutation;
               Linked;l
               CalcVarianceGroupedLeastSquares;
```

```
            If eG >= BestEfficiency Then
               Begin
                  BestEfficiency := eG;
                  XSelectBest := XCalc;
                  ZSelectBest := ZCalc;
                  YSelectBest := YCalc;
               End;
         End;
   End;

'2' :
   Begin
      SelectOfObservations;
      IdentityMatrixAndVector1;
      CalcVarianceLeastSqures;
      RandomPermutation;
      Linked;
      CalcVarianceGroupedLeastSquares;
      CalcEfficiency;
      BestEfficiency := eG;
      XSelectBest := XCalc;
      ZSelectBest := ZCalc;
      YSelectBest := YCalc;
      FOR J := 1 TO NuOfRepeat DO
         Begin
            Rn1 := Random;
            IF (Rn1 < 0.5) Then InterchangeOnePair
            ELSE RandPermMElements;
            Linked;
            CalcVarianceGroupedLeastSquares;
            CalcEfficiency;
            IF eG > = BestEfficiency Then
               Begin
                  BestEfficiency := eG;
                  XSelectBest := XCalc;
                  ZSelectBest := ZCalc;
                  YSelectBest := YCalc
               End
            ELSE
               Begin
                  XCalc := XSelectBest;
                  ZCalc := ZSelectBest;
                  YCalc := YSelectBest;
               End;
         End;
   End;
```

```
    '3' :
       Begin
          SelectOfObservations;
          IdentityMatrixAndVector1;
          CalcVarianceLeastSqures;
          RandomPermutation;
          Linked;
          CalcVarianceGroupedLeastSquares;
          CalcEfficiency;
          BestEfficiency := eG;
          AnnealEff := eG;
          XSelectBest := XCalc;
          ZSelectBest := ZCalc;
          YSelectBest := YCalc;
          Temper := T; Tfactor := 0.9;
          FOR J := 1 TO NuOfTemp DO
             Begin
                If Temper < 0.00005 Then TFactor := 0.97;
                FOR J := 1 TO NuOFRepeat DO
                   Begin
                      Rn1 := Random;
                      IF (Rn1 < 0.5) Then InterchangeOnePair
                      ELSE RandPermMElements;
                      Linked;
                      CalcVarianceGroupedLeastSquares;
                      CalcEfficiency;
                      SelectOfObservations;
                      IF NSucc >= NLimit Then GOTO 10;
                   End;
10:             Temper := Temper * TFactor;
             End;
       End;
       XCalc := XSelectBest;
       ZCalc := ZSelectBest;
       YCalc := YSelectBest;
       Linked;
       PrintResult;
End.
```

References

Adichie, J. N. (1967). Estimates of regression parameters based on rank test. Annals of Math. Statist.38, 894–904.

Brown, B. M. & Maritz, J. S. (1982). Distribution–free methods in regression. Austral. J. Statist. 24, 318–331.

Brown, B. M. (1985). Grouping problems in distribution–free regression, Austral. J. Statist. 27(2), 123–134.

Feldman, D. S., Gagnen, J., Hofmann, R., and Simpson, J. (1986). Stat View 512 + . Brain Power Inc.

Hogg, R. V. and Craig, A. T. (1978). Introduction to Mathematical Statistics. New York :Macmillan Publishing Co. Inc.

Kirkpatrick, S., Gelat, C.D. and Vecchi, M. P. (1983). Optimization by simulated annealing. Science, Vol. 220, 671–680.

Maritz, J. S. (1979). On Theil's method in distribution–free regression. Austral. J. Statist. 21, 30–35.

Maritz, J. S. (1981). Distribution–Free Statistical Methods. London : Chapman and Hall.

Mees, A., Sands, D. and Teo, K. L. (1989). Optimal planing of generator installation by methods of annealing type. Research Report, Department Of Mathematics, University of Western Australia.

Metropolis, N., Rosenbluth, A., Teller, A. and Teller, E. (1953). Equation of state calculation by fast computing mahines. J. Chem. Phys. 21, 1087.

Mood, A. M. (1950). Introduction to the theory of statistics. Mc Graw–Hill Company. New York (1950).

Pitman, E. J. G. (1979). Some basic theory in statistics. London : Chapman and Hall.

Sen, P. K. (1968). Estimates of the regression coefficient based on Kendall's tau. J. Amer. Statist. Assoc. 63, 1379–1389.

Theil, H. (1950). A rank method of linear and polynomial regression analysis. I.Proc. Ken. Ned. Akad. Wetensh. A. 53, 386–392.

Wald, A. and Wolfowitz, J. (1944). Statistical tests based on permutations of the observations. Annals of Math. Statist. 15, 358–372.