A LEVEL SET GLOBAL OPTIMIZATION METHOD FOR NONLINEAR ENGINEERING PROBLEMS

By

Hassen Ali Yassien

B. Sc. in Civil Engineering, Addis Ababa University, Ethiopia, 1979

Diploma in Hydrology, Free University of Brussels, Belgium 1984

Masters in Hydrology, Free University of Brussels, Belgium 1985

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF CIVIL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

December 1993

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of Civil Engineering

The University of British Columbia
Vancouver, Canada

Date March 20, 1994

Abstract

The mathematical equations used in civil engineering design procedures are predominantly nonlinear. Most civil engineering design optimization problems would therefore require the use of nonlinear programming (NLP) techniques for their solution. Those NLP packages with the ability to handle practical sizes of problems, and have been available on mainframe computers for many years, are only now becoming available on microcomputers. On top of this, these existing NLP techniques, which are dominated by the gradient methods, do not guarantee global solutions. As a consequence suitable optimization methods for civil engineering design are not being enjoyed by practitioners.

In this thesis, the level set optimization method, whose theory was initially presented in *"Integral global optimization"* by [Chew & Zheng, 1988] was further developed to address, in particular, practical engineering problems. It was found that Level Set Programming (LSP), offers a viable alternative to existing nonlinear optimization methods. While LSP does not radically alter the computational effort involved it has some unique characteristics which appear to be significant from the engineering users point of view.

LSP which is classified as a direct search method of optimization, utilizes the set theory concept of a level set. It uses estimates of moments of the objective function values at the confirmed points within a level set to control the search advance and as a measure of convergence on the global optimum.

The reliability and efficiency of LSP was verified by comparing its results with published results for both mathematical and engineering test problems. In addition to the published test problems, a new parametrically adjustable mathematical test problem was designed to test global optimization methods in general and to explore the strengths and weaknesses of LSP in particular. Experience with these test problems showed that LSP gave similar results to those cited in the literature as well as improved results or more complete sets of global solution.

The large number of solutions developed at each iteration of LSP permits meaningful graphical displays of the progressive reduction in the level set boundaries as the global solution is approached. Other displays were also found to provide insights into the solution process and a basis for diagnosing search difficulties.

Table of Contents

Abstract ii

List of Tables ix

List of Figures xi

Acknowledgement xiv

1 INTRODUCTION 1

1.1 Nonlinear Programming (NLP) 1

1.1.1 Direct Search Methods 2

1.1.2 Gradient based methods 3

1.2 Limitations of existing NLP methods 5

1.3 Global optimization 7

1.4 Optimization needs for the professional 9

1.5 Earlier work on level set optimization 11

1.6 Level Set Programming (LSP) 12

1.7 Thesis outline 14

2 LEVEL SET PROGRAMMING 16

2.1 Basic theory of Level Set Programming (LSP) 16

2.2 Overview of the LSP algorithm - one dimensional case 23

2.3 LSP Implementation 26

2.3.1 Cuboid approximation of level set boundary 27

2.3.2 Alternative methods of sample point generation 30
2.3.3 Revision of level set value-c 32
2.3.4 Detailed description of the LSP Algorithm - general case 36
2.3.5 Alternate termination criteria 38
2.3.6 Handling Constraints in LSP 42
2.4 Summary 45

3 LSP PERFORMANCE IMPROVEMENT 48

3.1 Partition sets and cluster analysis 48
3.1.1 Cluster analysis methods 53
3.1.2 General implementation of cluster analysis methods 56
3.1.3 A clustering routine for LSP 61
3.2 Penalty functions 65
3.2.1 Penalty parameters and LSP 68
3.3 Inequality and equality constraint relaxation 71
3.3.1 Tightening of relaxed constraints 73
3.4 Cuboid stretching 74
3.5 Skewness adjustment 75
3.6 Search domain modification 79

4 TEST PROBLEMS 83

4.1 Coding of mathematical test problems 86
4.2 Mathematical problems 87
4.2.1 The Road Runner function 89
4.3 Engineering test problems 94
4.3.1 Basin design [Problem 3-16-1-3] 96
4.3.2 Pipe network design [Problem 32-1-40-34] 101

4.3.3 10-member truss design [Problem 10-2-0-40] 107

4.3.4 38-member truss design [Problem 38-1-0-77] 112

4.3.5 Unit Hydrograph ordinate determination [Problem 19-1-21-0] . 115

4.3.6 Air pollution control design [Problem 4-11-0-40] 119

4.3.7 Irrigation system design [Problem 6-7-1-4] 123

4.3.8 Alkylation process [Problem 10-1-7-0] 128

4.3.9 Heat exchanger network configuration [Problem 16-1-15-0] . . 132

4.3.10 Power generation using fuel oil [Problem 4-11-0-5] 136

4.3.11 Chemical equilibrium [Problem 10-1-3-0] 139

4.4 Model fitting 142

5 EXPERIENCE WITH LSP 145

5.1 Intermediate output, its presentation, and interpretation 145

5.1.1 Plots of x_i, x_j pairs 146

5.1.2 Plots of cumulative function evaluations-N_f against iteration number-I 147

5.1.3 Plots of level set value-c against the iteration number-I 153

5.1.4 Further generalization of the $N_f \sim I$ and $c \sim I$ plot interpretations 155

5.2 Adjusting LSP parameters 158

5.2.1 Number of confirmed points in the acceptance set-N_{keep} 158

5.2.2 Termination criterion-V_M 161

5.2.3 Cluster criterion 162

5.2.4 Cuboid stretching parameters 163

5.2.5 Skewness adjustment parameters 163

5.2.6 Penalty and constraint relaxation and tightening parameters . . . 164

5.3 Use of a rhombohedron shaped search domain 165

5.4 Relationship between initial cuboid volume and its location, and N_f . . . 167

5.5 Some specific difficulties encountered while implementing LSP 168

5.6 Observations and recommendations 173

6 EVALUATION OF NLP PERFORMANCE 177

6.1 Existing evaluation criteria 178

6.2 Limitations of Schittkowski's performance criteria 181

6.3 Recommended set of evaluation criteria 182

6.4 Evaluation example: LSP versus GRG2 184

7 LSP COMPUTER IMPLEMENTATION 188

7.1 Subroutines 189

7.2 Confirmed point generation strategy 193

7.3 Intermediate results presentation 198

7.4 Final output 199

7.5 Alternative presentation of scatter plots 199

7.5.1 Static plots 200

7.5.2 Dynamic plots - Alternagraphics 202

8 SENSITIVITY ANALYSIS 208

8.1 Sensitivity with LSP 210

8.1.1 Sensitivity interpretation of the confirmed points at convergence plotted in the initial cuboid 211

8.1.2 Response of acceptance set to changes in the level set value-c . . . 213

8.2 Other approaches to obtaining sensitivity information 214

9 CONCLUSION 217

9.1 Introduction 217

9.2 Evaluating LSP performance using test problems 219

9.3 LSP performance improvements 220

9.4 Graphical output 222

9.5 Civil engineering design problems and LSP 223

Bibliography **228**

Appendices **233**

A MATHEMATICAL TEST PROBLEMS **233**

B DIRECT SEARCH METHODS **277**

List of Tables

3.1 Reduced number of function evaluations due to cluster analysis methods 63

4.1 Summary of test problems reported in Appendix A 87

4.2 Optimal results for the basin design problem 101

4.3 Input data for the pipe network 105

4.4 Available pipe diameters and unit prices 105

4.5 Optimal design for the pipe network given in ASCE, 1977 106

4.6 Optimal design for the pipe network found with LSP 106

4.7 Detailed results for the 10-member truss design 111

4.8 Comparison of truss weights for the 10-member truss design 111

4.9 Alternate optimal designs for the 38-member truss 114

4.10 Rainfall-Runoff data for Little Walnut Creek 118

4.11 Optimal hydrograph for the Little Walnut Creek 119

4.12 Stack and Emission data 121

4.13 Optimal design cited in the literature and that found with LSP 123

4.14 Constants for the irrigation system design 126

4.15 Optimal solution for the irrigation system design 127

4.16 Bounds for the variables involved in the alkylation process 131

4.17 Optimal solution for the Alkylation process 132

4.18 Stream data 134

4.19 Match data 134

4.20 Optimal solution to the heat exchanger configuration 136

4.21 Constants in fuel consumption equations 138
4.22 Optimal power generation cited in the literature 139
4.23 Optimal power generation found with LSP 139
4.24 Free energy constants (w_s) 140
4.25 Optimal solution to the Chemical equilibrium problem 141

5.1 Function evaluations using cuboid and rhombohedron 166

6.1 Schittkowski's NLP performance criteria 181
6.2 Recommended NLP performance criteria 185
6.3 Comparison of GRG2 and LSP using proposed criteria set 186

7.1 LSP's problem definition and operations 190
7.2 Comparison of N_f for two different point confirmation strategies 195
7.3 Comparison of N_f for different strategies for a constrained problem . . . 197

A.1 Data for m = 4 and n = 3 250
A.2 Experimental data used to fit the nonlinear regression model 254
A.3 Optimal solution for the regression model 254
A.4 Data for m = 5 and n = 4 259
A.5 Experimental data used to fit the nonlinear regression model 260

List of Figures

2.1 Sequential improvement of level set value-c. 24

2.2 Cuboid derived from limited number of confirmed points. 30

2.3 Cuboid defined by the smallest objective function values. 33

2.4 N_f for cuboids defined by different level set value criteria. 35

3.1 Connected and partition sets. 51

3.2 Level set "volume" in one dimensional problem with and without partitioning. 53

3.3 Single and multiple linkage clustering methods. 58

3.4 Stepwise dendrogram construction. 60

3.5 Cluster analysis approach for a two variable problem. 64

3.6 Modified penalty function. 70

3.7 Relaxation of constraints. 73

3.8 Shift of cuboid with skewness adjustment. 77

3.9 Contrast between cuboid and rhombohedron in two variable space. 80

3.10 Point generation in a rhombohedron search domain. 82

4.1 The LSP algorithm as implemented in this thesis. 85

4.2 The Road Runner function in one dimension to show the influence of a. . 90

4.3 The Road Runner function in two dimensions. 93

4.4 Treatment basin. 97

4.5 Two-looped water supply system. 103

4.6 10-member truss. 108

4.7 Detailed results for Venkayya's 10-member truss problem. 110

4.8 38-member truss. 112

4.9 Rainfall components. 116

4.10 Optimal unit hydrographs for Little Walnut Creek. 120

4.11 Alkylation process. 130

4.12 Two-boiler Turbine-Generator combination. 137

5.1 Local optima suggested by the disappearance of a point cluster. 148

5.2 $N_f \sim I$ plots. 150

5.3 Rastrigin function-Implied local optima by $N_f \sim I$ plot. (Actual screen dump). 152

5.4 Screen display for the Road Runner function, two dimensional case. (Actual screen dump). 153

5.5 Level set value-c versus iteration-I. 155

5.6 Low N_{keep} search for the Road Runner function. Demonstrates discontinuity in $c \sim I$ plot. (Actual screen dump). 156

5.7 Interpretation of simple $c \sim I$ and $N_f \sim I$ curves, unimodal and multimodal objective functions. 159

5.8 Interpretation of more complex $N_f \sim I$ and $c \sim I$ plots. 160

5.9 The crescent shaped feasible region of Subrahmanyam's problem [Problem 2-12-0-4, Appendix A]. 170

5.10 Acceptance set volume change after an iteration while cuboid remains constant. 172

7.1 Deletion of a point in dynamic graphics. 204

7.2 Linking of points for a 4 variable problem [Problem 4-1-0-3]. 205

7.3 Points in the brush highlighted in all the panels [Becker, 1987]. 206

8.1 Plots of confirmed points after convergence criterion is met. 212

8.2 Plots of confirmed points at convergence within the initial cuboid for a 6 variable problem (Actual screen dump) 213

8.3 Plots of confirmed points within the initial cuboid for different c values . 214

8.4 Points in the level set, for $c = c^*$ and $c = 1.01 * c^*$ - [Problem 4-6-1-2] (Actual screen dump) 216

Acknowledgement

I wish to express my most sincere gratitude to Dr. William F. Caselton, my teacher and supervisor, who had a profound and positive impact on my academic and professional attitudes. I greatly appreciate his advice, encouragement, guidance and support throughout my graduate studies. This thesis would not exist without his patient efforts and valuable suggestions. I am deeply indebted for his interest in this work, for his invaluable guidance, for the many hours spent on valuable discussions, for his meticulous proof reading, and for the diligence during the research.

My special thanks go to my supervisory committee members, Dr. Alan D. Russell, Dr. S. O. Denis Russell and Dr. Donald W. Thompson for their valuable suggestions during the course of the research. In spite of their busy schedules, they always found time to review the progress of my work and communicate back with me. They have contributed numerous clarifications of the research. Their efforts in reviewing this thesis are greatly appreciated.

I would also like to acknowledge my family members, who had always given me the moral support and warm passion across the miles. My thanks go to my wife Munira, for the patience and affection she has shown me over the past few years and the understanding she has shown during the travails of my research. Her presence in my life gave added meaning to this whole endeavour.

The financial support of CIDA to cover the expenses of my studies and the role of Associated Engineering International Ltd. in facilitating my stay in Vancouver is also appreciated.

Above all, I praise the Almighty Allah for all my achievements.

Chapter 1

INTRODUCTION

1.1 Nonlinear Programming (NLP)

The equations used by engineers to mathematically describe engineering problems and to describe physical behaviour are predominantly nonlinear. When an explicit optimization objective is involved then a mathematical formulation of the optimization problem will conform to the conventional nonlinear programming problem. The mathematical formulation of such an optimization problem can be written as

$$\begin{aligned} &\text{Minimize or Maximize } f(\mathbf{x}) \\ &\text{Subject to:} \\ &\qquad g_i(\mathbf{x}) \leq 0 \qquad \text{for } i = 1, m \\ &\qquad h_j(\mathbf{x}) = 0 \qquad \text{for } j = 1, l \end{aligned}$$

where $\mathbf{x}$ is an n-tuple vector of independent variables and $f(\mathbf{x})$, $g_1(\mathbf{x})$, $g_2(\mathbf{x}), ..., g_n(\mathbf{x})$, $h_1(\mathbf{x})$, $h_2(\mathbf{x}), ..., h_l(\mathbf{x})$ are nonlinear functions defined on the Euclidian n space.

The equation $f(\mathbf{x})$ is known as the objective function while the other two equation types, $g_i(\mathbf{x}) \leq 0$ and the $h_j(\mathbf{x}) = 0$, are the set of constraints. The space or region defined by these constraints is known as the feasible space or the feasible region.

The non linearity of any of these equations precludes the use of the well established and reliable Linear Programming methods. Approximation of the nonlinear equations to linear or quadratic expressions might facilitate a solution being found but can substantially alter the problem being solved. Finding a satisfactory solution generally necessitates the use of true nonlinear approaches to preserve the characteristics of the problem.

Currently the methods used in nonlinear programming are categorized as direct search or gradient search [Reklaitis et al., 1983]. These two method classes are discussed in the next sections.

1.1.1 Direct Search Methods

Direct search methods range from simple simultaneous point sampling techniques to more elaborate procedures that involve a coupling of sequential sampling with heuristic hill climbing methods [Reklaitis et al., 1983]. None of the direct search methods require the use of derivatives. Some of the most frequently cited direct search methods in the literature are: Blind search; Grid search; Univariate search; Conjugate direction methods; Powell's conjugate direction methods; Simplex method and Complex method.

Many of these methods have elements in common with each other and the level set method described in this thesis is no exception. Descriptions of each of the above listed methods are included in Appendix B of this thesis for reference purposes.

The blind search, which is also known as simultaneous search, is quite inefficient in its use of function evaluations [Leon , 1966 & Reklaitis et al., 1983]. In this strategy, N_f random points are generated and the objective function evaluated at each point. The N_f objective function values are compared and the lowest value is selected as the optimum value.

The size of N_f depends upon the desired probability of successfully finding the global optimum. For a bounded but otherwise unconstrained problem involving n variables and a length of d units on each side of the feasible region, the volume of the feasible region would be d^n. Take a small fraction of the feasible volume $a = (\frac{\delta}{d})^n$, where δ is a small length on each side of the bounded feasible region. The size of a is so small that variation of the objective function value within that region can be tolerated. Then for a purely random search, the probability of a single test to be outside of a would be $1 - a$.

Assume the probability of having at least one point in a is P, then, with N_f points

$$P = 1 - (1 - a)^{N_f}$$
$$N_f = \frac{log(1-P)}{log(1-a)}$$

and for small a,

$$N_f \approx -2.3 \frac{log(1-P)}{a}$$

For $P = 0.90$, which corresponds to a 0.90 confidence of obtaining the global optimum, the computational effort reflected by N_f, would be $2.3(\frac{d}{\delta})^n$ [Reklaitis et al., 1983]. With a value of $\delta = \frac{d}{40}$, a ten variable problem would need about $2.4 * 10^{16}$ function evaluations for 0.90 confidence. This number is prohibitively large from a computational standpoint. The optimal solution can be obtained with probability of 1.0 only as the number of sample points approaches infinity. Because of the high number of function evaluations involved, a blind search is not recommended for solving problems of even moderate size.

1.1.2 Gradient based methods

Gradient based methods make use of derivatives to determine the search direction for optimization. One of the common gradient based methods, the steepest descent method, uses only the first derivatives of the objective function for unconstrained problems [Edgar & Himmelblau, 1988]. The gradient vector at a point gives the direction of the greatest decrease in $f(\mathbf{x})$. For a minimization problem the search direction is specified by the negative of the gradient, i.e.

$$S^k = - \bigtriangledown f(\mathbf{x}^k).$$

S^k is the search direction at point k.

$\bigtriangledown f(\mathbf{x}^k)$ is the derivative of $f(\mathbf{x})$ at $\mathbf{x}^k$.

At any stage of minimization the transition from one point to the next is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k = \mathbf{x}^k + \lambda^k S^k = \mathbf{x}^k - \lambda^k \bigtriangledown f(\mathbf{x}^k).$$

$\Delta\mathbf{x}^k$ is the vector of increments from $\mathbf{x}^k$ to $\mathbf{x}^{k+1}$

λ^k is a scalar quantity that determines the step length in direction S^k and it can be either a predefined fixed value or its value can be optimized at each iteration to improve the speed of the convergence on the optimum point. In some cases it is possible to simplify a nonlinear function by making use of the quadratic approximation, at any point $\mathbf{x}^k$,

$$f(\mathbf{x}) \approx f(\mathbf{x}^k) + \bigtriangledown^T f(\mathbf{x}^k)\Delta\mathbf{x}^k + .5(\Delta\mathbf{x}^k)^T H(\mathbf{x}^k)\Delta\mathbf{x}^k$$

where $H(\mathbf{x}^k)$ is the Hessian matrix of $f(\mathbf{x})$. Some methods then use this second-order approximation of the objective function and the information obtained from the second partial derivatives of $f(\mathbf{x})$ with respect to the independent variables.

The quadratic approximation of $f(\mathbf{x})$ at $\mathbf{x}^k$ is differentiated with respect to each independent variable and the resulting expression equated to zero to define a stationary point.

$$\bigtriangledown f(\mathbf{x}) = \bigtriangledown f(\mathbf{x}^k) + H(\mathbf{x}^k)\Delta\mathbf{x}^k = 0 \tag{1.1}$$

The next search point, at point $k+1$, is obtained from this expression as

$$\mathbf{x}^{k+1} - \mathbf{x}^k = \Delta\mathbf{x}^k = -[H(\mathbf{x}^k)]^{-1} \bigtriangledown f(\mathbf{x}^k) \text{ or}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k - [H(\mathbf{x}^k)]^{-1} \bigtriangledown f(\mathbf{x}^k)$$

where $[H(\mathbf{x}^k)]^{-1}$ is the inverse of the Hessian matrix $H(\mathbf{x}^k)$

Both the step length and direction of search are determined from the same expression, Equation 1.1. Only one step would be necessary to reach the optimum if $f(\mathbf{x})$ was a true quadratic equation. But, since in general this is not the case, there would be multiple steps.

1.2 Limitations of existing NLP methods

For simple two variable problems, elementary direct search methods are often quite satisfactory, but they are neither efficient nor reliable for the higher dimensionality problems which typically arise in engineering [Edgar & Himmelblau, 1988].

Gradient procedures that use second order information are superior to those that use first order information only. But the danger is that usually the second order information may be only approximate as it is based, not on second derivatives, but approximates to second derivatives, with the consequence that it is no longer the original problem that is being solved [Edgar & Himmelblau, 1988].

All gradient procedures start their search from a point and follow a single path until the convergence criterion is met. The point at which the convergence criterion is met is regarded as the optimum point. If a different starting point is used to perform the search again, it is possible that it can lead to a different optimum point. This reveals that solutions found with gradient based optimization methods are often local optima. For a solution point $\mathbf{x}^*$ to be a local optimum (local minimum in this case) the objective function value at that point should be the least of all the objective function values in the neighbourhood. That is

$$f(\mathbf{x}^*) \leq f(\mathbf{x})$$

where $\| \mathbf{x} - \mathbf{x}^* \| \leq \zeta$, and ζ is a small value.

Unfortunately, except in certain mathematically simple cases, there is no mechanism which can confirm the global nature of the optimum point obtained with gradient methods. The global minimum is defined as the smallest of all the local minima in the region of interest. In some particular cases, the global minimum can appear at multiple points.

In order to apply gradient methods, the functions have to be continuous and differentiable so that their derivatives can be evaluated. But in engineering design we are often confronted with problems which involve un-differentiable functions, discontinuous functions and/or discrete variables. In such cases the use of gradient based methods is limited.

Nonlinear optimization problems may have more than one global optimum solution. Furthermore there may be other solutions which, while not globally optimal, produce values of the objective function close to the global optimum value and are of considerable practical interest. Unfortunately almost all existing NLP algorithms are incapable of identifying the full set of optimal points and provide little or no indication of near optimal solutions.

Information obtained as a byproduct of gradient searches, primarily in the form of Lagrange multipliers, Hessian and Jacobean matrices, does provide information but only in the immediate vicinity of the optimal solution. But this information can also be difficult to interpret, especially in the context of the original engineering problem and when the global nature of the solution is in question. It provides no clues concerning multiple global optima or near optimal solutions. In the event of an NLP method failing to confirm even a local optimum, the user is left with no suggestion for his next move.

Much of the information we get from the intermediate stages of gradient based NLP searches does not have pertinence to the engineering problem being solved. It simply indicates the outcome of some mathematical manipulations which are incidental to the real world problem and therefore have no real relevance to an engineering practitioner. Most of the existing NLP methods will perform well on certain specific types of problems but these are rarely representative of the more general nonlinear optimization problems faced by engineers. Thus, in general, existing gradient based or direct search NLP methods have limitations in addressing real world problems.

Other optimization techniques, which represent departures from the two classes which have been described above, have also emerged in the last decade. Genetic Optimization for example, which "mutates" binary strings, has already appeared in the engineering literature [Goldberg, 1989], while Random Tunnelling and Simulated Annealing have been proposed in the last few years. These methods have some limitation in their application, for instance Random Tunnelling converges rapidly for problems involving only one variable but good results are not expected for higher dimensional problems [Kan and Timmer, 1989], and Simulated Annealing is reported to be inefficient [Torn & Zilinskas, 1988], [Corana et al., 1987] and [Kan & Timmer, 1989].

1.3 Global optimization

When dealing with nonlinear optimization problems, it is possible that the problem possesses many local optima. Because of the highly nonlinear nature of the equations often involved this phenomenon is quite common with engineering system designs [Luus & Jaakola, 1973]. For general engineering optimization purposes a method is required that can identify the overall optimal solution among the many alternative local solutions. Such a method is known as a global optimizer. Its aim is to identify the smallest of all the minima in the region of interest and may not necessarily evaluate all of the local minima.

A sufficient gradient related criterion to positively confirm that a global optimum value has been achieved at a point does not exist. The only way to confirm that the point is a global solution is to evaluate the objective function at that point and compare it with objective function values at all other points.

In any optimization method the search for the optimum solution is performed by evaluating the objective function at some trial points within the search domain. A number of points, obviously more than one, have to be used to reach the final solution. The distribution of these trial points within the search domain is influenced by two distinct optimization goals, these are the global reliability goal and the local refinement goal [Torn & Zilinskas, 1988].

The global reliability goal is based on the assumption that the global optimum could be located at a point anywhere in the search domain and therefore assumes that no part of the search domain can be neglected if the global optimum is sought. Without any refinement in strategy it leads to search procedures where the trial points are distributed uniformly over the entire search domain. On the other hand, the local refinement goal is based on the assumption that the probability of detecting a point with improved solution is higher in the neighbourhood of a point with a relatively low objective function value than in the neighbourhood of a point with a relatively high objective function value. Again, without refinement, this goal has a natural tendency to generate sequences of points with decreasing function values that converge to local minima.

Most global optimization methods focus on exploring the whole of the search domain, but use local search strategies to numerically refine solutions. The various global optimization methods differ on how the search strategy shifts from one goal to the other.

The performance evaluation of local and global minimization methods differ in principle. Local methods are evaluated on the basis of how often and how efficiently they converge on a confirmable optimal point of any kind. But a global method is evaluated on two distinct qualities, the first one addresses the same issues as for local optimization methods but a second one, called reliability, measures the capacity to identify the true global optimum [Wang & Luus, 1978]. Clearly, because of these dual requirements and the complex nonlinear problems addressed, global techniques must be tested more comprehensively than gradient methods to prove their effectiveness.

Rather than strictly following mathematically formulated procedures, global optimization methods are more inclined to adopt some heuristic approaches than are gradient methods. For example, if the global optimum is difficult to find, it may be attractive to deal with an approximation of the objective function which yields a computational enumeration advantage over the original objective function. A second example of an heuristic approach is subdividing the search domain to smaller regions at some stage of the search and then performing the search in each of the subregions. Justification for introducing heuristics into optimization methods are discussed in [Zanakis & Evens, 1981] where they state the following.

> "The need for good heuristics in both academia and business will continue increasing fast. When confronted with real world problems, a researcher in academia experiences at least once the painful disappointment of seeing his product, a theoretically sound and mathematically 'respectable' procedure, not used by its ultimate user. This has encouraged researchers to develop new improved heuristics and rigorously evaluate their performance, thus spreading further their usage in practice, where heuristics have been advocated for a long time."

1.4 Optimization needs for the professional

In addition to the limitations of the existing methods in dealing with nonlinear optimization problems, mentioned in section 1.2, many of them have until recently been confined to implementation on mainframe computers. The fact that access to mainframe computers has been severely limited to most civil engineers has deterred practitioners from using the existing NLP optimization methods. But the substantial improvement in access to computers in recent years, through the introduction of personal computers and their rapid development, has not yet lead to a corresponding growth in the use of optimization methods in civil engineering practice.

As stated previously, the mathematical equations used in civil engineering design procedures are predominantly nonlinear, therefore most civil engineering design optimization problems require the use of nonlinear programming techniques for their solution. Those NLP methods with the ability to handle practical sizes of problems that have been available on mainframe computers for many years are only just beginning to become available on microcomputers. But there are other problems still facing the practitioner which will not be solved simply by improved access.

None of the nonlinear optimization methods which have been widely implemented to date offer any absolute assurance of finding the global optimum except under very ideal conditions. Generally they rely on the user to initiate the search from a number of different starting points so that the chances of stumbling onto the global optimum are enhanced. No systematic way of selecting starting points to ensure finding the global optimum is available and the responsibility for the overall global search strategy and its success is left entirely to the user.

Perhaps, most seriously of all, a practising civil engineer is not generally trained or skilled in numerical analysis. He therefore faces a considerable investment in time and money to confidently embark on nonlinear optimization to solve a real world problem. If he makes this investment then he still faces the shortcomings of existing methods described in the two previous paragraphs. Although not an uncommon topic in the civil engineering research literature, there has been evidence of only rare and limited use of optimization methods in civil engineering practice. Clearly existing nonlinear optimization methods have had insufficient appeal to convince civil engineers that their time should be invested in that direction. This prompted the research, reported in this thesis, into a methodology based on the exploration of level sets, that has received very little attention. Amongst many of the other positive attributes explained subsequently in this thesis, it has the potential of being a far more appealing nonlinear optimization methodology to the person who is not a specialist in numerical analysis.

1.5 Earlier work on level set optimization

The theory discussed in [Chew & Zheng, 1988] under "*Integral global optimization*" opens a new alternative to global optimization based on level set theory. But some other authors have expressed opinions that the level set method is not designed for solving practical problems [Torn & Zilinskas, 1988]. This is because, from the practitioner's point of view, there are certain issues, important in engineering problems, which Chew and Zheng did not address. As described below, these issues deal with superiority of the method, computational effort, multiple optima and significance of intermediate outputs.

Even though the authors hint that there had been improvements made in their work, the improvements are neither enumerated nor documented. This remains a hinderance in identifying their contribution to optimization methodology and assessing the superiority of the level set method over other existing methods.

A general statement was given stating that less computational effort is expended with the level set method compared to the pure random search method. Apart from this statement there was no systematic comparison between the level set method and the predominant (gradient) methods. Lack of comparative performance tests with respect to efficiency, reliability or global convergence provided no incentive for others to adopt level set optimization.

The authors admitted that only problems which have single global optimum had been fully addressed and made only a minor comment on the multiple global optima case. There was no discussion of recognizing near global optimum solutions. But the identification of multiple optima and near optimal solutions are some of the important benefits to practitioners which are identified and investigated in this thesis.

The significance of the level set method output values was also not recognized in [Chew & Zheng, 1988] and consequently the benefits of a graphical interface was overlooked.

1.6 Level Set Programming (LSP)

The formal mathematical presentation of the methodology upon which LSP is based has been presented under the title "*Integral Global Optimization*" [Chew & Zheng, 1988]. They claim equal or better performance when compared with some of the best "conventional" gradient search NLP methods. Their experience in solving a series of NLP problems supported this finding.

Level Set Programming (LSP) is classified as a direct search method of optimization. Like any other direct search optimization method, LSP avoids gradient evaluations and relies solely on evaluations of the objective and constraint functions at solution points. In comparison with most direct search methods it carries a much larger number of solutions at any one time and these solutions are dispersed over larger regions of the design space.

LSP utilizes the simple set theory concept of a level set which will be defined in Chapter 2. It uses estimates of statistical moments of a level set to assess the level set properties, to guide the search algorithm, as well as to measure convergence on a solution optimum. LSP adopts a global strategy where, at least in theory, the whole of the search domain is explored in the search for the global minimum.

The most significant features of LSP from a conventional engineering standpoint are: LSP is a global search method; the search does not follow a single path or is not influenced by a single point; it covers the whole search domain; its reliability to identify the global solution is high; the method is conceptually simple; all the computations and numerical results generated during the search are meaningful; and an elementary, and therefore fast, graphical interface can display much of the useful information at any stage of the search. Based on our experience to date, its computational burden is in the same order as gradient based search methods while its assurance of a successful global search is substantially better.

The name Level Set Programming is adopted to emphasize the central level set feature. The word programming, as appended here, appears to be justified in that the implementation of the methodology has an algorithmic structure with a rigorous theoretical criterion for convergence on the global optimum.

Although the theory of level set optimization already existed prior to this research, its full implementation and testing had never been documented. In this research, a comprehensive level set optimization algorithm named Level Set Programming (or LSP) has been developed.

Refinements to the level set approach, which helped increase the chances of identifying the global solution, increased computational efficiency and enabled the method to handle a wider range of problems were investigated and implemented. These refinements included the use of cluster analysis, penalty functions and constraint relaxation. It was also demonstrated that conventional engineering routines, which are necessary for evaluating the objective function or constraints (such as structural analysis methods and Hardy Cross pipe network analysis method), could be readily embedded into the optimization method. A set of control parameters were also introduced into the algorithm and their optimal values investigated. Furthermore, it was found that these parameters could be adjusted during the course of the search to increase the reliability and efficiency of the search.

Graphical displays of intermediate and final search results were also developed during the research and these have opened up a new perspective on the process of optimization. The human-machine interaction during optimization was also significantly improved with these graphical displays which guide the user towards search parameter adjustments for increased efficiency of search as well as facilitate a better understanding of the optimization process as it unfolds.

The strength of LSP has been demonstrated by solving a wide range of mathematical problems (about 200) and many difficult and realistic engineering problems (about 20) which have been presented in the literature as being challenging for optimization methods. In addition, a parametrically adjustable mathematical test problem was developed during the course of the research to further strengthen confidence in LSP's global optimization capabilities and to challenge global optimization methods in general.

1.7 Thesis outline

The theory behind level set programming and its implementation is explained in Chapter 2. Refinements in the LSP scheme and the introduction of some heuristic methods are then explained in Chapter 3.

Chapter 4 explains the general strategy and performances of LSP in solving mathematical and engineering problems. The engineering test problems formulation and their solutions are presented in the same chapter. The mathematical problems formulation and solutions are documented in Appendix A.

The experience gained in developing LSP and in solving test problems are explained in Chapter 5. This chapter explains the interpretation of intermediate results, discusses the use and adjustments of LSP parameters, the relationship between the volume of search domain and computational overhead, and finally identifies the principal difficulties which might be encountered while implementing LSP.

The established approaches for evaluating NLP methods and some new proposed criteria are given in Chapter 6. The most important factors in the field of optimization from an engineering practitioners point of view are also enumerated in the same chapter and contrasted with the established approaches to evaluating NLP methods.

Chapter 7 explains the operational use of LSP. The subroutines in the computer program developed and its input/output procedures are discussed.

Chapter 8 discusses sensitivity analysis with LSP. A contrast on the definition of sensitivity analysis between the classical methods and in the LSP context are presented. Interpretations of sensitivity information from LSP plots are also included. Finally Chapter 9 gives the general conclusions concerning LSP's effectiveness and suitability as an engineering tool.

Chapter 2

LEVEL SET PROGRAMMING

This chapter outlines the basic theory of LSP and then considers extensions and variations of the algorithm which were investigated in the course of this research.

2.1 Basic theory of Level Set Programming (LSP)

This section summarizes the essential theory for the implementation of LSP as described in this thesis. More extensive theory is provided in [Chew & Zheng, 1988] and includes details such as the properties of higher moments (third, forth, etc.) of the level set, etc., which do not appear to have any bearing on engineering implementation, and are therefore omitted here.

The general formulation of an optimization problem is

$$\begin{aligned}&\text{Minimize } f(\mathbf{x})\\&\text{Subject to}\\&\qquad g_i(\mathbf{x}) \leq 0 \qquad \text{for } i = 1, 2, ..., m\\&\qquad h_j(\mathbf{x}) = 0 \qquad \text{for } j = 1, 2, ..., l\\[1em]&\qquad \mathbf{x} \in X\end{aligned} \tag{2.1}$$

where

$f(\mathbf{x})$ is a single valued objective function

$g_1(\mathbf{x}), ..., g_m(\mathbf{x})$ are inequality constraints

$h_1(\mathbf{x}), ..., h_l(\mathbf{x})$ are equality constraints

X is a subset of E^n, and

$\mathbf{x}$ is a vector of n components $x_1, ..., x_n$.

and all functions are defined on Euclidian n space, E^n.

The optimization problem given in Equation 2.1 is a nonlinear programming problem when any or all of the constraints or the objective function are nonlinear. In the applications of interest here, $\mathbf{x}$ corresponds to a set of design or decision variables.

For an unconstrained but bounded minimization problem the goal is to identify $\mathbf{x}^*$, which is called the global optimum point, or a set of global optimal points, and c^*, which denotes the global optimum value of $f(\mathbf{x}^*)$, where the relation $f(\mathbf{x}) \geq f(\mathbf{x}^*) = c^*$ would be true for all feasible points $\mathbf{x}$. Thus the minimization problem is to find the minimum of $f(\mathbf{x})$ over S_b, where S_b is the region defined by the upper and lower bounds of each variable x_i, so that

$$c^* = \min\{f(\mathbf{x}) \mid \mathbf{x} \in S_b\} \tag{2.2}$$

A **level set**, which is at the heart of the optimization methodology in this thesis, is defined as the set of points which provide an objective function value below or equal to some specified value. If this specified value is represented by c, then the associated level set H_c can be defined by

$$H_c = \{\mathbf{x} \mid f(\mathbf{x}) \leq c\}. \tag{2.3}$$

Bounds on each component of $\mathbf{x}$ are implicit but constraints are not considered in this definition.

Certain first and second moments of the objective function values of the points contained in the level set play important roles in LSP and are defined below. The following expressions are for continuous integrable functions on S_b only adopted from [Chew & Zheng, 1988].

Mean value over the level set H_c

$$M(f,c) = \frac{1}{\mu(H_c)} \int_{H_c} f(\mathbf{x}) d\mu \tag{2.4}$$

and for $c \geq c^*$

$$M(f,c) \leq c$$

while for constants $c_2 \geq c_1 \geq c^*$

$$M(f,c_2) \geq M(f,c_1)$$

where Chew & Zheng defined μ as a measure of the level set and, c_1 and c_2 represent any two constants greater than or equal to the global optimum value.

Variance over the level set H_c

Similarly Chew and Zheng specified the variance and a modified variance over the level set H_c as

$$V(f,c) = \frac{1}{\mu(H_c)} \int_{H_c} [f(\mathbf{x}) - M(f,c)]^2 d\mu$$

$$V_M(f,c) = \frac{1}{\mu(H_c)} \int_{H_c} [f(\mathbf{x}) - c]^2 d\mu \tag{2.5}$$

The modified variance is slightly easier to compute and has the same general properties as the variance.

For the discrete sampling case Equations 2.4 and 2.5 can be written in the form of summations to provide estimates of $M(f,c)$ and $V_M(f,c)$

$$M(f,c) = \frac{1}{\sum_{\mathbf{x}^j \in H_c} a^j} \sum_{\mathbf{x}^j \in H_c} a^j f(\mathbf{x}^j)$$

$$V(f,c;S) = \frac{1}{\sum_{\mathbf{x}^j \in H_c} a^j} \sum_{\mathbf{x}^j \in H_c} a^j [f(\mathbf{x}^j) - M(f,c)]^2$$

$$V_M(f,c;S) = \frac{1}{\sum_{\mathbf{x}^j \in H_c} a^j} \sum_{\mathbf{x}^j \in H_c} a^j [f(\mathbf{x}^j) - c]^2 \tag{2.6}$$

In practice, there would be only a limited number of points in a level set, therefore, Equations 2.4 and 2.5 are rewritten to reflect this as follows.

$$M(f,c) = \frac{1}{N_{keep}} \sum_{j=1}^{N_{keep}} f(\mathbf{x}^j) \tag{2.7}$$

$$V(f,c) = \frac{1}{N_{keep}} \sum_{j=1}^{N_{keep}} [f(\mathbf{x}^j) - M(f,c)]^2$$

$$V_M(f,c) = \frac{1}{N_{keep}} \sum_{j=1}^{N_{keep}} [f(\mathbf{x}^j) - c]^2 \tag{2.8}$$

where N_{keep}, the number of points at which $f(\mathbf{x})$ is calculated for a discrete sampling scheme, is the number of points in the level set, and plays the role of the measure $\mu(H_c)$ in Equation 2.4. The equivalent moments for the integer variable or discontinuous function cases are computed by evaluating the objective function at discrete points.

Convergence on global optimum

Take a large real number c_0 such that the level set defined by this number over the function $f(\mathbf{x})$ is non empty, that is

$$H_{c_0} = \{\mathbf{x} \mid f(\mathbf{x}) \le c_0\} \neq \emptyset.$$

Define a decreasing series c_k

$$c_{k+1} = M(f, c_k)$$

so that

$$c_0 \ge c_1 \ge \ldots \ge c_k \ge c_{k+1} \ge \ldots \ge c^*.$$

where c^* is the global minimum value of $f(\mathbf{x})$ Now define a decreasing sequence of level sets H_{c_k} as

$$H_{c_k} = \{\mathbf{x} \mid f(\mathbf{x}) \le c_k\}$$

then

$$H_{c_1} \supseteq \ldots \supseteq H_{c_k} \supseteq H_{c_{k+1}} \supseteq \ldots \supseteq H_{c^*}. \tag{2.9}$$

where H_{c^*} is the set of global minima of $f(\mathbf{x})$.

Both c_k and H_{c_k} are decreasing and are bounded below. The lower bounds can be defined by

$$c^* = \lim_{k\to\infty} c_k \tag{2.10}$$

$$H_{c^*} = \lim_{k\to\infty} H_{c_k} = \bigcap_{k=1}^{\infty} H_{c_k} \tag{2.11}$$

Equations 2.9, 2.10 and 2.11 are important properties of level set values and level sets and are critical to LSP's optimization capabilities. At a global optimal point $\mathbf{x}^*$, where

$c^* = f(\mathbf{x}^*)$ is the corresponding global minimum value, the following are also important properties.

For $c \geq c^*$

$$\begin{aligned}
&M(f,c) \geq c^* \\
&M(f,c^*) = c^* \\
&V(f,c) \geq 0 \\
&V_M(f,c) \geq 0 \\
&V(f,c^*) = 0 \\
&V_M(f,c^*) = 0
\end{aligned}$$

The fact that $M(f,c)$, $V(f,c)$ and $V_M(f,c)$ are always single valued quantities, regardless of the dimensionality of $\mathbf{x}$, is exploited in the LSP algorithm. The first moment is used to redefine a new level set value-c and advance the LSP search. The properties of $V(f,c)$ and $V_M(f,c)$ provide a theoretical global convergence criterion. $V(f,c)$ and $V_M(f,c)$ are always non negative and approach zero from the positive direction as the value of c approaches c^*. Either $V(f,c)$ or $V_M(f,c)$ can be used for search termination, but in this thesis, $V_M(f,c)$ was used throughout the test problems.

A closed region in the decision domain is defined by the set of constraints, which includes the variable bounds, and is specified here as the feasible region S. In a constrained minimization problem the goal is to identify $\mathbf{x}^*$ and c^*, where the relation $f(\mathbf{x}) \geq f(\mathbf{x}^*) = c^*$ would be true for all feasible points $\mathbf{x}$. Thus the minimization problem is to find the infimum of $f(\mathbf{x})$ over S so that

$$c^* = \inf_{\mathbf{x} \in S} f(\mathbf{x}). \tag{2.12}$$

Assuming that there is a real number c such that the intersection of the level set and S is non empty, that is $H_c \bigcap S \neq \emptyset$, where $\emptyset$ signifies the null set, then the optimal solution for a constrained problem can be restated as

$$c^* = \min_{\mathbf{x} \in H_c \bigcap S} f(\mathbf{x}). \tag{2.13}$$

For any H_c, the level set H_{c^*} associated with c^* is defined by

$$H_{c^*} = \{\mathbf{x} \mid f(\mathbf{x}) = c^*; \mathbf{x} \in S\} \tag{2.14}$$

which constitutes the set of global optima. Restated in terms of level sets, the global optimization problem is to find c^* and H_{c^*}. In any point sampling scheme only a limited number of points can be realized.

The existence of linear or nonlinear constraints can be accommodated by defining a set of points which fulfil the condition of being both $f(\mathbf{x}) \leq c$ and $\mathbf{x}$ feasible.

Feasible points which fulfil the $f(\mathbf{x}) \leq c$ condition are called **acceptable points** in this thesis. The **acceptance set**, which consists only of acceptable points is defined as

$$\overline{H_c} = H_c \bigcap S = \{\mathbf{x} \mid f(\mathbf{x}) \leq c;\ \mathbf{x} \epsilon S\}. \tag{2.15}$$

Equation 2.11 can be extended to accommodate constrained problems. The lower bound of the acceptance set at convergence on the global optimum c^* is

$$H_c \bigcap S = \{\mathbf{x} \mid f(\mathbf{x}) \leq c; \mathbf{x} \in S\}$$

$$H_c \bigcap S = \lim_{c_k \to c} (H_{c_k} \bigcap S)$$

$$H_{c^*} = \lim_{c_k \to c^*} (H_{c_k} \bigcap S) \tag{2.16}$$

2.2 Overview of the LSP algorithm - one dimensional case

Figure 2.1 conveys the essential idea behind LSP. It shows a nonlinear function $f(x)$ which is to be minimized within specified bounds x^L and x^U of the single decision variable x. The problem is otherwise unconstrained.

If x is a continuous variable then there will be an infinite number of elements in the level set within the lower and upper bounds. It will simplify the exposition of the remaining theory, and more accurately reflect its implementation, if we think of level sets only in terms of samples at a set of discrete values of x or, equivalently, at a set of points in the decision space. This is not to suggest that x is necessarily a discrete variable, although discrete variables are not excluded in this formulation.

To determine an initial level set, a set of feasible points is first established and the objective function evaluated at each point. The initial level set value-c is then set equal to the highest objective function value, that is

$$c = Max\{f(x^j)\}$$

Those points which yield $f(x)$ values either equal to or less than c, and do not violate the bounds, then constitute the level set H_c for level c, that is $H_c = \{x \mid f(x) \leq c\}$. Figure 2.1(a) shows the randomly generated points within the initial bounds and the size of the level set H_c for the corresponding level set value-c.

Estimates of the mean and variance of the $f(x)$ values corresponding to the x values in the level set H_c are established using Equations 2.4 and 2.5 as

$$M(f, c) = \frac{1}{N_{keep}} \sum_{H_c, j} f(x^j)$$

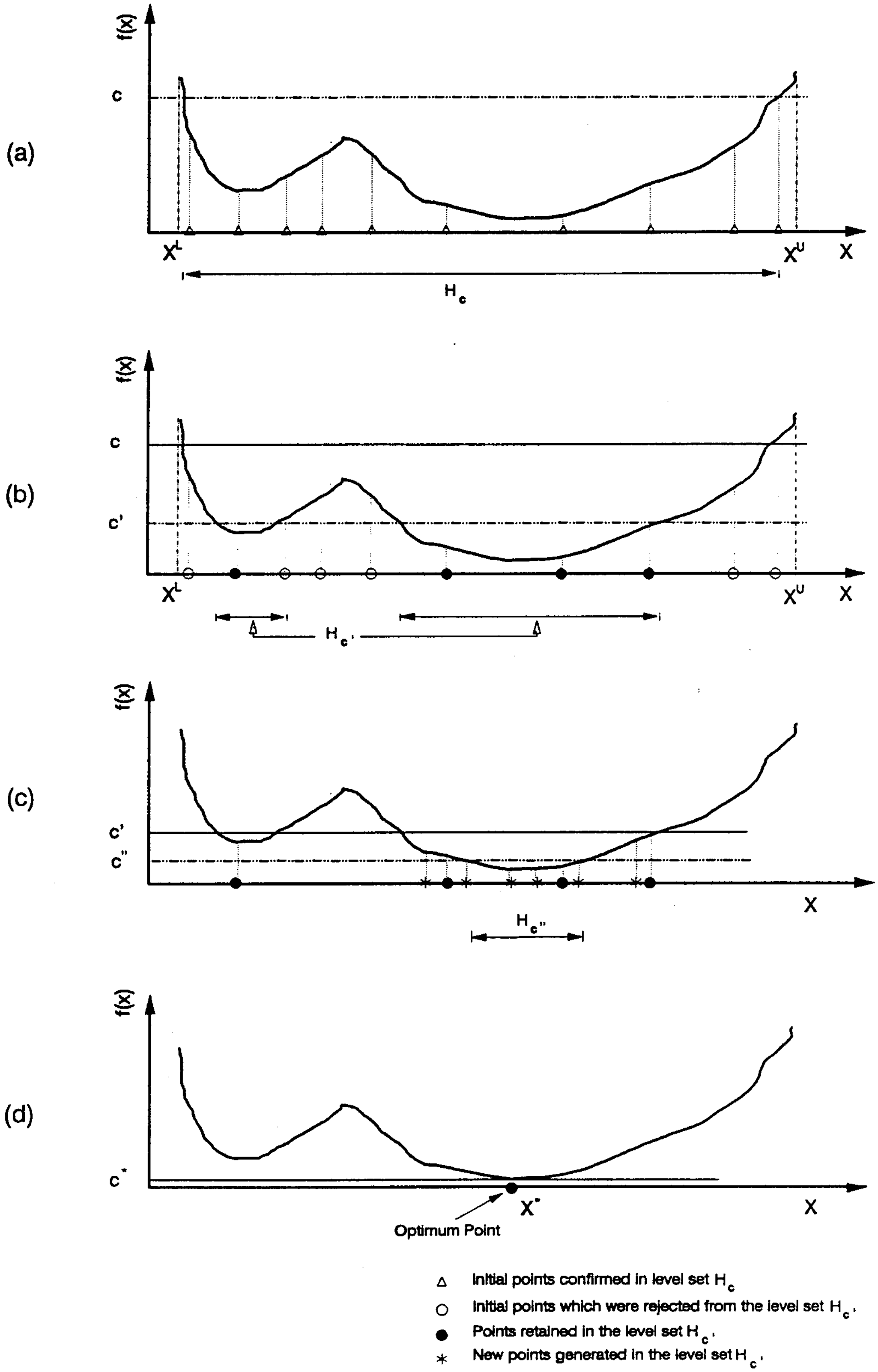

(a)
f(x)
c
X^L
X^U
X
H_c
(b)
f(x)
c
c'
X^L
X^U
X
H_c'
(c)
f(x)
c'
c"
X
H_c"
(d)
f(x)
c*
X*
X
Optimum Point
△ Initial points confirmed in level set H_c
○ Initial points which were rejected from the level set H_c'
● Points retained in the level set H_c'
∗ New points generated in the level set H_c'

Figure 2.1: Sequential improvement of level set value-c.

$$V(f,c) = \frac{1}{N_{keep}}[\sum_{H_c,j}(f(x^j) - M(f,c))^2].$$

$$V_M(f,c) = \frac{1}{N_{keep}}[\sum_{H_c,j}(f(x^j) - c)^2].$$

The search or algorithmic aspect of LSP is driven by using $M(f,c)$ to provide a revised and lower value for c, say c', at a successive iteration by simply setting $c' = M(f,c)$. The level set value c' is then used to establish a new "improved" level set $H_{c'}$. Once the new level set c' is identified, new points are randomly generated in the improved level set $H_{c'}$ to bring the total number of points in the level set to N_{keep}, see Figure 2.1(b). It is important to note that the level set $H_{c'}$ will always be a subset of the level set H_c (see Equation 2.9) although these subsets may not necessarily be connected (see definition in Chapter 3, section 3.1). Once $H_{c'}$ is established then the value of $M(f,c')$ can be obtained. The level set value can again be revised to $c'' = M(f,c')$ and a new level set $H_{c''}$ established.

This procedure of estimating $M(f,c)$ for the current level set, revising the level set value-c, and establishing the new corresponding level set, is then repeated with obvious progressive reduction in the level set value-c.

Finally,the search terminates at the global minimum point x^* with objective function value $f(x) = c^*$ as shown in Figure 2.1(c). The value of $V(f,c)$ acts as an indicator of convergence on the global optimum value c^*. Most significantly, the convergence of $V(f,c)$ on zero at the global optimum is not affected by the existence of local optima or by the existence of a number of separate global optimal solutions. The optimal level set H_{c^*} can include any number of points some distance apart but, as $V(f,c)$ measures dispersion only in $f(x)$, then the value of $V(f, c^*)$ would still be zero. In the special case where the problem has a single global optimum, H_{c^*} reduces to a single point.

2.3 LSP Implementation

In an application of LSP the predominant computational load is in evaluating the objective function and testing the constraints for feasibility at the many trial solution points which are prospective members of the sequence of level sets. This imposes a practical limit on the number of points which can be confirmed to lie within any one level set. At the same time it is essential that this number be large enough to ensure that the sampling is dense enough to reduce, to an acceptably low value, the risk of overlooking a global solution. More precisely, we would like the sample to be dense enough to avoid overlooking important features in the level set boundaries which are being inferred from the level set sample. The algorithmic scheme in LSP exploits the fact that a substantial number of the N_{keep} sample $\mathbf{x}$ points present in H_c will also be present in the level set $H_{c'}$ at the next iteration. In practice, using the mean value criterion for the revised level set value as indicated in section 2.2, the number of these "surviving" points has been found to be close to $\frac{N_{keep}}{2}$. The set of surviving points is used to provide a cuboidal approximation of the new level set boundary (see "Cuboid approximation" below). This approximate boundary is then used to support the efficient regeneration of acceptable points in the new level set. To infer the new level set boundary approximation with acceptable accuracy, and to minimize the risk of omitting any global solution, it is necessary for the number of points in the level set sample to be restored to N_{keep} prior to each revision in the level set value-c.

In the next subsections the important features of LSP implementation are discussed in detail. In the first subsection, approximation of the search domain at every iteration is addressed. Then, alternate methods of sample point generation are explored to identify a more reliable and efficient search. Reliability and efficiency of the LSP search is discussed in section 2.3.3. Initialization of the LSP search and a detailed description of the algorithm are presented in subsection 2.3.4. While $V(f,c)$ is the theoretical criterion for convergence, other search termination criteria are reviewed in section 2.3.5. Finally subsection 2.3.6 explains how the implementation of LSP is extended from the unconstrained case to the general constrained problem. This subsection describes a variety of techniques for handling constraints in the LSP search.

2.3.1 Cuboid approximation of level set boundary

A **cuboid**, which is also known as a hypercube or hyper-rectangle, is defined here as a rectangular parallelepiped in the decision domain. The sides of a cuboid are always parallel to the variable axes. It is defined by

$$D = \{\mathbf{x} \mid b_i^L \leq x_i \leq b_i^U,\, i = 1, 2, ..., n\}$$

where b_i^L and b_i^U are the lower and upper cuboid bounds respectively for each variable i.

The points in a current level set can be used to infer the boundaries of a minimal cuboid which envelops the level set. This is computationally far more economical than attempting to generate more accurate but nonlinear estimates of a minimal envelope of the level set. It should, at the same time, be recognized that more precise boundary estimates using say ellipsoids might provide greater computational efficiency when generating new level set points. The simplicity of the cuboid approach is appealing however and the use of nonlinear envelopes has not been explored in this work.

A general n dimensional optimization problem will be assumed so that the variable vector will be $\mathbf{x} = (x_1, x_2, \dots, x_n)$. To initiate the search, N_{keep} points are first generated within the feasible region. The highest objective function value defines the initial level set value-c and a corresponding level set H_c.

The first general iterative cycle begins by reassigning the level set value to $c' = M(f, c)$ (where the prime symbol denotes the next iteration). Points whose objective function value is greater than c' are discarded. Only n_x points (where in practice $n_x \simeq N_{keep}/2$) would then remain in the new level set $H_{c'}$. The minimum and maximum values for the i^{th} variable (i.e. i^{th} dimension) in the new level set, estimated from the n_x surviving points, would be $b_i^{'L} = Min\{x_i^1, x_i^2, \dots, x_i^{n_x}\}$ and $b_i^{'U} = Max\{x_i^1, x_i^2, \dots, x_i^{n_x}\}$ respectively. Thus the vector of lower bounds for all variables would be

$$\mathbf{b}^{'L} = \{Min_{H_{c'}}[x_1], Min_{H_{c'}}[x_2], \dots, Min_{H_{c'}}[x_n]\}$$

and the upper bound vector would be

$$\mathbf{b}^{'U} = \{Max_{H_{c'}}[x_1], Max_{H_{c'}}[x_2], ..., Max_{H_{c'}}[x_n]\}$$

Then $[\mathbf{b}^{'L}, \mathbf{b}^{'U}]$ defines the cuboid for the level set $H_{c'}$, and its volume is designated as Vol.

> **Volume** is defined here as the product of the sides of a cuboid. For an n dimensional cuboid, the volume is expressed as
>
> $$Vol = \prod_{i=1}^{n} d_i$$
>
> where d_i is the length of the side of the cuboid measured on the i^{th} variable axis.

This cuboid, in effect, is an approximation of the boundary of the true level set $H_{c'}$. New points are randomly generated within this cuboid and those which satisfy the $f(\mathbf{x}) \leq c$ condition enter into the level set $H_{c'}$. The number of new points is as many as those discarded in the previous iteration.

Those feasible points which have been sampled are defined as **confirmed points**. This is introduced here to distinguished between points which exist in theory and those points which have been verified numerically to meet the relevant set conditions.

There is a danger that the cuboid estimate based on a limited number of confirmed points in a level set will be an underestimate, i.e. exclude a portion of the true level set with the consequent risk of missing a global optimum. Figure 2.2 displays a cuboid derived from limited number of confirmed points. The figure clearly shows that part of the level set is excluded from the cuboid. Corrections to the cuboid size to minimize the risk are discussed in Chapter 3, section 3.5.

It is clear in the example shown in Figure 2.1(b) that the level set $H_{c'}$ does not form a continuous set but consists of two disconnected subsets, technically a "partition set", which will be discussed in Chapter 3, section 3.1. The concept of a level set remains but the more complex boundaries make generation of new points in the level set far less efficient when the cuboid approximation is adopted. Such inefficiencies will often arise with multiple optima problems, when clusters of confirmed level set points eventually form around each optimum. One method of restoring the efficiency of the search is to handle each distinctive cluster as a separate sub-problem. In that case the search is run independently within a set of smaller cuboids, each enveloping one cluster, and will lead to the optimum solution for each sub-problem. The global solution to the original

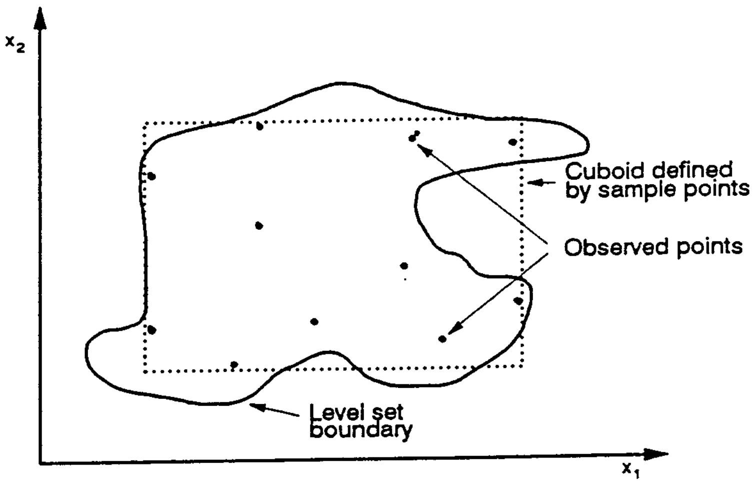


Figure 2.2: Cuboid derived from limited number of confirmed points.

problem is then given by the best of the sub-problem solutions. Identifying clusters and partitioning them is a problem which can be resolved by statistical approaches and these are also discussed in Chapter 3, section 3.1.

2.3.2 Alternative methods of sample point generation

The point generation scheme based on cuboids was described under LSP implementation in section 2.3.1 above. Other methods were considered in this research and are reviewed here.

In all direct search methods experimental feasible points are generated, the objective function is evaluated at each point, and then a ranking procedure based on the objective function values is used to guide the next stages of the search or detect the overall optimum solution. A good, computationally efficient, point generation method provides the maximum number of acceptable points for a given computational effort, and thereby increases the chance of finding the global optimum. The greatest opportunity for increasing the computational efficiency of direct search methods lies in an efficient sampling technique. In addition to increasing the efficiency of the search, a good sample point strategy also increases the reliability of finding the true optimum solution. Here, reliability is defined as the number of runs producing the true optimum solution divided by the total number of runs.

Grid sampling

This type of point generation occurs entirely at discretized, or systematically spaced, grid points. This method has two major shortcomings;

i) The location of sample points is fixed once the grid size and the starting point of the grid system are chosen. Variation in either the grid size or its starting point can lead to different solutions.

ii) If the grid point generation is adapted to iterative search with progressive domain reduction, there is a grid scaling problem after each iteration.

With grid scaling changes, and the retention of some points from earlier grid samples, the distribution of points can become distinctly non random after just a few iterations. Any appeal of systematic grid sampling is therefore lost with progressive domain reduction techniques such as LSP.

Sampling around existing best points

In this strategy a set of sample points is generated randomly and the best points identified by comparison. The bad points are discarded and replaced by new points generated around the surviving points. The principal difficulty with this method is in fixing the scale of the perturbation around the confirmed points. Too big a perturbation leads to an unnecessarily inefficient search. Too small a perturbation leads to greater efficiency in generating new points, but the risk of converging on local optima is significantly increased. With a small perturbation there is a natural tendency for the new points to form clusters around the best points sampled early in the search, but at the expense of exploring the rest of the feasible region. This leads to local convergence, especially with low N_{keep}.

Experiments were performed with all the above methods in conjunction with LSP and were rejected in favour of a point generation strategy based on cuboids. The cuboid strategy was found to produce the greatest search reliability and reasonable efficiency.

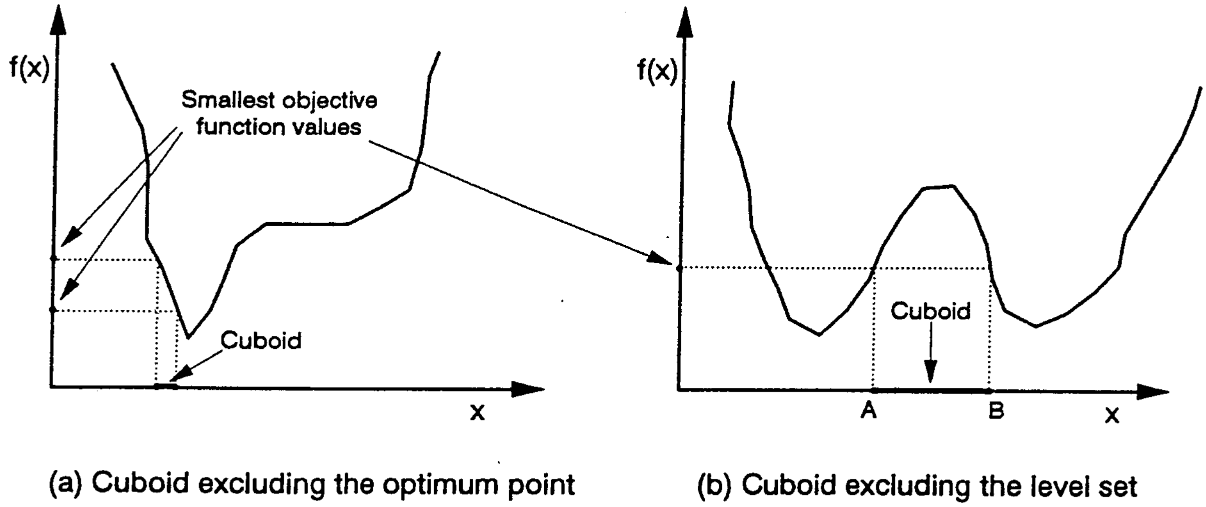
2.3.3 Revision of level set value-c

At every iteration some of the points in the previous acceptance set are discarded and the rest remain in the current level set. The screening criterion for keeping points in the acceptance set is simply that their objective function value is less than or equal to the current level set value-c. The current level set value-c can be set to any value between the minimum and the maximum objective function values at the confirmed points in the previous acceptance set. Only the mean value of the previous level set, which was proposed in [Chew & Zheng, 1988] though without justification, has been considered so far in this chapter, but other possibilities were explored in this research. First two extreme possibilities will be discussed.

If the maximum objective function value in the previous acceptance set is used as the new level set value then the cuboid volume and the level set value-c would remain nearly constant throughout the search, irrespective of the type of problem. In that case the search becomes almost a blind search, which is computationally inefficient.

If a small value is used as the level set value-c, so that just a few points with the smallest objective function values survive to the next acceptance set, the search loses its global nature and focuses on a small region in the search domain. As the volume of the cuboid reduces to a small size at the early stages of the search, the search can terminate at a non optimal arbitrary point, as shown in Figure 2.3(a) for a single variable problem.

Another example where the generation of new points would be almost impossible is shown in Figure 2.3(b). For the single variable problem, the two points which give close to the minimum objective function value are observed at the first iteration, say at points A and B as shown in Figure 2.3(b). When the minimum objective function value is taken as the level set value-c then the new cuboid would be the length $\overline{AB}$. Unfortunately there is no point which fulfils the level set condition in the cuboid except the extreme points themselves. In this case the search would be extremely inefficient or fail completely.



(a) Cuboid excluding the optimum point

(b) Cuboid excluding the level set

Figure 2.3: Cuboid defined by the smallest objective function values.

In general the revision in the level set value-c has to fulfil two conditions. Firstly, all of the important regions in the search domain should be retained in the cuboid following the process of discarding points which do not fulfil the level set condition. Secondly, the improvement in the level set value-c at each iteration must be substantial to avoid an impracticably large number of iterations and hence computationally unacceptable searches.

A set of test problems were used to investigate the relationship between the number of function evaluations and the level set value-c reduction at each iteration and, if possible, to identify a better alternative to setting $c' = M(f, c)$. Two different approaches were used to fix the revised level set values and define the current cuboid.

In the first approach, N_{keep} feasible points are first generated in the initial cuboid. Then the highest and lowest objective function values are identified and set to Max_f and Min_f respectively. The level set value-c is then defined as

$$c = Min_f + \kappa(Max_f - Min_f) \tag{2.17}$$

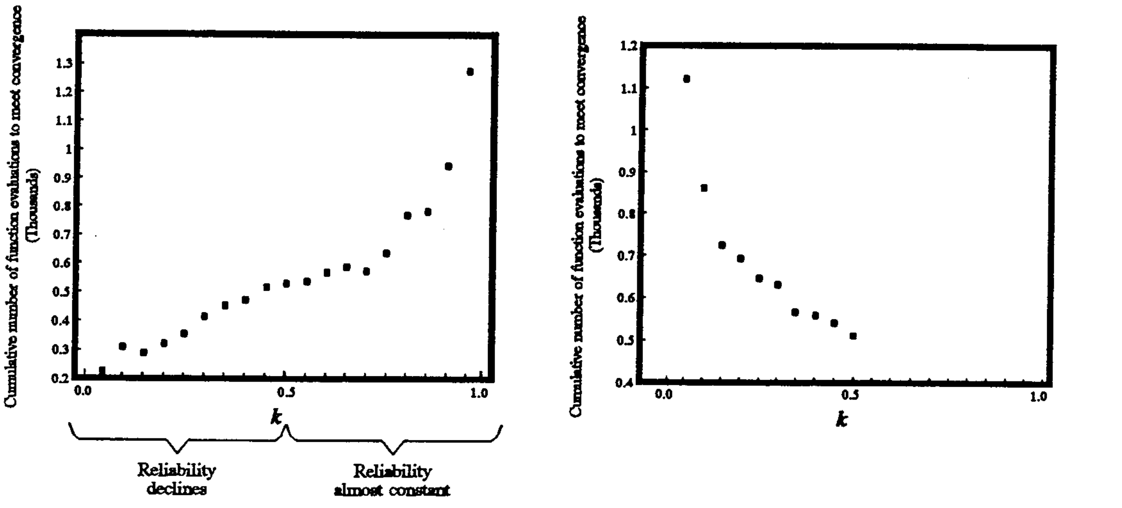
where κ is a constant and $0 \leq \kappa \leq 1$.

Once the new level set value-c is established, points whose objective function values are less than c (say n_x points) are retained and the rest discarded. The current cuboid is defined using the retained n_x points. New feasible points are generated to replace the discarded points, each point fulfilling the $f(\mathbf{x}) \leq c$ condition. This same procedure is then repeated at all subsequent iterations.

The results presented in Figure 2.4(a) and (b) were obtained with the Rosenbrock valley test problem [Problem 2-7-0-0, Appendix A]. The figures show the plots of κ versus the number of function evaluations. The values were averaged over 10 different runs. Different runs of the same problem with different values of κ were tried to identify the κ value which produces the least number of function evaluations. The same number of points, N_{keep}, in a level set and the same convergence criterion were used for all runs.

A typical plot of the number of function evaluations versus κ for the first approach is given in Figure 2.4(a). As shown in the figure, the highest number of function evaluations to meet the convergence criterion occurs when $\kappa = 1$, and this number gets lower as the value of κ approaches 0. But as κ approaches zero, the reliability of converging on the global optimum also declines, i.e. the global nature of the search is lost and it becomes a local search. In addition, indications of the existence of multiple optima are lost.

The second approach always generates new points within a cuboid prescribed by confirmed points in a level set defined by $M(f, c)$ but allows the value of c to advance below this level. This approach ensures that approximately $n_x = \frac{N_{keep}}{2}$ points are used to estimate the cuboid for point generation purposes. It avoids the difficulty of estimating cuboid bounds where n_x is very small. To generate new points a new level set value-c is defined as in equation 2.17. The new level set value-c could be as high as roughly $M(f, c)$ when $\kappa = .5$ or as low as Min_f when $\kappa = 0$. A plot of number of function evaluations against κ is given in Figure 2.4(b).



(a) Cuboid defined by level set value-c

(b) Cuboid defined by M(f,c), acceptability of points defined by c

Figure 2.4: N_f for cuboids defined by different level set value criteria.

In the case of this second approach, the search maintains its global nature, and can detect multiple optima. The numerical experiments showed that decreasing the level set value-c below $M(f,c)$ slightly improves the reliability. On the other hand, the number of function evaluations increases with lower values of c, and the search can be as inefficient as the blind search in the worst case i.e. when $\kappa = 0$.

In general, it is recommended that $M(f,c)$ be used as a criterion to advance the search, that is to set the level set value-c equal to $M(f,c)$. Then redefine the current cuboid using the approximately $\frac{N_{keep}}{2}$ surviving points which fulfil the $f(\mathbf{x}) \leq c = M(f,c)$ condition, and regenerate new points within this new cuboid. Although the choice of $M(f,c)$ for the revised level set value is essentially heuristic, these limited numerical experiments, as well as experience with the many test problems, appears to support this choice.

2.3.4 Detailed description of the LSP Algorithm - general case

Implementation of the LSP algorithm for the constrained case, using a cuboid approximation for the level set, is given below.

Let ϵ be a small number which will be compared with the variance of the objective function values to confirm convergence.

Let $D_0 = [\mathbf{b}_i^L, \mathbf{b}_i^U]$ for $i = 1, 2, .., n$; be the initial cuboid, where b_i^L and b_i^U are the initial lower and upper bounds respectively for the i^{th} variable.

Let $\xi = (\xi_1, \xi_2, ..., \xi_n)$ be an independent n - tuple random number which is uniformly distributed on $[0,1]^n$.

LSP operates by following the steps described below.

Initialization

1. Generate N_{keep} feasible random points in D_0 as follows:

   For a trial point j, generate $x_i^j = b_i^L + (b_i^U - b_i^L)\xi_i^j \quad$ for $i = 1, 2, ..., n$

   This will establish a trial point j, where $\mathbf{x}^j = (x_1^j, x_2^j, ..., x_n^j)$. Note that the generated point will be random uniform distributed within D_0.

   Test for feasibility, reject if infeasible. Continue until N_{keep} feasible points are obtained.

2. Evaluate the objective function, $f(\mathbf{x}^j)$, for each generated point. Assign the highest objective function value, c_1 as the initial level set value-c, that is

$$c_1 = Max\{f(\mathbf{x}^j)\}$$

Algorithm

3. Redefine a new cuboid D_1 using the remaining approximately $\frac{N_{keep}}{2}$ points as

   $D_1 = \{\mathbf{x} \mid b_i^{L1} \leq x_i \leq b_i^{U1}, \quad i = 1, 2, ..., n\} \subseteq D_0$

   where b_i^{L1} and b_i^{U1} are the lowest and highest values of the i^{th} variable among those points in the current level set.

4. Generate new points in the new cuboid.

   For a new trial point j, generate $x_i^j = b_i^{L1} + (b_i^{U1} - b_i^{L1})\xi_i^j \quad$ for $i = 1, 2, ..., n$

   Test for feasibility, reject if infeasible.

   Evaluate $f(\mathbf{x}^j)$; reject if $f(\mathbf{x}^j) >$ level set value c.

   Generate as many confirmed points as those discarded in step 8 of the previous iteration. That is, restore the number of points in the acceptance set to a total of N_{keep}.

5. Evaluate the mean of the objective function values, and assign this mean value to the new level set value-c for the next iteration. That is

$$c' = M(f,c)$$

6. Calculate $V(f,c)$, the variance of the objective function values.

$$V(f,c) = \frac{1}{N_{keep}}[\sum_{H_c}(f(\mathbf{x}) - M(f,c))^2]$$

   If the variance is less than the pre-specified convergence tolerance, ϵ, terminate the search, otherwise continue to the next step.

7. Discard points whose objective function values are greater than c'.

8. Set $c = c'$

9. Go back to step 4 and redefine a new cuboid.

Let the value of c at termination be represented by c_t. $V(f, c_t) \leq \epsilon$ is satisfied so that $c^* \approx M(f, c_t)$. The solution is presented in the form of a final acceptance set $H_{c^*} \approx H_{ct}$ and will contain points in the immediate vicinity of the global optimum or optima.

2.3.5 Alternate termination criteria

Because of its unique search mechanism, LSP's requirements for a termination criterion differ from other NLP search methods. In addition to the termination criterion described in section 2.3.4, three other alternatives were considered and are described below.

i) Size of cuboid

In the special case where the problem is unimodal, the cuboid at $c = c^*$ is theoretically a single point and cuboid volume can be exploited to provide a convergence criterion.

If $Vol = \prod_{i=1}^{n} d_i$ defines the volume of the cuboid at any stage of the LSP search in an n dimensional problem, where d_i is the length of the side of the cuboid measured on the i^{th} variable axis, then this volume tends to zero as convergence is approached. The search could be terminated when the volume is less than a pre-specified small value. Since the length of each side of the cuboid gets closer to zero as the search progresses, the fact that $Vol \rightarrow 0$ as $d_i \rightarrow 0$ for $i = 1, ..., n$ is obvious. But the danger is that the convergence criterion could be met even if only one or two sides approach zero length and a premature or pseudo-convergence could result. To avoid this a criterion requiring that the sum of the cuboid side lengths, $\sum_{i=1}^{n} d_i$, be close to zero could be used in addition to the volume criterion. As a further alternative the size of each individual cuboid side length in the final cuboid could be compared to pre-specified termination criterion. This last mentioned criterion is similar to that used in many local optimization methods [Archetti & Schoen,1984; Betro & Schoen, 1987; Boender & Kan, 1987], which stipulate that if $\mathbf{x}^*$ is the current estimate of the minimum point and $\mathbf{x}$ is the last predicted minimum point then convergence is considered to have been achieved if

$$| x_i - x_i^* | \leq \varepsilon_0 \qquad \text{for } i = 1, 2, ..., n.$$

where ε_0 is a pre-specified small value.

ii) Difference between consecutive level set values

The difference between the objective function values at consecutive iterations is used as a convergence criterion by some direct search methods. For instance, the following simultaneous termination criteria are used for Newton's direct search method [Edgar and Himmelblau, 1988].

$$\mid \frac{f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)}{f(\mathbf{x}^k)} \mid < \epsilon_1$$

or as $f(\mathbf{x}) \rightarrow 0$,

$$\mid f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k) \mid < \epsilon_2$$

$$\mid \frac{x_i^{k+1} - x_i^k}{x_i^k} \mid < \epsilon_3$$

or as $x_i^k \rightarrow 0$,

$$\mid x_i^{k+1} - x_i^k \mid < \epsilon_4$$

$$\|\mathbf{s}^{k+1}\| < \epsilon_5$$

or

$$\|\nabla f(\mathbf{x})\| < \epsilon_6$$

where k is the iteration number, $\mathbf{s}$ is the search direction and ∇ is the gradient operator.

Similarly with LSP, differences between level set values at consecutive iterations can be used as a termination criterion. In the later iterations of a successful LSP search, improvement of the level set value-c with each iteration decreases. This is an indication that the points in the level set are distributed around stationary points in the objective function surface. A maximum acceptable value, ϵ_o, for the difference between two consecutive level set values, can serve as a termination criterion. That is, $\mid c_i - c_{i+1} \mid \leq \epsilon_o$ dictates termination.

iii) Coefficient of variation of the objective function values

The variance of the objective function values in the acceptance set give a precise theoretical convergence condition, and has to be equal to zero at the global optima. But in actual implementation, the variance of the objective function values is compared against a predefined small value at each stage of the search. Since values of the objective function with high absolute value produce a higher variance, and objective function values close to zero produce lower variance, a scaling difficulty arises with the implementation of variance as a convergence criterion.

To overcome the scaling difficulty, the dimensionless coefficient of variation, defined here as modified variance divided by the level set value-c, can be used as a convergence criterion. This criterion seems to be quite satisfactory for a wide range of problems, but a difficulty arises when the level set value equals zero and the search stops because of numerical overflow. The level set value can be equal to zero for two reasons: when the global optimum is just below or equal to zero; and when the level set value starts with a positive number and improves to a negative optimal value, taking a value equal to zero in the course of the search process.

Generally then, the choice of search termination criterion is tied to the magnitude of the optimum objective function. Except for the criterion discussed under *'size of cuboid'*, any of the above mentioned criteria, as well as the one discussed in section 2.3.4, can be used for LSP search termination. The *'size of cuboid'* criterion undermines the search for multiple optima solutions.

2.3.6 Handling Constraints in LSP

LSP can accommodate constraints by making use of three simple and straightforward approaches. These are: rejection, reduction and penalty methods. All three methods can be applied in the same LSP solution of a given problem, depending on the nature of the constraints involved. An explanation of each method is given below.

Rejection method

For a constrained problem with a feasible region S, defined by the set of the constraints, the mean value and the variances over the level set were redefined by [Chew & Zheng, 1988] as

$$M(f,c;S) = \lim_{c_k \downarrow c} \frac{1}{\mu(H_{c_k} \bigcap S)} \int_{H_{c_k} \bigcap S} f(\mathbf{x}) d\mu$$

$$V(f,c;S) = \lim_{c_k \downarrow c} \frac{1}{\mu(H_{c_k} \bigcap S)} \int_{H_{c_k} \bigcap S} [f(\mathbf{x}) - M(f,c;S)]^2 d\mu$$

$$V_M(f,c;S) = \lim_{c_k \downarrow c} \frac{1}{\mu(H_{c_k} \bigcap S)} \int_{H_{c_k} \bigcap S} [f(\mathbf{x}) - c]^2 d\mu \tag{2.18}$$

where k indicates the iteration.

The above expressions are called the rejection mean value, rejection variance and rejection modified variance of $f(\mathbf{x})$ over $H_c \bigcap S$ respectively as discussed in section 2.1. The following important characteristics are retained. Given

$\mathbf{x}^* \in S$ is the global minimum solution and

$c^* = f(\mathbf{x}^*)$ is the corresponding global minimum value.

for $c > c^*$

then

$$M(f,c;S) \geq c^*$$
$$M(f,c^*;S) = c^*$$

$$V(f, c^*; S) = 0$$
$$V_M(f, c^*; S) = 0$$

For a discrete sampling (or integer problem), equation 2.18 can be rewritten as

$$M(f, c; S) = \frac{1}{\sum_{\mathbf{x}^j \in H_c \bigcap S} a^j} \sum_{\mathbf{x}^j \in H_c \bigcap S} a^j f(\mathbf{x}^j)$$

$$V(f, c; S) = \frac{1}{\sum_{\mathbf{x}^j \in H_c \bigcap S} a^j} \sum_{\mathbf{x}^j \in H_c \bigcap S} a^j [f(\mathbf{x}^j) - M(f, c)]^2$$

$$V_M(f, c; S) = \frac{1}{\sum_{\mathbf{x}^j \in H_c \bigcap S} a^j} \sum_{\mathbf{x}^j \in H_c \bigcap S} a^j [f(\mathbf{x}^j) - c]^2 \tag{2.19}$$

Reduction method

Whenever there are equality constraints, the number of variables in the problem can theoretically be reduced by the number of equality constraints. In effect reduction can be performed by solving a system of simultaneous nonlinear equations. Most of the time handling equality constraints in this fashion is difficult, if not impossible, since in practice nonlinear constraint functions are seldom simple enough to be able to equate the function with a single variable.

Penalty method

When the set of constraints defines a very small or narrow feasible region, or one governed by equality constraints, generating feasible points can be difficult. In this case the objective function can be modified by adding some terms which penalize (i.e. increase) the value of the objective function when any sample point outside the feasible region is evaluated. The existence of points that violate the constraints is permitted within the acceptance set but is discouraged later in the search. This approach is common to all penalty function methods.

Let S be a closed subset of X. Consider the constrained minimization problem

$$c^* = \min_{\mathbf{x} \in S}[f(\mathbf{x}) + \alpha p(\mathbf{x})]$$

where α is a positive value.

Definition:- a function p on X is a penalty function for the constrained set S if

$$p(\mathbf{x}) \geq 0 \qquad \forall \mathbf{x} \in X$$

$$p(\mathbf{x}) = 0 \qquad \text{iff } \mathbf{x} \in S$$

$$p(\mathbf{x}) > 0 \qquad \forall \mathbf{x} \notin S$$

Suppose c_k, the level set value at the k^{th} iteration, so that $c_k \geq c^*$, is decreasing and tends to c^* as $k \to \infty$ and α_k is a positive increasing sequence which tends to infinity as $k \to \infty$. Let

$$H_k = \{\mathbf{x} \mid f(\mathbf{x}) + \alpha_k p(\mathbf{x}) \leq c_k\} \tag{2.20}$$

then

$$\lim_{k \to \infty} H_k = \bigcap_{k=1}^{\infty} H_k = H_c \bigcap S$$

and

$$lim \frac{1}{\mu(H_k)} \int_{H_k} f(\mathbf{x}) d\mu = \frac{1}{\mu(H_c \bigcap S)} \int_{H_c \bigcap S} f(\mathbf{x}) d\mu.$$

As defined earlier, the mean value of $f(\mathbf{x})$ over its level set H_c within a closed feasible region defined by the constraint set S is given by

$$M(f, c; S) = \frac{1}{\mu(H_c \bigcap S)} \int_{H_c \bigcap S} f(\mathbf{x}) d\mu.$$

Because c^* is a lower bound of c

$$\lim_{k \to \infty} \frac{1}{\mu(H_k)} \int_{H_k} f(\mathbf{x}) d\mu = M(f, c; S)$$

The limit does not depend on the choices of sequences $\{c_k\}$ and $\{\alpha_k\}$.

Suppose $c > c^*$ then the limit

$$M(f, c; p) = \lim_{k\to\infty} \frac{1}{\mu(H_k)} \int_{H_k} f(\mathbf{x}) d\mu$$

is called the penalty mean value of $f(\mathbf{x})$ over the level set H_k. Suppose $c^* = f(\mathbf{x}^*)$ is the global minimum value of $f(\mathbf{x})$ over S; then

$$c^* = M(f, c^*; S) = M(f, c^*; p)$$

The penalty mean value, variance and modified variance of $f(\mathbf{x})$ have the same convergence properties as the constrained mean value, variance and modified variance; therefore they share the same properties [Chew & Zheng, 1988].

2.4 Summary

The general theoretical description of the use of level sets for nonlinear optimization was laid down by Chew and Zheng [Chew & Zheng, 1988]. In this chapter, the theory has been extended, the potential importance elaborated and some practical application issues addressed.

One of the main ideas explored is the relationship between the reliability of LSP solutions and the revision of the level set value at each iteration. The choice of revised level set value was found to affect the reliability of the solutions and the overall performance of LSP but the use of $M(f, c)$ was confirmed as being appropriate for general applications.

Different methods of sample point generation were also investigated. The exploration of alternative methods has helped to give more insight into a key numerical and computational aspect of this implementation of level set theory. Even though many methods were examined, only one of the methods, the cuboid approach, is recommended for general application.

Various search termination criteria were also examined. Any of these criteria can be used in practice without affecting the theoretical global convergence characteristics though the best choice is dependent upon the nature of the problem being solved.

In practical implementations of LSP, the modified variance, V_M is adopted as a termination criterion instead of the variance evaluated around the mean. The reason is that the modified variance is larger and therefore provides a stronger convergence condition and is, as well, easier to evaluate. Moreover the modified variance is more sensitive to the entry of points into the acceptance set which have the lowest values of the objective function. This becomes a significant benefit in the final convergence stages of the search. In addition to using variance as a convergence criterion, a plot of level set value-c against each iteration is displayed to give a visual indication of the typically asymptotic convergence of level set value-c on c^*.

How closely the final value of the convergence criterion $V_M(f, c)$ approaches zero in practice is a question of numerical precision and the practical needs of the problem. It is governed primarily by the number of sample points maintained in the level set and accuracy of the computational device used in evaluating the various nonlinear functions involved in the problem. In general the most important aspect of LSP is fixing the number of points in an acceptance set. Too few points increases the chances of missing very localized but significant fissures in the objective function surface, too many points places an unnecessarily high computational burden. For a fixed number, N_{keep}, of points in the acceptance set the density of points per unit length on the variable axes increases with the number of iterations.

The search process is not infallible, a very localized fissure in the $f(\mathbf{x})$ surface which contains the global optimum can be overlooked, but the $V_M(f, c)$ value still converges on zero over one or more local optima. Such problems are more likely to emerge when the number of sampling points in a level set is too low. Problems which are peculiarly difficult for LSP are thus possible and some of these are discussed in Chapter 5, section 5.5.

Chapter 3

LSP PERFORMANCE IMPROVEMENT

Optimization based purely on the exploration of level sets has been described by some authors as a theoretical approach with little practical significance [Torn & Zilinskas, 1988], since they felt that the method lacks strong convergence characteristics. There have been few attempts to implement pure level set methods in any general way. Those implementations that have been made are poorly documented and there is little existing literature on enhancing the performance with LSP type methods. A variety of approaches to improve the efficiency of the LSP search have therefore been investigated in this research. The more successful techniques dealing with partitioning of search domain, relaxation of constraints and redefining of search domain are discussed in this chapter. In addition to improving the computational efficiency of LSP, these techniques also tend to produce more accurate convergence on optimal solutions, and enhance the reliability of detection of boundary solutions. Further discussion of actual experience when implementing these techniques and setting the values of the various LSP parameters is provided in chapter 5.

3.1 Partition sets and cluster analysis

The volume occupied by the level set is always less than or equal to any containing cuboid volume. The ratio of level set volume to cuboid volume has a direct influence on the efficiency of generating new points in the level set; when this ratio is low considerable computational effort will be expended on unsuccessful point generation. Very low ratios can result from problems involving widely separated, multiple global optima, or from a long, narrow, level set with its principal axis at an oblique angle (e.g. 45^o) to the variable axes.

For a typical unimodal problem, the ratio of level set to cuboid volume remains both high and nearly constant throughout the search. But if the optimization problem has multiple global optima, then the cuboid would include a large amount of non level set space. This results in a very low level set to cuboid volume ratio. Moreover the level set could consist of discontinuous sets within the cuboid.

Some new terms are introduced here to facilitate discussion of the computational implications when the level set becomes discontinuous.

Two sets B and C, where $B \neq C$, define a **disjoint set** if $B \bigcap C = \emptyset$, that is if B and C do not have any element in common. A class $\mathcal{A}$ of sets is called a disjoint class of sets if each pair of distinct sets in $\mathcal{A}$ is disjoint [Lipschutz, 1965].

A class $\mathcal{A}$ of non empty subsets of A is called a **partition** of A if and only if each $a \in A$ belongs to some member of $\mathcal{A}$ and the members of $\mathcal{A}$ are mutually disjoint [Lipschutz, 1965].

A space A is **connected** if, whenever it is decomposed as the union $B \bigcup C$ of two non empty subsets, then $\overline{B} \bigcap C \neq \emptyset$ or $\overline{C} \bigcap B \neq \emptyset$, where the upper bar designates the complement set and $\emptyset$ designates the null set [Armstrong, 1983].

If a level set defines a region where all points in the level set belong to a single set, then the level set is called connected as in Figure 3.1(a). Henceforth in this thesis a connected subset of the partition will be referred to simply as a "subset".

If a level set defines a set of small subregions and there is no common point between any two subregions, then the level set is called a **partition level set.**

When the subsets of the partition are close together they will, in practice, be treated as if they were a connected set. Figure 3.1(b) shows a partition level set in two dimensions.

Sub-cuboid will be defined here as the cuboid enclosing an individual subset of a partition level set.

Minimal cuboid enclosure will be defined in this thesis as the set of sub-cuboids which enclose the level set in a specific cuboid. The minimal cuboid enclosure would be equal to the cuboid itself if the level set is connected, otherwise it would be a collection of more than one sub-cuboid.

As the iterative process of LSP unfolds, the size of the region occupied by the acceptance set gets smaller. However, this region may contain a single connected level set in the special unimodal case, or it can enclose a partition level set forming separate discontinuous regions. Points generated in any of these subregions would be accepted if they meet the level set condition $f(x) \leq c$ and do not violate any of the constraints. The efficiency of generating an acceptable sample point in the cuboid would be directly proportional to the ratio of the acceptance region volume to the cuboid volume.

For a multiple optima problem, the chances of generating an acceptable point becomes smaller as the search approaches the global optimum value of the objective function. This

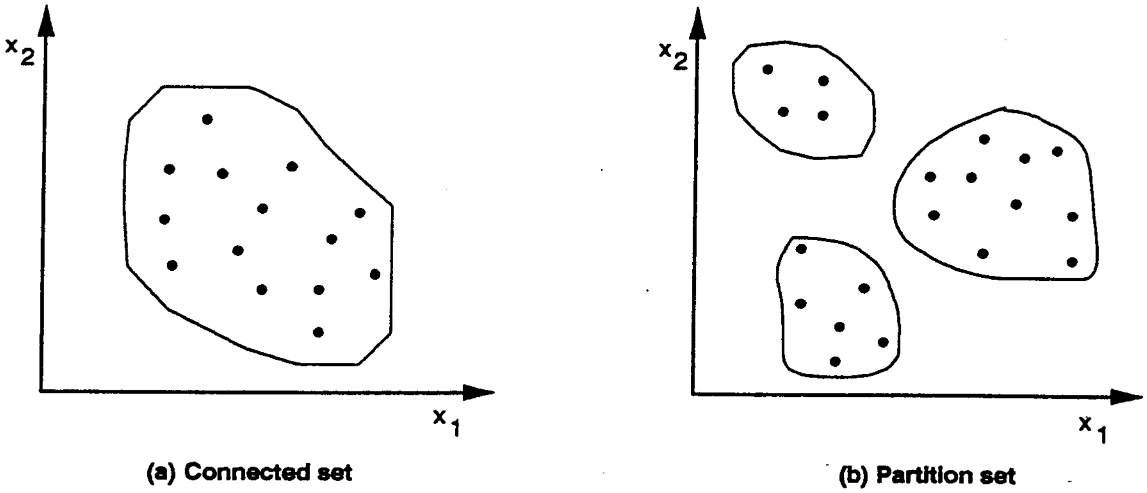


Figure 3.1: Connected and partition sets.

is because the volume of the acceptance region becomes extremely small near convergence whereas, at this stage of the search, the cuboid typically maintains more or less a constant volume. This is particularly true when the global optimum points are at stationary points in the objective function surface. Although an inefficient search may eventually converge, the time required to obtain a solution may be prohibitive. To overcome the inefficiency due to the presence of multiple optima it is possible to subdivide the cuboid into a set of smaller cuboids without sacrificing the integrity of the search. This is provided, of course, that no subregion containing a global optimum is omitted from the smaller cuboid set. If the existence of distinct clusters of points is suspected during a search, statistical cluster analysis can be used as an aid to identifying the clusters, and hence the sub-cuboid boundaries. Details on cluster identification and classification will be discussed in section 3.1.1. The current level set value-c would be applicable to all of the sub-cuboids at the point of dividing the cuboid into smaller sub-cuboids.

After the clusters have been identified their individual subcuboid dimensions are defined. The combined sum of the volume of all sub-cuboids would be smaller than the volume of the cuboid before subdividing into smaller subregions. Thus the total volume of search is reduced relative to the volume of the level set.

Once subcuboids are defined the optimization problem is then treated as a separate problem in each subregion and each is solved independently. The process of performing cluster analysis and subdividing the cuboid into smaller subcuboids can be repeated at subsequent stages of the LSP search but will result in solving an ever increasing number of sub-problems. Such an exhaustive division of search domain leads to an inefficient search and is generally unjustified.

An example to show the merits of using smaller cuboids at some stage of the search for the unconstrained but bounded one dimensional case is given in Figure 3.2. Here the cuboid is one dimensional and its "volume" is represented by the length of its single side. At the k^{th} iteration, the chances of generating a feasible point fulfilling $f(x) \leq c_k$ is roughly 0.5, that is the level set designated as H_{c_k} in the figure, occupies about half the cuboid length. At the next iteration, the level set value-c is lowered to c_{k+1}, and the corresponding level set volume $H_{c_{k+1}}$ becomes so small that the efficiency of generating feasible points fulfilling $f(x) \leq c_{k+1}$ drops to below 0.25. After this stage of the search it would be advantageous to work with two distinct cuboids where the chances of generating points fulfilling the level set condition would approach 1.0, depending on how precisely the cuboid boundaries are estimated.

Recognizing such benefits from partitioning, it is almost essential that a cluster analysis subroutine be incorporated into LSP. From the implementation point of view, the precision needs for the identification of clusters and cluster bounds are not critical. Except, of course, that the cluster boundaries and hence cuboid boundaries do not inadvertently exclude a global optimum.

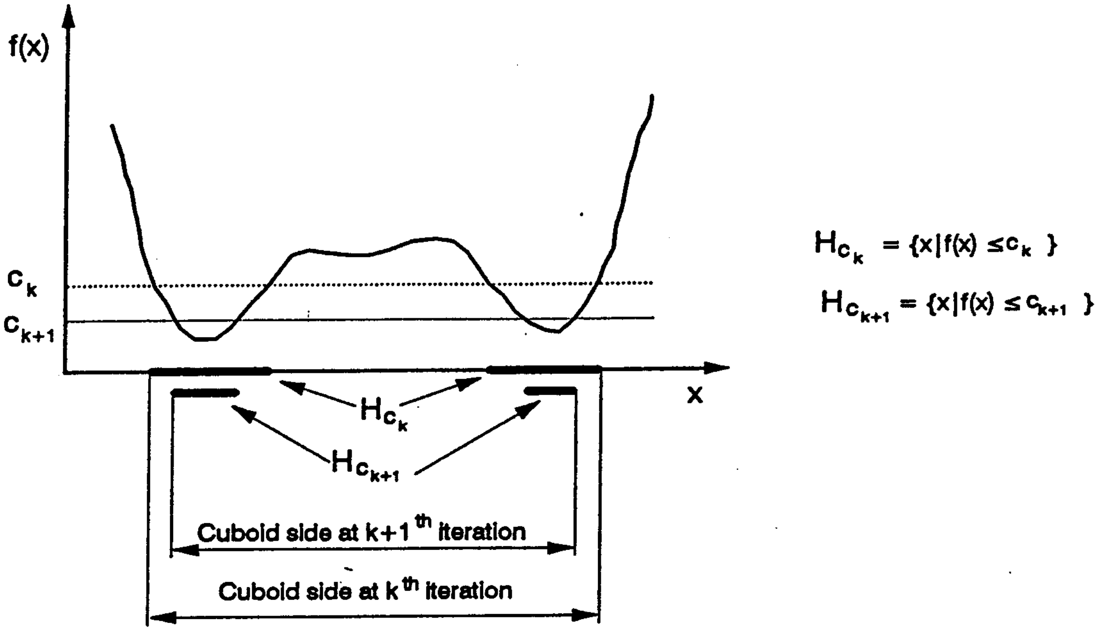

f(x)
c_k
c_{k+1}
x
H_{c_k}
$H_{c_{k+1}}$
Cuboid side at k+1 th iteration
Cuboid side at k th iteration
$H_{c_k} = \{x | f(x) \leq c_k\}$
$H_{c_{k+1}} = \{x | f(x) \leq c_{k+1}\}$

Figure 3.2: Level set "volume" in one dimensional problem with and without partitioning.

3.1.1 Cluster analysis methods

A cluster can be defined as a set of points that possess some common characteristic which is not shared by members of another group, or as a set of points that bear some relationship to each other but not with those outside that particular set. In Euclidean space, where the relationship between points is a function of their distance from each other, clusters are defined as points which are located close to each other. In some cases a similarity criterion, which expresses the common characteristics shared between points, is used instead of distance to classify points into clusters. When partitioning level sets, Euclidian distance is the appropriate measure of the relationship between points in the decision space. Cluster analysis is a technique which allocates points to a cluster in such a way as to maximize the distance between clusters and minimize the distance between points within a cluster. Cluster analysis procedures can be divided into two major categories, which are briefly explained here.

- *Hierarchical Methods:*

  These produce a tree-like taxonomic system where, at one end of the tree every individual point is a cluster and at the other end all points are included in a common cluster. At intermediate levels, clusters are formed by aggregating lower clusters. The hierarchical cluster configuration is usually represented graphically as a dendrogram as shown in Figure 3.4. This graphical structure is a common tool used to express the results of a clustering analysis. The hierarchical method can be divided into agglomerative and divisive methods.

  i) Agglomerative methods: These are methods where similar data points are sequentially aggregated to form a single cluster. The methods construct the hierarchical tree from individual points at the branch tips to a single root. Specific agglomerative methods differ according to their definition of distance between a point and a cluster, or between two clusters [Mezzich & Solomon, 1980].

  ii) Divisive clustering methods: These methods subdivide the aggregate set of data points into smaller subgroups by partitioning, such that the variance within each group is minimized and the variance among the groups, their mutual separation, is maximized without breaking up natural groupings. One of the techniques is the Automatic Interaction Detection, AID, which employs a series of discrete variable splits, each dictated by the maximum reduction in the empirical unexplained variance [Jenson, 1977].

- *Non-hierarchical methods:* In contrast to hierarchical methods, these produce configurations that do not present rankings in which lower order clusters must become members of larger more inclusive clusters. These are methods which essentially produce the final partition in a single step. The major approaches used in non-hierarchal methods are as follows [Mezzich & Solomon, 1980].

i) Total enumeration of partitions methods: In this method an attempt is made to enumerate all clustering possibilities, and then select the best cluster arrangement. A quantitative clustering index is used to choose the best alternative. Since this method involves exhaustive searches, it appears to be computationally unattractive even for modest data sets.

ii) Nearest centroid sorting methods: The basic feature in this method is the selection of seed points to be used as cluster nuclei around which the set of points can be grouped. When the number of clusters is fixed, the location of the centres of the clusters are updated after each full cycle of allocation of points. Given the number of clusters, the method allocates points so that the within cluster sum of squares are minimized [Hartigan, 1977]. For a variable number of clusters, points are allocated on the basis of the nearest centroid sorting process. The number of clusters changes during the allocation process by partitioning those clusters which have a large within cluster sum of squares to form smaller clusters, and joining any two clusters with small between cluster sum of squares to form a larger cluster. This is controlled by using certain parameters for "coarsening" and "refinement" set by the user [Mezzich & Solomon, 1980].

iii) Reallocation methods using variance-covariance criteria: The basic procedure in this category is to reallocate points among a set of clusters in such a way as to optimize some overall discriminant function or variance-covariance criterion.

iv) Density search methods: These methods use the allocation of points in a metric space, looking for regions of high point density separated by regions of low density for identification of clusters.

From the experience in this research, the agglomerative hierarchical clustering method was found to be easy to implement with the least complications in theory and application and providing acceptable precision for LSP purposes. This method is discussed in detail in the next section.

3.1.2 General implementation of cluster analysis methods

Suppose there are m points, in n dimensional Euclidean space, which we wish to arrange into a hierarchal classification. The data set for the m points forms an $m \times n$ matrix. The unit of measurement for the $m \times n$ raw data matrix is standardized, which means that all variables are transformed to a single unit, before computing distance measurements. This ensures that each variable is weighted equally, otherwise the Euclidean distance d_{ij} will be influenced most strongly by the variable which has the greatest magnitude [Davis, 1973]. The Euclidean distance, d_{ij}, between two points is computed using equation 3.1.

$$d_{ij} = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \tag{3.1}$$

where x_{ik} denotes the k^{th} variable measured at point i and x_{jk} is the k^{th} variable measured at point j. In all, n variables are measured at each point and d_{ij} is the distance between point i and point j. The distance between all possible pairs of points will result in an $m \times m$ symmetrical matrix. The points are arranged into a hierarchy based upon the magnitude of d_{ij}. Initially all pairs of points with mutually short distances are grouped together to form the beginning of clusters. Then those points which are not in any of the clusters formed at the initial stage, together with the small clusters already formed, are regrouped to form larger clusters. Small clusters are treated in the same way as points, and are lumped together on the basis of their mutual distances. The process of regrouping is repeated until all of the points have been placed into a complete classification scheme.

The first step in clustering by the pair-group method is to find the mutually closest distance in the matrix, the corresponding data pairs form the centres of subsequent clusters. After the centres have been identified, all the other points are connected to the centres one by one to form a dendrogram.

The most common and simple clustering method is the single linkage method, which employs a predefined arbitrary critical distance as a criterion for clustering. Clustering will only occur between pairs of points and small clusters if the distance between them is less than the critical distance. If all pairwise distances are greater than the critical distance no clustering can occur. After the initial clusters have been formed the distances between the remaining points and the closest point within a cluster are computed one by one. Those points which have distances less than the critical distance from a cluster are entered into the cluster. The procedure continues until either no more points can enter any cluster, or all points agglomerate to a single cluster.

A similar method, referred to as the multiple linkage or complete linkage method, uses the distance between an unclustered point, and the farthest point in a cluster as a measure for clustering. The distance between an unclustered point and the most distant point within a cluster is checked against the critical distance for the viability of being a member in that cluster. Figure 3.3 shows two different results of cluster analysis resulting from a single data set using single and multiple linkage cluster analysis methods.

Another type of cluster analysis method, the weighted average cluster analysis, uses equally weighted average distance between the cluster and a newly connected point. This method gives equal importance to the set of points which have already formed a cluster and the new point which is to enter the cluster, hence the greater influence of the latter. In this method, once the first grouping is completed, the distance matrix must be recalculated, treating each set of grouped points as a single point. Then the next step is to regroup the small clusters already formed on the basis of shortest mutual distance to

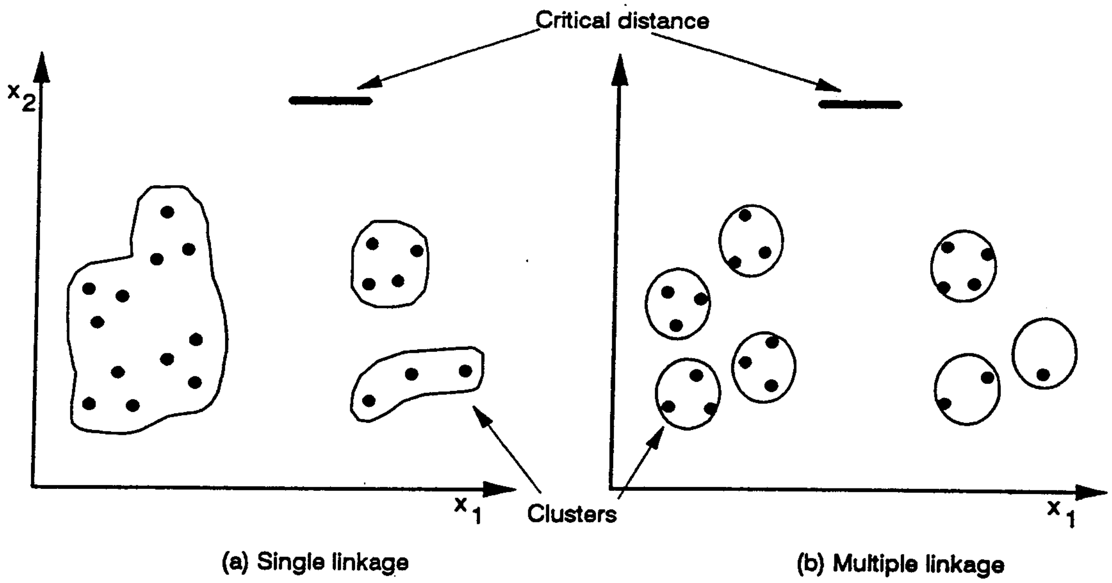


Figure 3.3: Single and multiple linkage clustering methods.

form a larger cluster. A simple and effective way to calculate the distance between the grouped points is to use the arithmetic mean of distances between every pair of points, each point from a different cluster. For example, two clusters, say the first one containing points $A\&B$, and the second one containing points $C\&D$, are made to be regrouped to form a single big cluster. The distance between the two small clusters is calculated as $(\overline{AC}+\overline{AD}+\overline{BC}+\overline{BD})/4$, where each pair of letters under a bar line designate the distance between the two points. In general the distance between two clusters, D_{IJ}, is expressed as

$$D_{IJ} = \frac{1}{n_I n_J}\sum_{i,j} d_{ij}, \quad i \in I,\ j \in J \tag{3.2}$$

where

D_{IJ} is the distance between cluster I and cluster J.

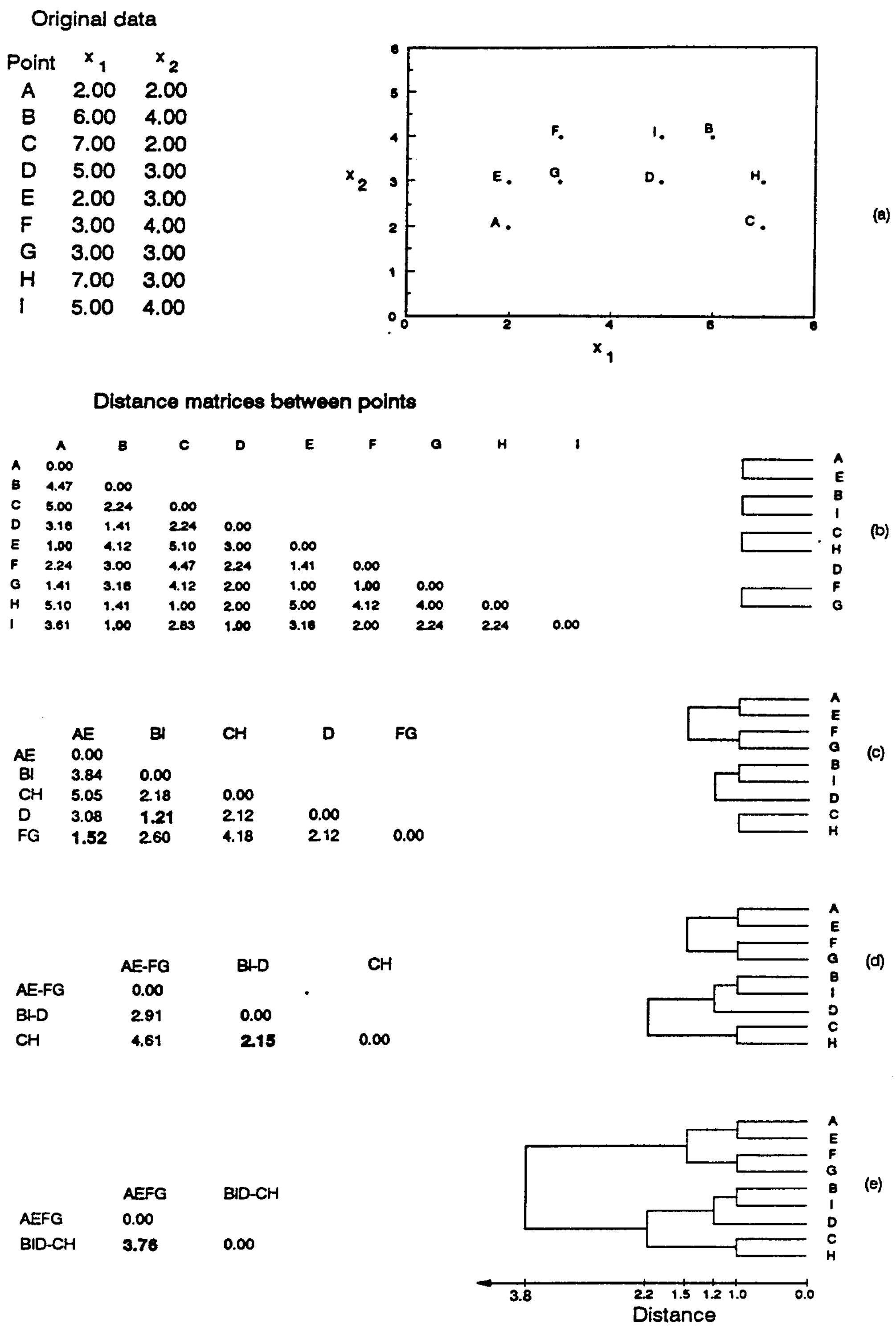
n_I and n_J are number of points in cluster I and J respectively.

d_{ij} is the distance between points i and j.

Because points in a cluster are treated as a single point, averaging the distances in the cluster introduces distortion into the dendrogram. Points located far out from the centre of the cluster have the greatest influence on the structure. The distortion is introduced when two clusters with unequal number of points join to form a bigger cluster. For example, if two points A and B are in a cluster, and a third point C is to enter the cluster, then the distance between the cluster and the third point is calculated as the average distance between points A and C and the distance between points B and C. Thus point C is involved twice while the other points are each used only once. The distortion is increasingly apparent as successive levels of clusters are averaged together. The severity of this distortion can be evaluated by examining the matrix of cophentic values, which describes the apparent distance within the dendrogram. The cophentic values and original distance matrix can be correlated to reveal the overall degree of distortion.

Unweighted average clustering methods are similar to the weighted average clustering methods except that they assume an equal number of points in each cluster. In these methods late entries into a large cluster have almost no influence on the distance calculations of the clusters.

Figure 3.4 demonstrates how to construct a dendrogram using 9 data points in a two dimensional space. Figure 3.4(a) shows the original data set in two dimensions. The matrix of distances is calculated from the ordinates of each point. The pairs of points with mutually closest distances, are grouped and connected together as shown in figure 3.4(b). Figure 3.4(c), 3.4(d) and 3.4(e) demonstrate, step by step, how the distance matrixes are calculated for already clustered points, how new points are entered into the structure, and how small clusters are sequentially agglomerated to form the final dendrogram.



Original data

Point	x_1	x_2
A	2.00	2.00
B	6.00	4.00
C	7.00	2.00
D	5.00	3.00
E	2.00	3.00
F	3.00	4.00
G	3.00	3.00
H	7.00	3.00
I	5.00	4.00

Distance matrices between points

	A	B	C	D	E	F	G	H	I
A	0.00								
B	4.47	0.00							
C	5.00	2.24	0.00						
D	3.16	1.41	2.24	0.00					
E	1.00	4.12	5.10	3.00	0.00				
F	2.24	3.00	4.47	2.24	1.41	0.00			
G	1.41	3.16	4.12	2.00	1.00	1.00	0.00		
H	5.10	1.41	1.00	2.00	5.00	4.12	4.00	0.00	
I	3.61	1.00	2.83	1.00	3.16	2.00	2.24	2.24	0.00

	AE	BI	CH	D	FG
AE	0.00				
BI	3.84	0.00			
CH	5.05	2.18	0.00		
D	3.08	**1.21**	2.12	0.00	
FG	**1.52**	2.60	4.18	2.12	0.00

	AE-FG	BI-D	CH
AE-FG	0.00		
BI-D	2.91	0.00	
CH	4.61	**2.15**	0.00

	AEFG	BID-CH
AEFG	0.00	
BID-CH	**3.76**	0.00

Figure 3.4: Stepwise dendrogram construction.

There are many ways of allocating points into taxonomic classifications. The choice of methods depends mostly on how the analyst views the needs of his problem and wishes to trade off computational effort against the desire for accuracy of cluster identification.

3.1.3 A clustering routine for LSP

A cluster analysis subroutine which performs a weighted average cluster analysis was incorporated to LSP. This method is chosen for its simplicity and the fact that it needs only a few user selected parameters. Even though the method might not be the most accurate, it is good enough for LSP purposes since the search does not need precise clustering method for its solution. With LSP the potential for overall reduction in computational load and gain in reliability from sophisticated clustering methods is not obvious.

A user selected type clustering criterion, which is expressed as the ratio of the minimum distance between two clusters and the distance between the mutually farthest points within a cluster, is used as an input for the analysis. The subroutine checks if the clustering criterion is met, and performs sub-grouping of the data set into a number of clusters when the criterion is met. The bounds for each cluster, or subgroup, are identified and then used to define the initial cuboids for the respective clusters. This subroutine is adapted from [Zupan, 1982].

The computational time the search needs while performing the clustering routines depends on the strength of the clustering criterion. In this thesis, a clustering criterion is referred to as a strong criterion when the ratio of distances between two clusters to the distance between the farthest points within a single cluster is high. A strong criterion implies the expectation of very distinct clusters.

If a very strong clustering criterion is used, it might take longer computational time to meet the criterion, but once the clusters are identified and the problem is subdivided into the true set of distinct sub-problems, the subsequent LSP search becomes faster.

The increase in efficiency is demonstrated in Table 3.1 through an example in the latter part of this section. Since aggregating points into clusters and identifying their bounds does not need high precision in LSP, the clustering criterion does not in fact need to be strong. But if a weak criterion is used for problems whose optimal points are at extreme corners of the initial cuboid, the imperfect cluster identification might result in missing the global optima. In practice, the probability of generating a sample point at an extreme boundary corner is almost nil for finite sampling schemes. A heuristic method has been developed to avoid this situation and is discussed in section 3.5.

The following two variable, unconstrained but bounded, problem with two global optima demonstrates the danger of premature clustering and the computational benefit of introducing cluster analysis into LSP.

$$\begin{aligned}&\text{Minimize } f(\mathbf{x}) = 100 - (x_1 + x_2 - 10)^2\\&\text{Subject to:}\\&\qquad 0 \le x_1 \le 10\\&\qquad 0 \le x_2 \le 10\end{aligned}$$

The true global optimum is $f(\mathbf{x}^*) = 0.0$ and the optimal points are located at $(0, 0)$ and $(10, 10)$, which are the diagonal corners of the initial cuboid.

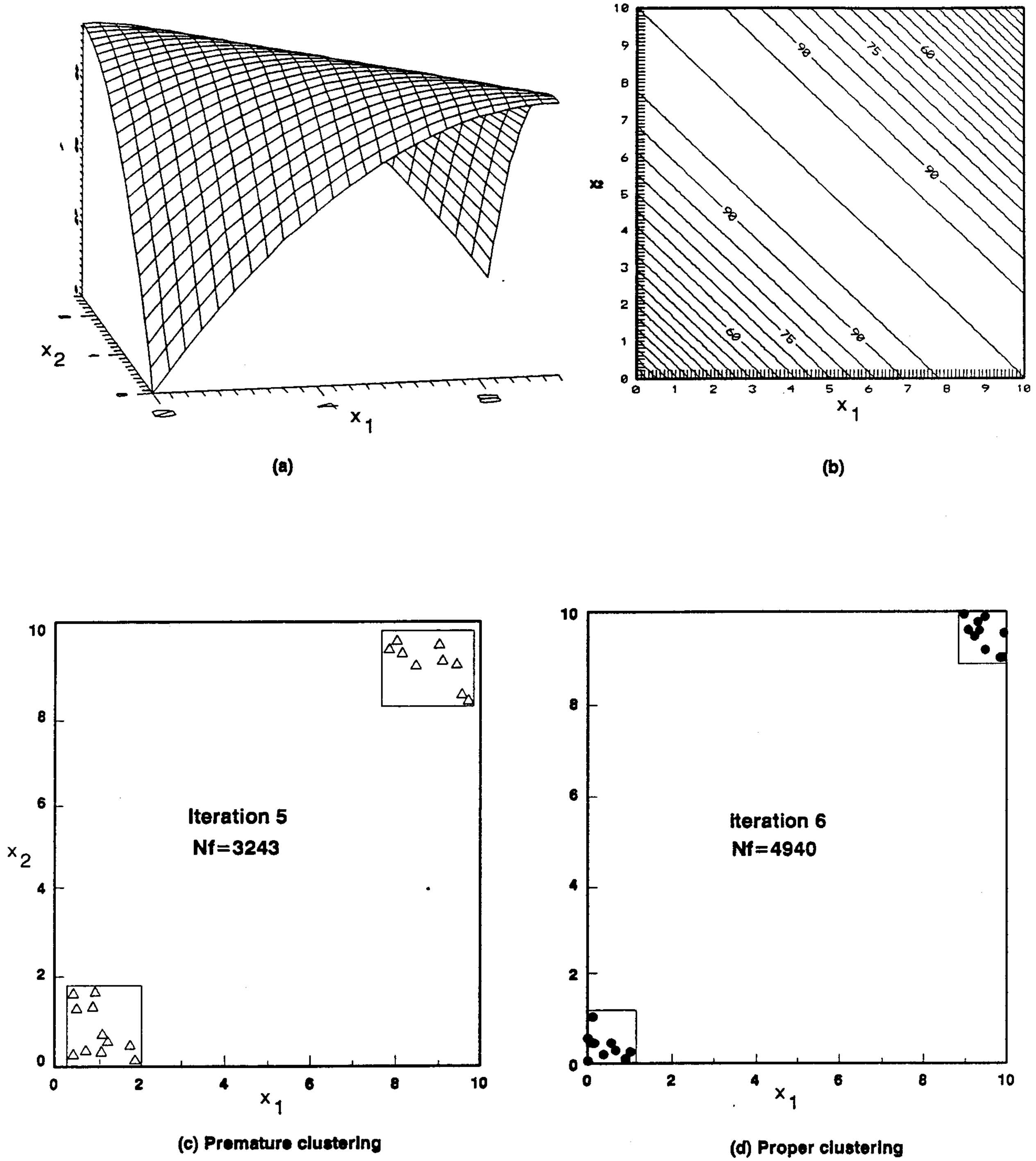
Figures 3.5(a) and 3.5(b) show this optimization problem in three and two dimensions respectively. Figures 3.5(c) and 3.5(d) show plots of the points in the level set in the $x_1 \sim x_2$ plane at consecutive iterations. In both cases the two small rectangles around the clusters of points inside the initial cuboid show the dimensions of the new sub-cuboids if the problem is to be partitioned into two sub-problems. Figure 3.5(c) demonstrates how a premature clustering can exclude the true global optimal points. However if clustering is performed at the next iteration, the global optimal points would be included in the new sub-cuboids, as shown in Figure 3.5(d). Premature clustering can be avoided by adopting a stronger clustering criterion.

Table 3.1 shows the number of function evaluations required to meet the LSP convergence criterion for the above example, with and without using cluster analysis method. These results are from a single LSP run for each case. The number of points maintained in the acceptance set for each sub-problem was the same as for the acceptance set without the cluster analysis.

Iteration	Number of function evaluations	
	without cluster analysis	with cluster analysis
5	3,024	3,243
6	4,236	4,940
13	1,420,092	—
14	3,686,648	—
15	8,018,142	—
37	—	6,550

Table 3.1: Reduced number of function evaluations due to cluster analysis methods

N_{keep} value of 30 was used for this example. This is higher than the recommended value of $N_{keep} = 20$ but appropriate in known multiple global optima cases. Without cluster analysis, this example had not met the convergence criterion $V_M \leq 0.0001$ at the 15^{th} iteration when the number of function evaluations was over $8.0 * 10^6$. In the second case, where the cluster analysis method was used, the problem was partitioned into two different sub-cuboids immediately after the sixth iteration. With $N_{keep} = 30$ in each of the two sub-problems, it took 37 iterations to meet the convergence criterion. The number of iterations cited is the sum of iterations expended before dividing the problem into smaller regions plus iterations expended to meet the convergence criterion for both sub-problems. Even though the number of LSP iterations is high when the problem is handled in two separate subregions, the total number of function evaluations is only 6,550.



(c) Premature clustering

(d) Proper clustering

Figure 3.5: Cluster analysis approach for a two variable problem.

As was experienced in a number of other examples, the high number of LSP iterations does not necessarily imply inefficiency of the overall search, but the cumulative number of function evaluations. In this instance partitioning of the problem permitted efficient point generation and lowered the number of function evaluations by a factor estimated at $\frac{1}{1200}$ or lower.

3.2 Penalty functions

Generally, feasible sample points are randomly generated with LSP for constrained optimization problems. In some cases the feasible region defined by the set of inequality constraints might be so small compared with the initial variable bounds that acceptable point generation requires an impractically large number of trial points. Similarly other problems might have equality constraints involving nonlinear equations which do not lead to reducing the number of variables by elimination. Such difficulties can be overcome by introducing penalties into the objective function.

One conventional penalty approach converts the constrained problem to an equivalent unconstrained problem so that those methods developed for unconstrained problems can then be applied [Bazaraa, 1979]. Suppose there is an optimization problem with only equality constraints, given as follows

$$\begin{aligned}
&\text{Minimize } f(\mathbf{x}) \\
&\text{Subject to: } h_i(\mathbf{x}) = 0 \qquad \text{for } i = 1, 2, .., l \\
&\qquad\qquad\quad \mathbf{x} \in E^n
\end{aligned}$$

This constrained problem can be transformed to an equivalent unconstrained form as

$$\begin{aligned}
&\text{Minimize } f(\mathbf{x}) + \beta(h_i(\mathbf{x}))^2 \\
&\text{Subject to: } \mathbf{x} \in E^n
\end{aligned}$$

where β is the penalty parameter, a non-negative large number.

A similar modification is made to problems with inequality constraints. If the original problem is

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}) \\ &\text{Subject to: } g_j(\mathbf{x}) \leq 0 \qquad \text{for } j = 1, 2, ..., m \\ &\qquad\qquad\quad \mathbf{x} \in E^n \end{aligned}$$

then the transformed problem would be expressed as

$$\begin{aligned} &\text{Minimize } [f(\mathbf{x}) + \beta \text{ maximum } \{0, g_j(\mathbf{x})\}] \\ &\text{Subject to: } \mathbf{x} \in E^n. \end{aligned}$$

For general optimization problems involving both equality and inequality constraints, given as

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}) \\ &\text{Subject to: } h_i(\mathbf{x}) = 0 \qquad \text{for } i = 1, 2, ..., l \\ &\qquad\qquad\quad g_j(\mathbf{x}) \leq 0 \qquad \text{for } j = 1, 2, ..., m \\ &\qquad\qquad\quad \mathbf{x} \in E^n \end{aligned}$$

a penalty function, $\alpha(\mathbf{x})$ is defined as

$$\alpha(\mathbf{x}) = \sum_{j=1}^{m} \phi[g_j(\mathbf{x})] + \sum_{i=1}^{l} \psi[h_i(\mathbf{x})] \tag{3.3}$$

where ϕ and ψ are functions satisfying

$$\begin{cases} \phi(y) = 0 & \text{if } y \leq 0 \\ \phi(y) > 0 & \text{otherwise} \end{cases}$$

$$\begin{cases} \psi(y) = 0 & \text{if y} = 0 \\ \psi(y) > 0 & \text{otherwise} \end{cases} \tag{3.4}$$

and y is a dummy variable representing $h_i(\mathbf{x})$ and $g_j(\mathbf{x})$ respectively.

Usually, ϕ and ψ are used in their power form as

$$\phi(y) = [max\{0, y\}]^r$$

$$\psi(y) = \mid y \mid^r$$

where r is a positive integer [Bazaraa, 1979].

The modified objective function, which is also called the auxiliary function, is then written

$$f(\mathbf{x}) + \beta\alpha(\mathbf{x}) \tag{3.5}$$

and the transformed problem would be finally expressed as

$$\text{Minimize } f(\mathbf{x}) + \beta\alpha(\mathbf{x})$$

$$\text{Subject to: } \mathbf{x} \in E^n$$

In the above expressions the penalty parameter β could be a fixed value or a sequence tending to infinity such that for each iteration k, $\beta_k \geq 0$ and $\beta_{k+1} > \beta_k$. With the quadratic penalty function, where $\alpha(\mathbf{x}) = \sum_{j=1}^{m} \phi[g_j(\mathbf{x})]^2 + \sum_{i=1}^{l} \psi[h_i(\mathbf{x})]^2$, the penalty parameter β in Equation 3.5 must become infinite in order to achieve complete convergence on $\mathbf{x}^*$ [Gill et al., 1989].

In the case of an exact penalty function, where $\alpha(\mathbf{x}) = \sum_{j=1}^{m} \phi \mid g_j(\mathbf{x}) \mid + \sum_{i=1}^{l} \psi \mid h_i(\mathbf{x}) \mid$, the penalty parameter β in equation 3.5 takes a constant value and $\beta \geq 0$. There is then a specific value of β so that $\mathbf{x}^*$, the unconstrained problem minimizer, is also the solution to the original constrained problem [Gill et al., 1989].

The meaning of "large value" in connection with specifying β is not very clear for a particular problem and selecting an inappropriate value of β can lead to computational difficulty. If β is too small, the penalty function may be unbounded below and as a consequence produce a solution point far from the feasible region. If β is too large then it induces an ill conditioned Hessian matrix which can imply slow convergence for many NLP algorithms. Matrix condition is defined as the ratio of its largest and smallest eigenvalues, and a matrix is ill conditioned if this ratio is large.

With large values of β, more emphasis is placed on feasibility and most procedures for unconstrained optimization will move quickly towards a feasible point. But it is possible for search termination to occur on a feasible point even though the point is far from optimum. This is especially true in the presence of nonlinear equality constraints [Bazaraa, 1979].

There is no clear indication that progressively increasing β has any advantage over a fixed large β in saving computational effort. But using several values of β, together with their corresponding solution points, to predict the solution point with the next β value is a classical search procedure. Interpolation is sometimes used to predict the next β from the past values [Gill et al., 1989].

3.2.1 Penalty parameters and LSP

Generally LSP responded best to the use of an exact penalty function as opposed to the quadratic penalty function. The gentle slopes induced in the objective function surface by exact penalty functions encourages LSP to form larger volume cuboids at every iteration when compared with the surface produced by the quadratic penalty function.

In practice, LSP starts finding the solution for the auxiliary function with a large β, and reduces the value of β by 10%-30% at every iteration. In this research a sequential reduction of the penalty parameter has been found to provide the greatest enhancement to the efficiency of the LSP search. This approach is exactly the opposite of the traditional NLP penalty function strategy, where the value of β increases as optimum is approached.

The advantage of sequentially reducing the penalty parameter is that the decrease brings about the acceptance of more infeasible points to the current cuboid. With the introduction of more infeasible points into the cuboid, the cuboid volume is larger than in the constant penalty parameter case. This in effect retards the rate of reduction of cuboid volume over consecutive iterations. Although slow cuboid reduction requires more iterations to reach convergence, its compensating advantage is that the newly generated points are retained for a greater number of iterations than with the constant penalty parameter strategy. Furthermore, because points farther out of the true feasible region are tolerated with a lower penalty parameter value, fewer function evaluations per acceptable point are required. The process of reducing β continues until $\alpha(\mathbf{x})$, and hence the penalty term $\beta\alpha(\mathbf{x})$, goes to zero. This must occur before convergence so that all of the unmodified constraints are satisfied and the original problem is finally solved.

Figure 3.6 demonstrates the effect on the level set boundaries of modifying the penalty parameter for a one variable problem with two constraints. The objective function is a straight line. The problem can be formulated as follows

$$Minf(x) = kx$$

Subject to:

$$x \geq 3$$

$$x \leq 3.1$$

where k is a constant.

A penalty is applied to just one of the constraints, i.e. to $x \geq 3$ only. Those lines designated as $P_1(\mathbf{x})$, $P_2(\mathbf{x})$, etc. indicate plots of the auxiliary functions, that is the objective function plus the penalty term, at the first, second, etc. iterations. The $c_1, c_2, etc.$ indicate the level set c-value established in the LSP search at the corresponding iterations.

The plot of the intersection points of the level set values and the corresponding

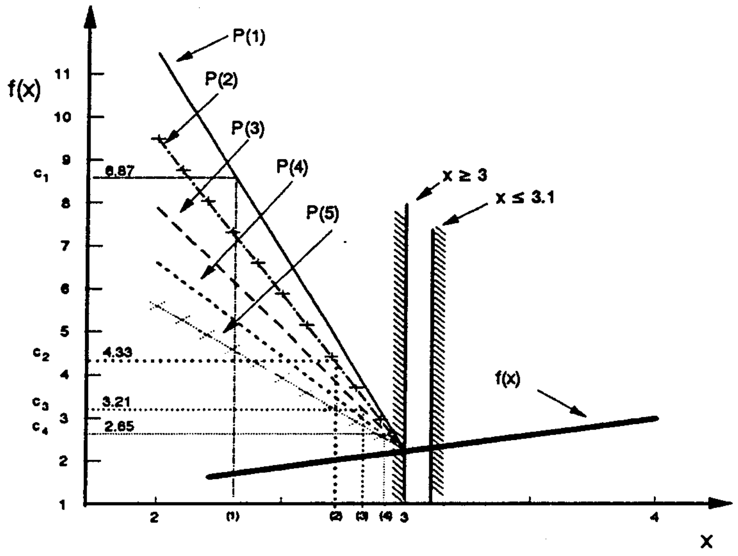


Figure 3.6: Modified penalty function.

auxiliary function values represents the objective function of an equivalent unconstrained problem.

The whole idea of reducing the penalty parameter is to generate new points more efficiently and at the same time smoothly contract the level set boundary so that the search eventually excludes any infeasible space. The full benefits of penalty modification may not be achieved unless a near optimum rate of reduction is used, otherwise the following problems may arise:

- If the reduction of β between consecutive iterations is large in the initial iteration, then the influence of the penalty term on the objective function must be small in the later iterations as β approaches its lower bound value. Then many infeasible points generated at the later iterations would be present in the acceptance set. In that case the search is not directed towards the feasible region and may converge on an infeasible point.

- A large reduction of β allows more infeasible points to remain in the level set for a longer period. If the problem is divided into subregions because of suspected multiple optima, it is possible that a subregion may contain only infeasible points. Then the search in that subregion would obviously lead to an infeasible solution point.

- A very small reduction of β does not allow many infeasible points into the acceptance set, hence introduces an inefficient search.

In order to overcome the problems mentioned, two measures are taken.

i) The reduction of β between consecutive iterations is kept small (between 0.1 and 0.25 of β).

ii) A lower bound for the parameter β is predetermined. The upper bound can be as high as required, though extremely high values would lead to an inefficient search.

Typical values of β and its reduction are given in Chapter 5, section 5.2.6.

3.3 Inequality and equality constraint relaxation

When the feasible region defined by the set of the constraints is very small or very narrow, or defined by equality constraints, the generation of acceptable points can be inefficient and even difficult. To overcome such difficulties and speed up the search, the constraint functions can be initially relaxed and the relaxation eliminated in the later iterations. The relaxation method may also be applied to the bounds of the original cuboid if it is felt that the bounds are too tight for efficient point generation. Penalties are associated with the relaxation of constraints, so that infeasible points are discouraged from staying in the acceptance set. As a result of the penalties introduced into the problem, the final solution is likely to be found in the feasible region.

The idea of constraint relaxation in this research was first investigated using slack variables as introduced in the reduced gradient method [Lasdon & Waren, 1982]. But the introduction of these new variables increased the dimensionality of the problem without any compensating gain in performance. Instead, the constraints are allowed to relax by simply adding (or subtracting) a constant term to the right hand side of the constraints. Equality constraints are replaced by two constraints with the constant added or subtracted from their right hand sides to form an interval. The constraint relaxation raises the volume of the feasible region which in turn raises the ratio of the feasible region volume to the cuboid volume, as a result the efficiency of generating acceptable points is increased. Figure 3.7(a) and (b) demonstrate how constraints are relaxed for equality and inequality constraints respectively for a two variable problem.

The number of function evaluations at the first iteration N_f^1, gives a general indication on how tight the constraints are. For a highly constrained problem, N_f^1 would be very large when compared with the value of $2 \times N_{keep}$ expected for an ideal unconstrained problem.

Setting the magnitude of the relaxation is a trial and error process. The constraints are relaxed by an initial value and N_f^1 is compared with $2 \times N_{keep}$. If the response to this initial relaxation does not seem to be promising, the search should be interrupted during or after the first iteration and the constraint relaxation adjusted until the user feels efficient point generation has been achieved. While this process is unquestionably heuristic, initial trial relaxation values which appeared to be "sensible" in the context of the problem being solved were often successful or were easily corrected from the feedback provided by N_f^1.

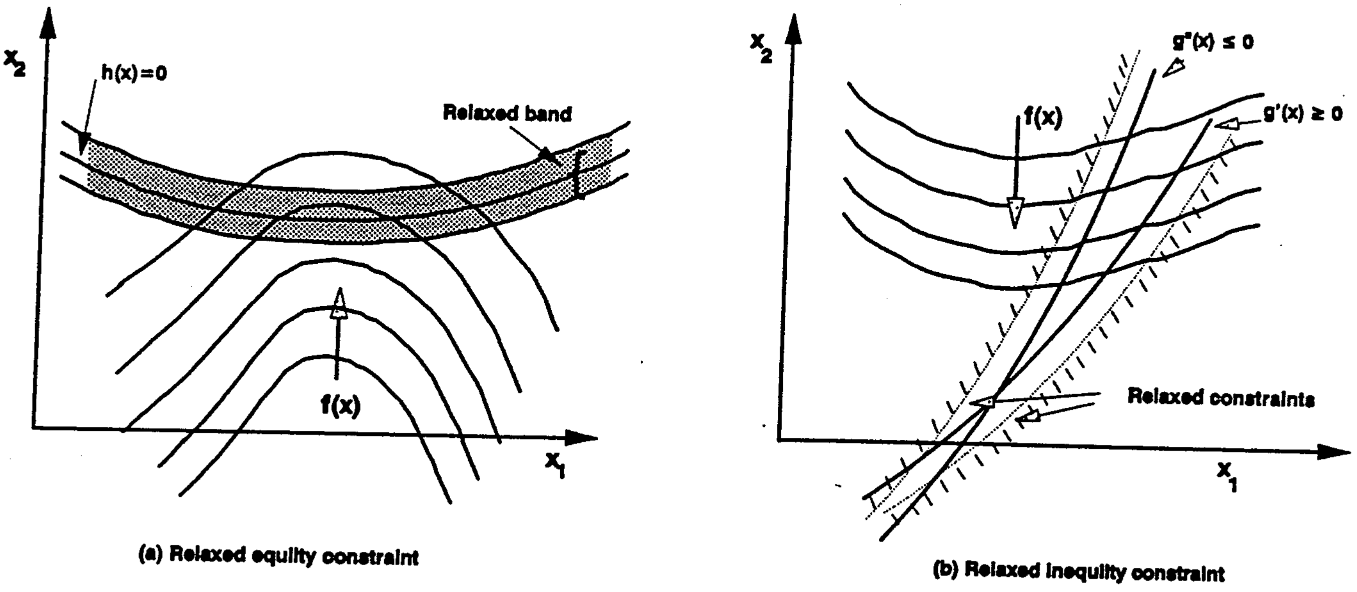


Figure 3.7: Relaxation of constraints.

3.3.1 Tightening of relaxed constraints

The initially relaxed constraints are progressively tightened so that the level set will ultimately be confined to the feasible region. Ideally the relaxed constraints will be tightened back to their original form before the convergence criterion is met. This is achieved by reducing the relaxation by a fixed fraction at each iteration. How far constraints should be tightened at each iteration is a compromise between the smallness of the feasible region and how fast the search is expected to achieve convergence. If the relaxation reduction per iteration is high, the search might contract into a very small region and the original difficulty will again be faced. On the other hand a very small reduction at each iteration can lead to a high number of iterations. Reducing the relaxation to a typical value of 0.75 of its value at the previous iteration would leave to a reduction to 0.01 (i.e. 0.75^{16}) of the initial relaxation after the 16^{th} iteration. The ideal value of the tightening parameter is related to the complexity of the constraint functions and the size of the feasible region but was found to be robust in the region of 0.75 for virtually all of the test problems where relaxation was appropriate.

3.4 Cuboid stretching

At any stage of the search, the acceptance set is expected to be a subset of the cuboid. The minimum and maximum coordinate values of the confirmed points are used to locate the cuboid sides. Due to limited sampling, the true bounds of the level set may actually extend beyond the observed points. Therefore, the confirmed points always tend to produce a biased under estimate of the bounds of the true level set. This bias is corrected by extending the cuboid faces outwards by some prescribed amount. This is referred to as cuboid stretching. The ideal size of this correction is not known in practice but over correction only incurs a small penalty in the efficiency of point generation, while under correction can have serious consequences of omitting a global solution. Overcorrection is preferred and a good heuristic estimate can be made.

The following expressions to stretch the bounds of the cuboid were proposed in [Chew & Zheng, 1988]. For a one dimension case, let ξ be a random variable on the interval $[a, b]$ with distribution function F, and F is assumed to be a uniform distribution.

Assume that t points are sampled, that is $\xi_1, \xi_2, ..., \xi_t$ are confirmed points in the acceptance set and t is the number of confirmed points. We consider to estimate the unbiased estimators of a and b from the sampled points. Suppose $a = 0$ and $b = 1$

$$B_l = \min \{\xi_1, \xi_2, ..., \xi_t\}$$

$$B_u = \max \{\xi_1, \xi_2, ..., \xi_t\}$$

Then the distribution functions ϕ_0 and ϕ_1 of B_l and B_u are respectively

$$\phi_0(y) = \begin{cases} 0 & y \leq 0 \\ 1 - [1 - F(y)]^t & 0 < y < 1 \\ 1 & y \geq 1 \end{cases}$$

$$\phi_1(y) = \begin{cases} 0 & y \leq 0 \\ [F(y)]^t & 0 < y < 1 \\ 1 & y \geq 1 \end{cases}$$

The expected mean values for each bound are given as

$E[B_l] = \int_0^1 (1 - F(y))^t \, dy$, which for uniformly distributed F is simplified to $\frac{1}{t+1}$ and

$E[B_u] = 1 - \int_0^1 (F(y))^t \, dy$, which is simplified to $\frac{t}{t+1}$.

The corrected new bounds are estimated as

$$B_l" = B_l - \frac{B_u - B_l}{t-1}$$

$$B_u" = B_u + \frac{B_u - B_l}{t-1}$$

where B_l" and B_u" are the unbiased estimators of the end points.

The consequence of this bias correction is that each cuboid side is stretched by $\frac{1}{n_x - 1}$ times the length of the original size at every iteration, where n_x is the number of confirmed points surviving in the acceptance set.

3.5 Skewness adjustment

Skewness adjustment is a combination of shift of cuboid location and cuboid stretching at the end of each iteration, and was also proposed in [Chew & Zheng, 1988]. It was confirmed in this research to be one of the most important heuristic devices for speeding up an LSP search and increasing the reliability of convergence on optimal points.

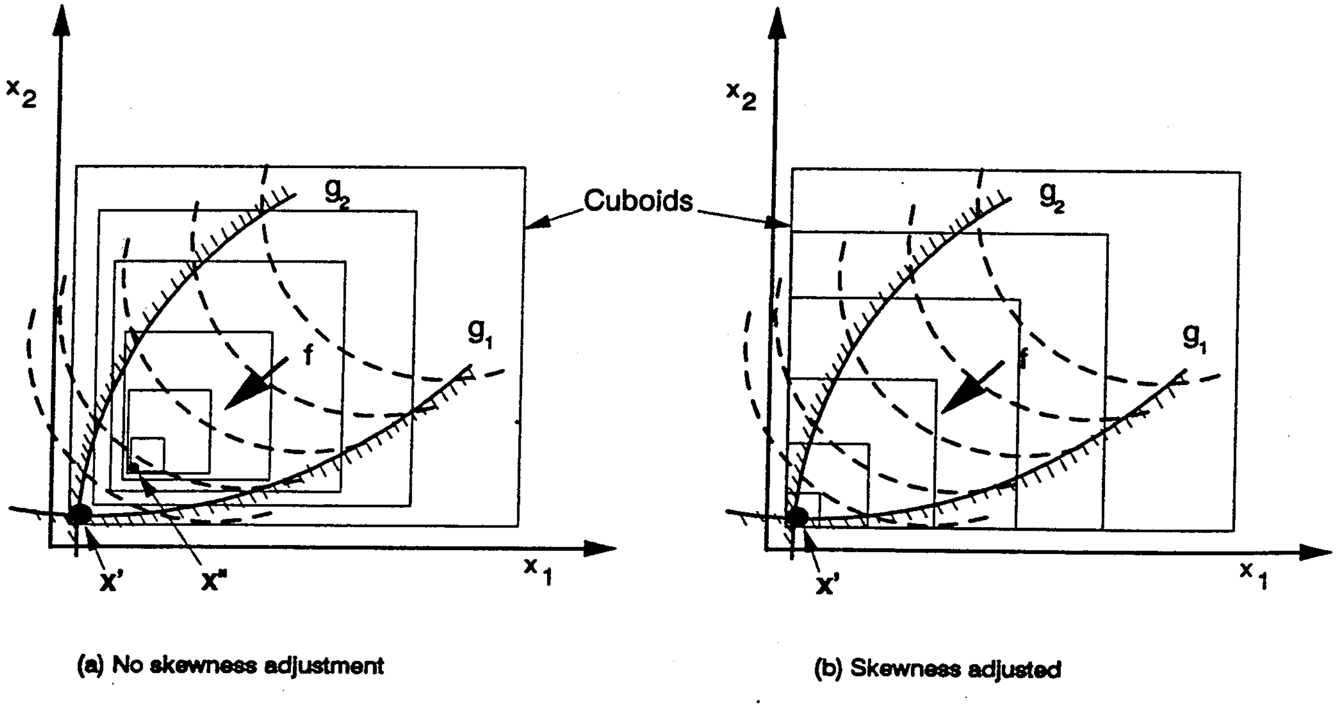
Since the adopted point generation scheme for new points in a cuboid utilizes a uniform random distribution along the axis of each variable, the possibility of missing an important point is high when the optimal point lies at a boundary, and even greater if it lies at a corner of the feasible region formed by multiple boundaries. The danger of missing an important boundary point can be lowered either by substantially raising N_{keep}, which would in turn increase the total number of function evaluations, or shifting the cuboid to ensure that it straddles the most promising region. When the acceptance set at any iteration is skewed, that is when the point with the lowest objective value is close to the boundary of the cuboid, an adjustment can be made to both the size and location of the cuboid so that the minimum point would be shifted towards the centre of the cuboid. A slight shift of the cuboid about the current minimum point helps more points to be generated around the current minimum point.

With a skewness adjustment the shift or translation of the cuboid is governed by the location of the single best confirmed point. This introduces a bias where more points are generated close to the current best point. But, since the magnitudes of the shift are controlled by certain user selected parameters, there is a risk that the detection of multiple optima can be undermined with imperfect skewness parameters.

The magnitude of the shift is somewhat arbitrary. Quantitatively a shift which is too large may lead to missing an optimal point, while a shift which is too small does not speed up the search nor direct the search towards the optimum significantly. Experience in this research suggests that there is a range of shift which results in very low chance of missing the global optimum and this range is in the order of 4% to 10% of each current cuboid side length.

Figure 3.8 demonstrates the principle of skewness adjustment in a two dimensional problem. There, the global optimum is located at x', while g_1 and g_2 are two constraint functions and the objective function and its gradient is represented by the $f(\mathbf{x})$ isoquants and the arrows. The rectangular boxes represent the cuboids at consecutive iterations, the largest at the earliest iteration and the smallest at the last iteration. In the iterative process of LSP, points generated close to the right corner of the cuboid would be discarded earlier since they would not fulfil the level set condition. The remaining points would tend to cluster towards the global optimum point. But because of the limited sample point generation and deficiency of sampling on the boundary itself, points around the lower left corner of each cuboid may not be sampled. An unadjusted cuboid at any iteration except the first excludes the lower left corner of the previous cuboid. Therefore the location of the current cuboid gets further away from the global optimum with each iteration. Consequently, convergence without skewness adjustment would occur at a non optimal point x''.

When the skewness adjustment is implemented, as shown in Figure 3.8(b), the cuboid is shifted towards the global optimum at every iteration and eventually convergence occurs at the true global optimum.



(a) No skewness adjustment

(b) Skewness adjusted

Figure 3.8: Shift of cuboid with skewness adjustment.

Skewness adjustment can be used for problems of any dimensionality and the adjustment is applied independently in each dimension. The procedure in each dimension is as follows.

Define the skewness adjustment δ at a single iteration as

$$\delta = \frac{(x' - B_l) - (B_u - x')}{(B_u - B_l)}$$

defining $X_r = B_u - B_l$, then

$$\delta = \frac{(2x' - (B_u + B_l)}{X_r}. \tag{3.6}$$

Where B_l and B_u are the lowest and highest observed points measured on this variable's axis, x' is the coordinate of the minimum confirmed solution point in the current cuboid and X_r is the length of the cuboid side in this dimension. Note that δ can be $+ve$ or $-ve$ in the interval $-1 \leq \delta \leq 1$.

Adopt three skew parameters: δ_0, which acts as a skewness activator threshold; and δ_1 and δ_2, which limit the size of the skewness adjustment. Their specific values will be discussed in Chapter 5, section 5.2.5, but must be bounded as follows.

$$0 \leq \delta_0 \leq 1$$

$$0 \leq \delta_2 \leq \delta_1 \geq 1$$

Adjustments to the cuboid bounds are made in relation to the size of the skew and the skew parameters δ and applied only if $\delta > \delta_0$ or $\delta < -\delta_0$ [Chew & Zheng, 1988].

$$\text{If } \delta \geq \delta_0 \begin{cases} B_l' = B_l + \delta\delta_2 X_r \\ B_u' = B_u + \delta\delta_1 X_r \end{cases}$$

$$\text{If} \delta \leq -\delta_0 \begin{cases} B_l' = B_l + \delta\delta_1 X_r \\ B_u' = B_u + \delta\delta_2 X_r \end{cases}$$

where B_l' and B_u' are the new lower and upper bounds of the cuboid.

Such an adjustment results in both stretching and shifting of the cuboid. The stretching of the cuboid Δ would be between $\delta_0(\delta_1 - \delta_2)$ and $(\delta_1 - \delta_2)$, whereas the shift of the centroid of the cuboid Λ ranges from $\delta_0\delta_1$ up to δ_1, depending upon the degree of the skew, that is

$$\delta_0(\delta_1 - \delta_2) \leq \Delta \leq (\delta_1 - \delta_2)$$

$$\delta_0\delta_1 \leq \Lambda \leq \delta_1. \tag{3.7}$$

Experience in this research revealed that skewness adjustment is not a good approach for discrete variable problems, especially when the sides of the cuboid become small relative to the discretization interval. For a discrete variable problem, the shift and stretch of the cuboid introduced by skewness adjustment is not just a small fraction of the cuboid dimensions. If the cuboid side length for an integer variable was say 3 units, then the minimum shift along that axis, will result in a shift of 33% of the total length. Such a significant alteration in the location of the cuboid can lead to the exclusion of important points.

3.6 Search domain modification

The efficiency of generating acceptable points in a cuboid declines as the volume ratio between the region occupied by the acceptance set and the cuboid gets smaller. One of the typical cases is when the $x_i \sim x_j$ scatter diagram of points in the acceptance set form a cluster around a diagonal of the current cuboid. In such cases generating points within the cuboid boundary is inefficient. Two different techniques were considered in this thesis to improve on this situation. The first technique was the rotation of the axes of the cuboid so that one of the sides of the new cuboid would be almost parallel to the line which passes through the centre of the cluster. This technique requires substantial extra computational effort since it involves transforming all points to the new rotated axes, constructing the cuboid and generating new points, and then transforming each new point back to the initial axes to check feasibility and compute objective function. The second technique, which is discussed in detail in this section uses a non-right angle parallelepiped as shown in Figure 3.9 instead of a cuboid as a search domain. The non-right angle parallelepiped is referred to here as a rhombohedron.

Rhombohedron is defined as a parallelepiped whose faces are parallelograms in which the angles are oblique and adjacent sides are unequal.

The use of a rhombohedron increases the search efficiency by the ratio of the volume of the (minimal) cuboid to the volume of the (minimal) rhombohedron.

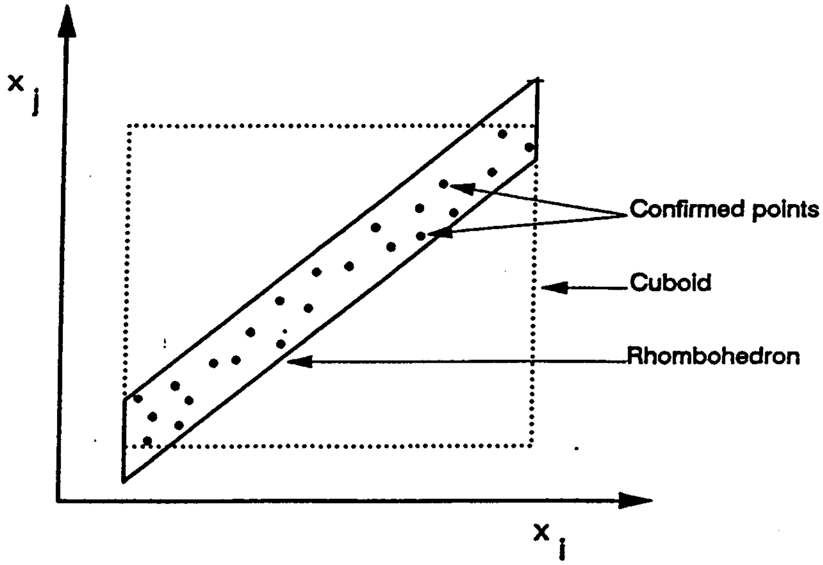


Figure 3.9: Contrast between cuboid and rhombohedron in two variable space.

The shift from the cuboid to the rhombohedron search domain can be instigated when the displayed scatter diagrams of the current acceptance set indicate a strong correlation between a pair of variables. Alternatively a statistical procedure to calculate a correlation matrix of all variables can be initiated at any stage of the search. If the absolute value of any of the correlation coefficients is greater than a pre-specified value, a linear regression analysis can be performed between the highly correlated variables. Choice of the dependant and independent variables is arbitrary.

Point generation in a rhombohedron is demonstrated in Figure 3.10. Initially a point is randomly generated on the independent variable axis within the current cuboid, for example point a in Figure 3.10. The corresponding value of the dependent variable is then calculated on the regression line, i.e. at point b in the figure. The generated point to be considered in the level set would be randomly chosen around the regression line plus or minus a pre-specified band width. That is to say the candidate point would be generated randomly on the line joining points c and d.

Finally the dependent variable value at point b is randomly perturbed up or down to produce a generated point. The limits of this perturbation were defined in this work by $\pm 3 \times se$, where se is the standard error of the estimate returned by the regression analysis. This ensured that the random generation of new points occurred within a rhombohedron large enough to virtually eliminate the risk of excluding any of the acceptance set.

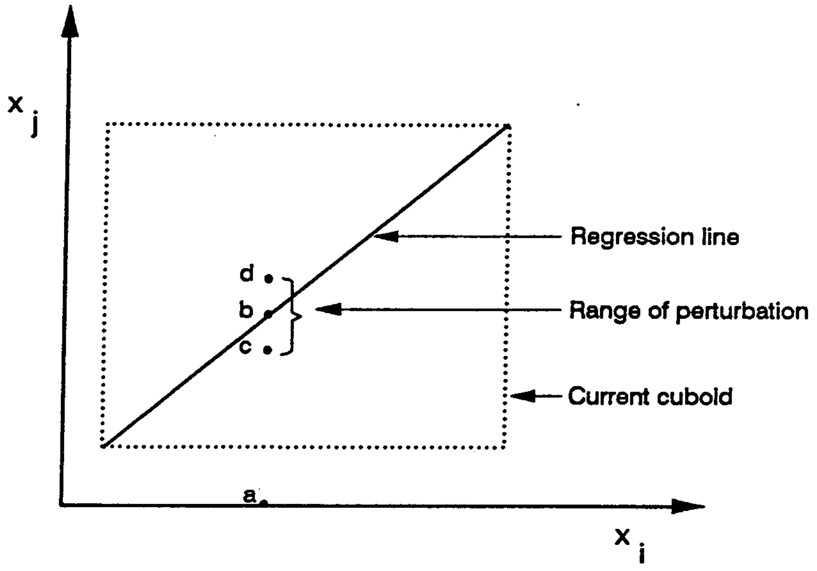


Figure 3.10: Point generation in a rhombohedron search domain.

Chapter 4

TEST PROBLEMS

The merits of an optimization method lie in its ability to solve a wide range of problems within the required bounds of accuracy and reliability. A newly developed method cannot be evaluated solely on its theoretical details but has to be tested numerically. Ideally the same standardized testing procedure can be applied to all optimization methods but such a procedure has not yet evolved. The practical alternative is to apply the optimization method to a wide variety of suitably challenging test problems. Another equally important alternative is to use test problems which provide insights into the strengths and weaknesses of a particular method.

A major component of this research effort was expended on exploring the viability of LSP in solving both mathematical and practical engineering problems. This involved solving some practical engineering optimization problems where the dimensions and the non linearities of the problem challenge existing optimization methods. In addition, a new test problem, designed specifically to explore perceived weaknesses with LSP, was developed.

About 200 nonlinear optimization problems with published solutions were collected from the mathematical literature. A selection of 78 of the more challenging of the 200 mathematical test problems solved with LSP are listed in Appendix A. Solutions cited in the literature and the LSP solution(s) are given for each problem. The problems documented in the Appendix were chosen for two major reasons. First, for the challenge they have presented to many of the existing NLP methods. Second, because they were found to provide useful insights into LSP performance.

A second group of test problems consists solely of engineering problems. The problems in this group are published practical problems known to be difficult for many NLP methods and of interest to engineers. Some of these publications present specific NLP approaches developed to solve just one of these problems.

A third group of test problems involves parameter fitting for a specified model. The values of the model parameters are established by optimizing various nonlinear best fit criteria. This is an area of application where LSP provides a unique and potentially attractive perspective. Some of these models are used to demonstrate the capacity of LSP to refine estimates of parameters previously established with nonlinear regression methods.

In every one of the problems, the problem formulation used with LSP was tested for numerical agreement with the published solutions. Except for very few cases, all the solutions in the source literature were verified. Any deviation from the original solution is reported and the discrepancy discussed.

Whenever possible, the widest possible variety of problems are addressed to explore the robustness of LSP. However, the main difficulty experienced was in finding suitably challenging mathematical and engineering test problems. Most of the published test problems uncovered in the literature were felt to be inadequate for testing all qualities of LSP. In response, one new, parametrically adjustable, mathematical test problem was developed during this research. Significantly, many of the existing NLP methods fail to solve this new test problem unless a favourable starting point is selected. Details of this new test problem are provided in section 4.2.1 of this chapter.

The algorithm used in addressing all of the test problems in this research is presented in Figure 4.1 as a flow chart. The algorithm performs all tasks automatically except those indicated as optional in the figure. The optional tasks can be made functional at

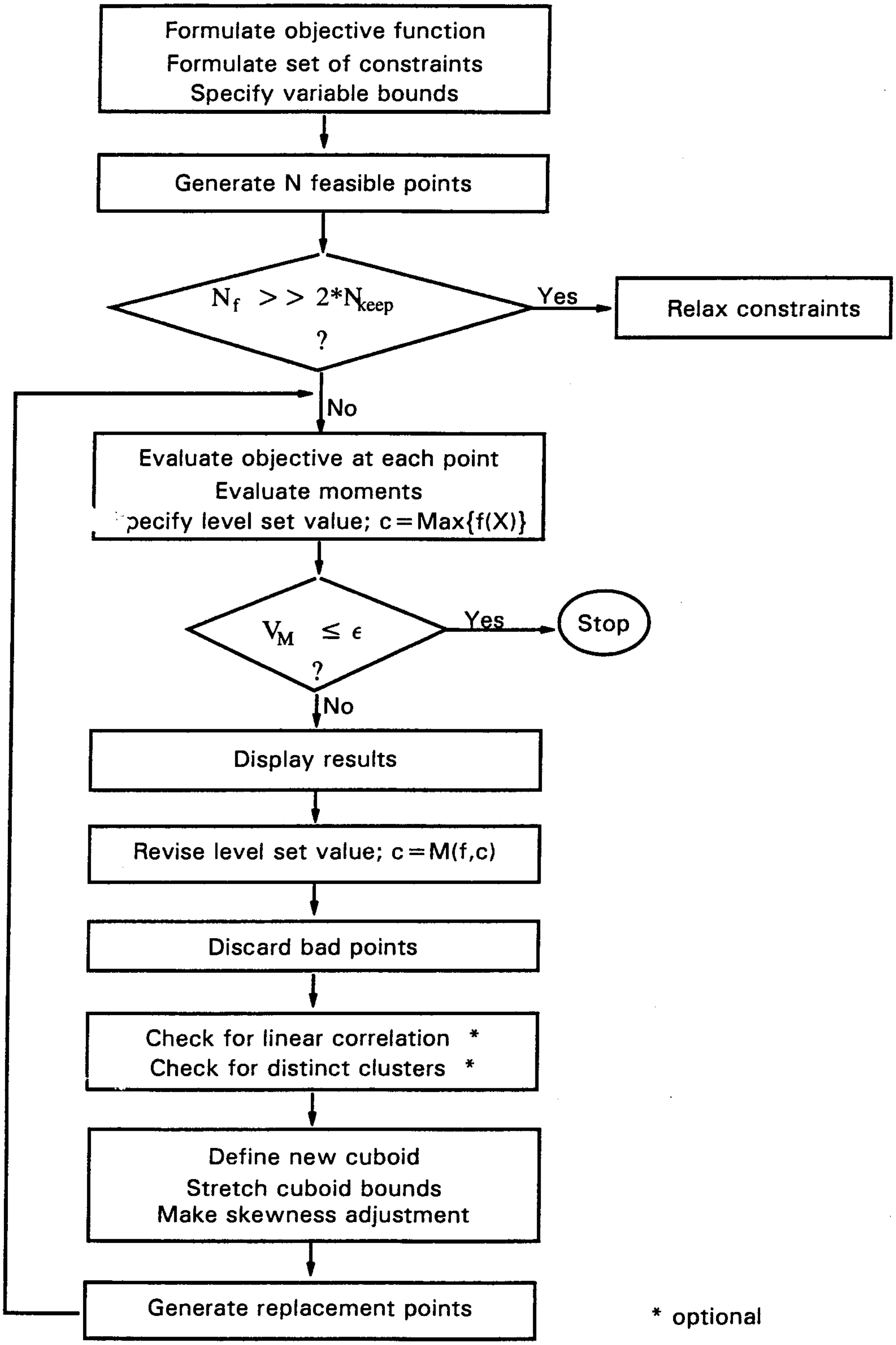

Formulate objective function
Formulate set of constraints
Specify variable bounds
Generate N feasible points
$N_f >> 2*N_{keep}$?
Yes
Relax constraints
No
Evaluate objective at each point
Evaluate moments
Specify level set value; c=Max{f(X)}
$V_M \leq \epsilon$?
Yes
Stop
No
Display results
Revise level set value; c=M(f,c)
Discard bad points
Check for linear correlation *
Check for distinct clusters *
Define new cuboid
Stretch cuboid bounds
Make skewness adjustment
Generate replacement points
* optional

Figure 4.1: The LSP algorithm as implemented in this thesis.

any stage of the search by the user as the need arises or may be triggered automatically when default conditions occur. These optional tasks can be computationally intensive and are often not crucial to finding a solution.

4.1 Coding of mathematical test problems

The mathematical problems documented in Appendix A are coded in such a way that the number of variables and constraints involved can be easily recognized from the code. Each problem coding takes the form

[Problem $A - B - C - D$, Location in thesis]

where

A designates the number of variables in the problem

B designates the index number within that particular dimensionality group

C designates the number of equality constraints

D designates the number of inequality constraints.

For instance, [Problem 3-7-2-1, Appendix A] would mean the seventh problem within the set of 3 variable problems, and the problem has two equality and one inequality constraint. It is documented in Appendix A of this thesis.

Most practical optimization problems assume upper and lower bounds on each variable but these bounds are not counted here as constraints. The only instances when bounds are considered as inequality constraints are where the source literature dictates their use for specific reasons.

The original source for each problem is referenced.

4.2 Mathematical problems

It is not easy to find test problems which can reveal all the strengths and weaknesses of a single NLP method. Recognizing this difficulty, a wide range of solved test problems were chosen from different journals and books. The more interesting mathematical test problems solved using LSP are documented in Appendix A. Results given in the literature and those obtained with LSP are documented along with each problem. The distribution of these problems according to the number of variables is summarized in Table 4.1.

Number of variables	Unconstrained problems	Constrained problems	Total
1	5	1	6
2	8	15	23
3	5	12	17
4	3	9	12
5	-	4	4
6	-	6	6
7	-	3	3
8	-	3	3
9	-	2	2
15	-	1	1
20	1	-	1

Table 4.1: Summary of test problems reported in Appendix A

About two thirds of the test problems documented in Appendix A involve up to 4 variables. Their specific characteristics are intended to challenge NLP methods apart from the issue of dimensionality. Furthermore, these low dimensional problems are relatively easy to visualize. The space defined by the set of constraints and the surface of the objective function can often be interpreted and clarified in simple 2 and 3 dimensional graphs. Such graphs help visualization of the search process and its response to the difficulties presented by the problem. The experience gained with the low dimensional problems can often be exploited in higher dimensional problems.

Comparison of LSP results with the solutions given in the literature showed that, in about 95% of the cases, the LSP results confirmed those cited in the literature. In about 5% of the cases, LSP gave better results than have been previously published. Two types of improvements were observed. One was the identification of multiple solutions with LSP when the source literature gives only some of these solutions. LSP generally found the extra solution point(s) on a single run. For example, only two solution points were reported in the literature for [Problem 2-11-0-0, Appendix A], but LSP found two additional global optimal points. A second type of improvement was observed when LSP converged at a different solution point from the one given in the literature and with an improved objective function value. Such improvements were rare since test problems often favour the NLP methods that they were developed to demonstrate. Examples of test problems with improved solutions are: [Problem 4-8-0-3]; [Problem 4-10-1-2]; and [Problem 7-2-0-3, Appendix A].

There were several instances where, when the solution cited in the literature was tested, the constraints had been significantly violated, although this was not acknowledged. In other instances, the claimed optimal value of the objective function did not agree with the calculated value at the solution point cited. For example, the constraints of [Problem 9-2-6-0, Appendix A] were significantly violated at the optimal solution point in the published solution. Similarly the optimal objective function value cited in the literature for [Problem 1-4-0-0, Appendix A] can not be achieved at the given solution points. LSP found multiple global optimal points which produce a superior objective function value for the same problem. When corrections are made to such incorrect solutions in this thesis they are not reported as improvements, but as errors in the source literature.

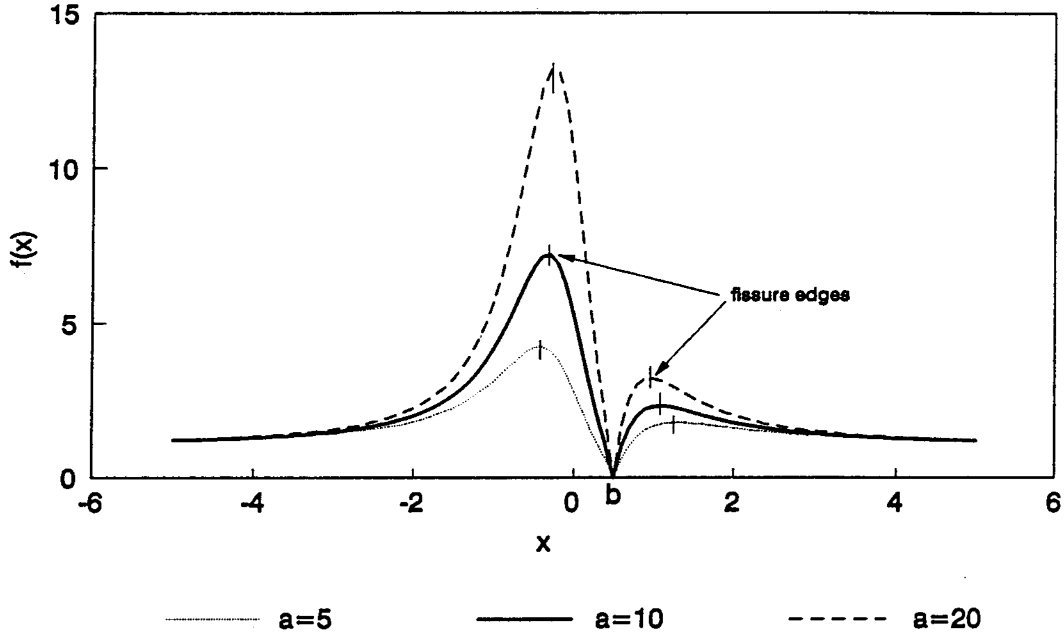
Some special problems were encountered by LSP when solving some of the test problems and are reported. Either the results were inferior to those mentioned in the literature or the number of function evaluations was considerably higher. An interesting example is [Problem 2-12-0-4, Appendix A], discussed in detail in Chapter 5, section 5.5, where the cause for the difficulty is associated with the geometry of the feasible region.

4.2.1 The Road Runner function

This new test problem was designed to test global optimization methods in general as well as to specifically challenge LSP and explore its weaknesses. It produces an objective function surface which has an easily missed fissure containing the global optimum and a topology which can cause fragmentation and dispersion of the level set. The scale of its prominent features and the location of the global optimum point are controlled by two parameters. The first parameter a, in Equation 4.1 adjusts the size of the fissure on the objective function surface. It controls how wide and how deep the fissure can be. Here, fissure width is defined as the distance between the fissure edges on each variable axis. Depth of the fissure is defined as the difference between the maximum and minimum objective function values. The second parameter b controls the location of the global minimum. Figure 4.2 shows the influence of parameters a and b on the objective function, particularly at the edges of the fissure region.

This test problem has been named *"The Road Runner function"*, after a popular film cartoon which is set in terrain which resembles the extreme topography of the objective function surface produced. The problem is a challenge for all local optimization methods unless a favourable starting point is chosen. It is also a challenge for LSP since it is particularly sensitive to sparse sampling of the level set. The additive structure of this function allows it to be easily extended to any number of dimensions while retaining the same critical features.

The general n dimensional formulation of the Road Runner function is given in Equation 4.1



a	x @ fissure edges	f(x) at edges	Fissure width	Fissure depth
5	-0.43114, 1.27782	4.22497, 1.76965	1.70894	4.22497
10	*-0.32818, 1.08210*	*7.24521, 2.31046*	*1.41028*	*7.24521*
20	-0.26295, 0.95756	13.24932, 3.21143	1.22051	13.24932

Figure 4.2: The Road Runner function in one dimension to show the influence of a.

$$f(\mathrm{x}) = \sum_{i=1}^{n} \{(x_i - b)^2 + a \mid x_i - b \mid\}^{\frac{1}{x_i^2+1}} \tag{4.1}$$

where a and b are parameters which directly adjust the scale of the critical topographic features and n is the required dimensionality for x. This function will always have its single global minimum at $x_i = b$; $i = 1, ..., n$ where $f(\mathrm{x}) = zero$.

The function becomes a unimodal problem if the search domain is restricted to the small region around the global optimum. The search domain has to be bigger than this minimal region to observe all of the features of the function. Thus, the bounds on each variable should be wide enough to enclose regions beyond the fissure width. Multiple local optimal points develop with this bigger search domain and their number increases exponentially with dimensionality. The total number of local optimal points is 3, 9 and 27 for one, two and three dimensional functions respectively. In general, there are 3^n local optima for an n dimensional function. Only one of the local points is the global optimum for each case. Two and three dimensional plots of the two dimensional version of this function are given in Figure 4.3 using the values $a = 10$ and $b = 0.5$. In this case the function is

$$f(\mathbf{x}) = \{(x_1 - .5)^2 + 10 \mid x_1 - .5 \mid\}^{\frac{1}{x_1^2+1}} + \{(x_2 - .5)^2 + 10 \mid x_2 - .5 \mid\}^{\frac{1}{x_2^2+1}}.$$

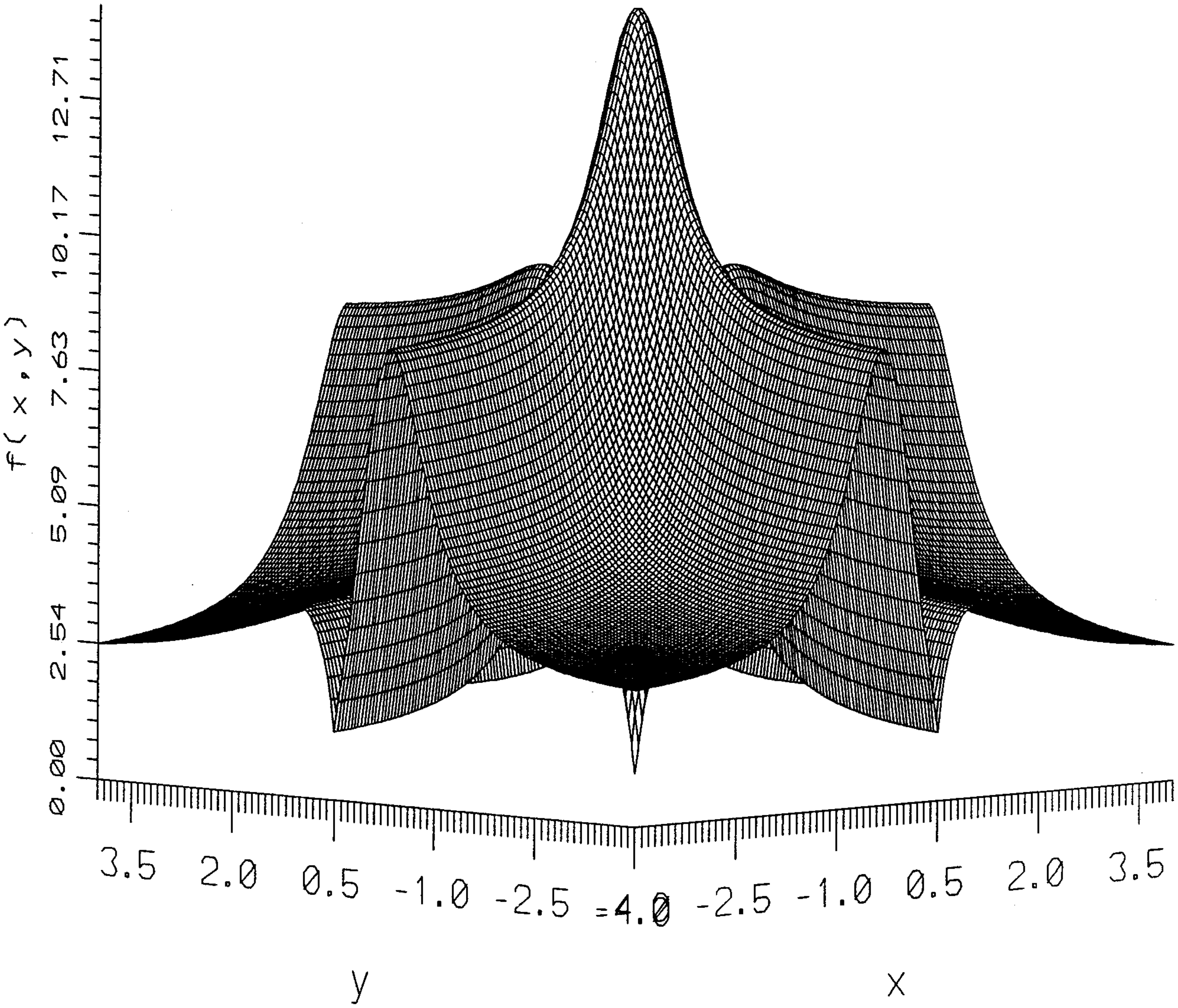
Figure 4.3(a) shows that the function has two sets of valleys. Each set consists of four valleys having similar depths. The depth of each valley increases as it goes farther away from the global optimum. The 4 valleys which are parallel to the axes have all equal depths. The other 4 valleys which are diagonal to the axes have also similar depths but different from the first four. The local optimal points appear at the intersection of the valleys and at the boundaries formed by the bounds of each variable. The global point is at $\mathbf{x} = (0.5, 0.5)$, at the centre of a very narrow fissure, where $f(\mathbf{x})$ achieves its global minimum of zero.

Figure 4.3(b) clearly reveals the influence of starting point location on the solutions obtained with gradient optimization methods. Gradient based methods always converged at a local optimum unless they started at a point very close to the global optimum. Figure 4.3(b) shows 36 arbitrarily chosen starting points, and the corresponding optimal values obtained using gradient optimization methods. The results shown in Figure 4.3(b) were found using the popular NLP methods such as FMIN, GRG2 and GAMS which uses

MINOS. Searches starting at points with circular marks converge at the true global optimum. Those searches which started at points other than those identified by circles converged at local optimal points. The optimal objective function values mentioned in the figure are for the purely unconstrained cases, i.e. where there were no bounds on the variables.

As was intended this problem challenges LSP, particularly when the initial variable bounds are wide and the value of N_{keep} is low. With a large cuboid, the ratio of the fissure volume to the cuboid volume becomes very small, so points generated in the early stage of the search are unlikely to fall within the fissure. Because of the nature of the objective function surface, points far away from the fissure produce better objective function values than those that are close to the fissure but not actually within it. As the level set value-c approaches 1.0, there is little size reduction in the cuboid with successive iterations, but the level set value-c continues to improve at each iteration. This situation is the least advantageous for LSP because the ratio between acceptance region and cuboid volumes gets very small where the search becomes inefficient.

If an initial cuboid with one of its corners very close to the global optimum is used, the chances of generating sample points within the fissure are much lower than when the global solution is more centrally located in the cuboid. The reason is that, in regions close to the global optimum, points produce very high objective function values and are quickly rejected from the level set. Therefore, unless points with low objective function value are sampled on either side of the global minimum, the region close to the global point can be excluded from the current cuboid at an early stage of the search. The situation gets worse with low N_{keep}. Consequently the search terminates at a point other than the global optimum. The reliability of LSP can always be improved by raising N_{keep}. With a higher number of points, the chances of generating points in the fissure is increased at the early stage of the search.



f(x , y)
0.00
2.54
5.09
7.63
10.17
12.71
3.5
2.0
0.5
-1.0
-2.5
-4.0
-2.5
-1.0
0.5
2.0
3.5
y
x

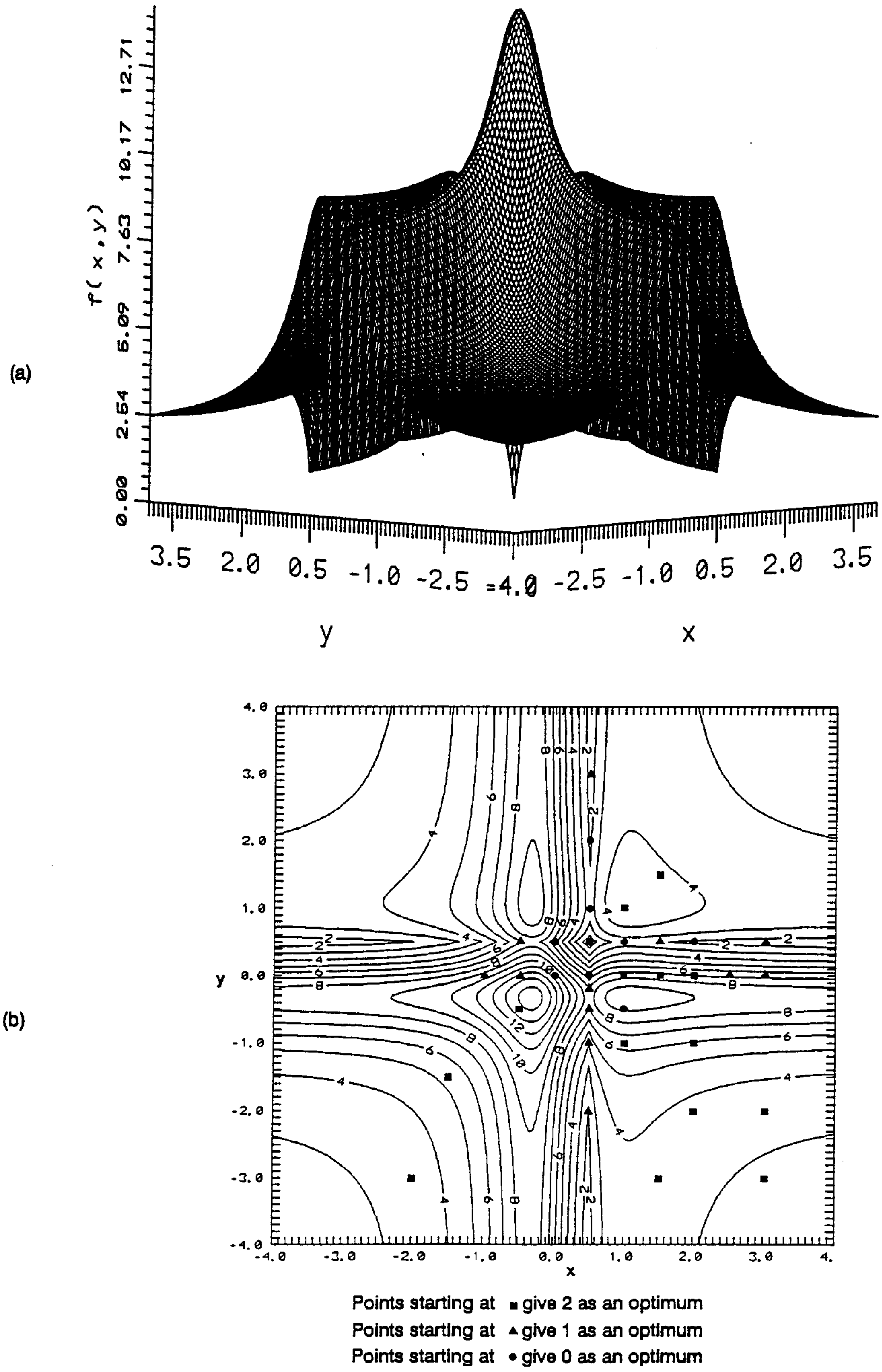


Figure 4.3: The Road Runner function in two dimensions.

The Road Runner function was solved using LSP for various number of variables. The result for the 2 dimensional function with $N_{keep} = 20$, convergence criterion $V_M \leq 1*10^{-5}$, $a = 10$ and $b = 0.5$ and the initial bounds $-4 \leq x_1, x_2 \leq 4$, was

$$f(\mathbf{x}^*) = 4.0E - 4$$

$$\text{at } \mathbf{x}^* = (0.5000, 0.5000).$$

The number of function evaluations expended to meet this convergence criterion was 4,230.

For a two dimensional Road Runner function there are 8 valleys stretching radially away from the global optimum. The valleys are equally spread in the search domain and the minimum point in each valley occurs at the boundaries formed by the variable bounds. Such a distribution of the local minimal points is believed to introduce the greatest inefficiency in set based search schemes and in particular to LSP. Due to the orientation and spatial distribution of the valleys, the domain modifications discussed in Chapter 3, section 3.6, do not improve the LSP efficiency as there is no search domain which can enclose all confirmed points more efficiently than the cuboid. The function with rotated axes has features similar to the original one, so it does not bring any computational gain whatsoever.

4.3 Engineering test problems

For the most part real world engineering test problems for optimization methods deal with the search for superior designs. Such problems often involve both continuous and discrete variables. In addition the functions involved may be discontinuous and non differentiable. Most of the established nonlinear optimization methods are designed to handle only continuous and differentiable functions and are local rather than global optimizers. These restrictions may account for some of the resistance by civil engineering practitioners to the application of these existing optimization methods for solving practical problems. There are very few well documented civil engineering test problems in the literature and this scarcity represented one of the major difficulties in testing LSP's capabilities in that field of specialization. In addition, those examples which do exist in civil engineering were not developed to isolate and test specific qualities of optimization methods, other than perhaps their high dimensionality capabilities.

Floudas and Pardalos [Floudas & Pardalos, 1990] documented a collection of engineering test problems for constrained global optimization algorithms. A few of the more interesting problems have been selected from this collection. Significantly, some of the results given in the literature are not global solutions. For instance, solutions to the 4-variable problem [Problem 4-10-1-2, Appendix A] and the six variable problem [Problem 6-5-3-12, Appendix A] are far from the optimums which are cited on page 29 of the book entitled *A collection of test problems for constrained global optimization algorithms*, [Floudas & Pardalos, 1990]. In some cases the reported optimum value of the objective function is different from that obtained by using the values in the solution vector. The 4-variable problem already mentioned [Problem 4-10-1-2, Appendix A] is an example of such inconsistency. In other cases the optimal points given for some test problems violate the constraints. Examples of these are the 7-variable problem on page 153 and the 27 variable on page 67 in the same mentioned book.

Because of such inconsistencies, many test problems which at first sight seem to be well documented were not of any value for LSP testing purposes. In spite of having to reject many, about 20 credible published engineering problems were found and used for test purposes. These problems varied from 3 to 38 variables, and included both discrete and continuous variables. They cover applications of optimization to pipe network design, structural truss design, treatment basin design, heat exchange design, air pollution control design, irrigation system design and hydrological model parameter determination. Details of 11 of the most challenging problems, including their solutions, are given in the next subsections.

In most cases the LSP results were comparable to the solutions given in the literature, but again some improvements were observed. For instance, a 12.91% improvement on the published objective function value was found with the pipe network design [Problem 32-1-40-34, section 4.3.2]. Another improvement was obtained with a better fit for the hydrologic model parameters which, from a hydrologist's point of view, produced a far more reasonable unit hydrograph for the UH ordinate determination problem [Problem 19-1-21-0, section 4.3.5].

4.3.1 Basin design [Problem 3-16-1-3]

Source: F. Douglas Shields Jr., and Edward L. Thackston, "Designing Treatment Basin Dimensions to Reduce Cost", ASCE, Journal of Environmental Engineering Vol 117 No 3 May/June 1991.

Method of optimization used in the source literature: Choosing the best curve out of a series of Length/Width versus cost plots drawn separately for different number of spurs.

Problem formulation:

A rectangular shallow treatment basin is to be designed with internal spur dikes, as shown in Figure 4.4. The designer is expected to choose the basin length L, the basin width W at water surface, the number of dikes N and the ratio of spur dike length to basin length r. The choices have to satisfy the governing constraints at a minimum cost. The geometric and hydrologic relationships are given as follows.

Geometric relationships

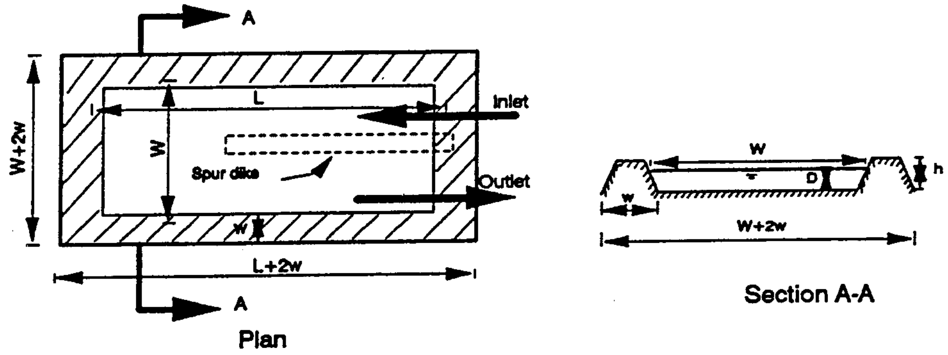


Figure 4.4: Treatment basin.

$$A_w = LW - NrLW_s$$

$$V = LWD - NrLV_u$$

$$\frac{L_f}{W_f} = rL(N+1) \div \frac{W}{(N+1)}$$

or

$$\frac{L_f}{W_f} = (\frac{L}{W})r(N+1)^2$$

where

A_w is the water surface area calculated as the total basin area at the water surface elevation minus the area occupied by the spur dikes

V is the basin volume, which is calculated as the total basin volume assuming that the volume of side slopes below water is negligible, minus the volume of the spur dikes

L is the length of basin

W is the width of basin

W_s is the width of spur dike at the water surface

N is the number of spur dikes

r is the ratio of spur dike length to basin length

D is the depth of water

V_u is the unit volume of spur dike below the water line

L_f is the flow length

W_f is the flow width

Hydraulic relationships

$$HE = \frac{t}{T}$$

$$T = \frac{V}{Q}$$

where

HE = hydraulic efficiency

t = mean hydraulic retention time

T = volumetric residence time

Q = mean flow rate

For a given Q, the value of T can be adjusted by changing the basin geometry. Experimentally the hydraulic efficiency is found to be related to the flow length, L_f, width, W_f, ratio, expressed as

$$HE = 0.84[1 - exp(0.59\frac{L_f}{W_f})]$$

Therefore

$$\frac{V}{Q} = \frac{t}{HE}$$

or

$$\frac{LWD - NrLV_u}{Q} = \frac{t}{0.84\{1 - exp[-0.59(\frac{L}{W})r(N+1)^2]\}}.$$

For a small volume of the spur dikes, $NrLV_u$, the basin volume is approximately LWD, then

$$LW = \frac{tQ}{0.84D\{1 - exp[-0.59(\frac{L}{W})r(N+1)^2]\}}.$$

Normally t, Q, and D are fixed by design constraints so

$$LW = \frac{a}{\{1 - exp[-0.59(\frac{L}{W})r(N+1)^2]\}}$$

where $a = tQ/0.84D$.

The basin cost includes the costs of land, dikes, inlet and outlet structures and access roads. If f is the ratio of the unit cost per foot of spur dike to the unit cost of the perimeter dike, and if all other costs are negligible relative to the cost of the dikes and land, then the basin cost may be approximated by

$$cost = [(2L + 2W + 4w) + fNrL]c_d + [(L + 2w)(W + 2w)]c_l$$

where w is the base width of the perimeter dike, c_d is the cost per unit length of perimeter dike and c_l is the cost per unit area of land.

If Q and t are known and the dike design has determined D, r and w then the only remaining geometric variables to be chosen are L, W and N.

Two cases are considered:

Case a) No constraint on land.

Case b) Land is a constraint, width is fixed to 500 ft.

The optimization problem has a nonlinear objective function involving three decision variables. One of the decision variables, N, takes integer values only. The problem has one nonlinear equality constraint. The formulation is

$$\text{Minimize } cost = [(2L + 2W + 4w) + fNrL]c_d + [(L + 2w)(W + 2w)]c_l$$

Subject to:

$$LW = \frac{a}{\{1 - exp[-0.59(\frac{L}{W})r(N+1)^2]\}}$$
$$W, L, N \geq 0$$

The predetermined constant values used and physical conditions of the basin are given as follows.

- The basin is rectangular, and spur dikes are longitudinal, with $r = 0.85$.

- Perimeter dikes have crown widths of 10 ft, heights of 7 ft and side slopes of 1:3, their unit volume is 217 cu ft/ft, with a base width, w, of 52 ft.

- Spur dikes have crown widths of 2 ft, heights of 7 ft, and side slopes of 1:2. Width at the water line, W_s, is 10 ft, and the unit volume below the water line is 100 cu ft/ft.

- Basin depth is 5 ft.

- Average flow rate, Q, is 27 cu ft/sec.

- The required residence time is 45 hours.

- The unit cost of the perimeter dikes c_d, is \$20/ft.

- Unit cost of spur dikes is \$10/ft, so $f = 0.5$.

- The unit cost of land is \$1000/acre, or \$0.23/sq ft.

The optimal solution cited in the literature and LSP's output are given in Table 4.2.

Observations:

- It took 34 minutes to find the optimal solution for Case a of Table 4.2, and only 4 seconds for Case b with LSP on a 80386-33Mhz microcomputer.

	Case a		Case b	
	Literature	LSP	Literature	LSP
N	1	1	0	1
$L(ft)$	1245	1266	2310	2083.3
$W(ft)$	890	869	500	500
Cost (1000 $)	131	131	150	155.5
Error (ft^2)	368.58	0.67	314.57	0.26
Error is the violation of constraints				

Table 4.2: Optimal results for the basin design problem

- LSP found that the problem has multiple global optima for case (a). A number of near optimal points were also found. The sizes of the final cuboid on the length variable L side ranged between 1,267 and 1,303 ft. On the width side it ranged between 853 and 869 ft whereas the number of spur dikes was always one. The value of the cost within the final acceptance set varied only by about 0.15%.
- When the results cited in the literature are used, some constraints are violated significantly.

4.3.2 Pipe network design [Problem 32-1-40-34]

Source:

1. E. Alperovits and U. Shamir, "Design of optimal water distribution systems", Water resources research, Vol 13, No 6, pp 885-900, 1977.
2. O. Fujiwara, B. Jenchaimahakoon, N. C. Edirisinghe, "A modified Linear programming gradient method for optimal design of looped water distribution networks", Water Resources Research, Vol 23, No 6, pp 977-982, June 1987.
3. G. V. Loganathan and J. J. Greene, "Global approaches for the nonconvex optimization of pipe networks", Water Management in the 90's, 1993.

Methods of optimization used in the source literature: LPG, Linear programming gradient in [Alperovits & Shamir, 1977] and [Fujiwara et al., 1987]. The solution is obtained via a hierarchial decomposition of the optimization problem. The primary variables are the flow in the network. For each flow distribution the other decision variables are optimized by linear programming. Post optimality analysis of the linear program provides the information necessary to compute the gradient of the total cost with respect to changes in the flow distribution. The gradient is used to change the flows so that a (local) optimum is approached.

Two global search schemes, MULTISTART and ANNEALING, were used in [Loganathan & Greene, 1993]. The problem is decomposed into a two stage problem where one search strategy selects link flows and a second strategy, linear programming, seeks the optimal pipe design.

Problem formulation:

Consider the network shown in Figure 4.5.

- the network has 7 nodes, the water demand and ground elevation at each node are known.
- the head at each node has to be at least 30 m above the ground elevation of the node.
- the network has 8 links with known lengths.
- the data on allowable gradient, available pipe dimensions, maximum diameter of pipe to be used, Hazen Williams coefficients and unit prices for each pipe size are known.

Each link is assumed to comprise m pipes and the overall pipe cost is calculated as

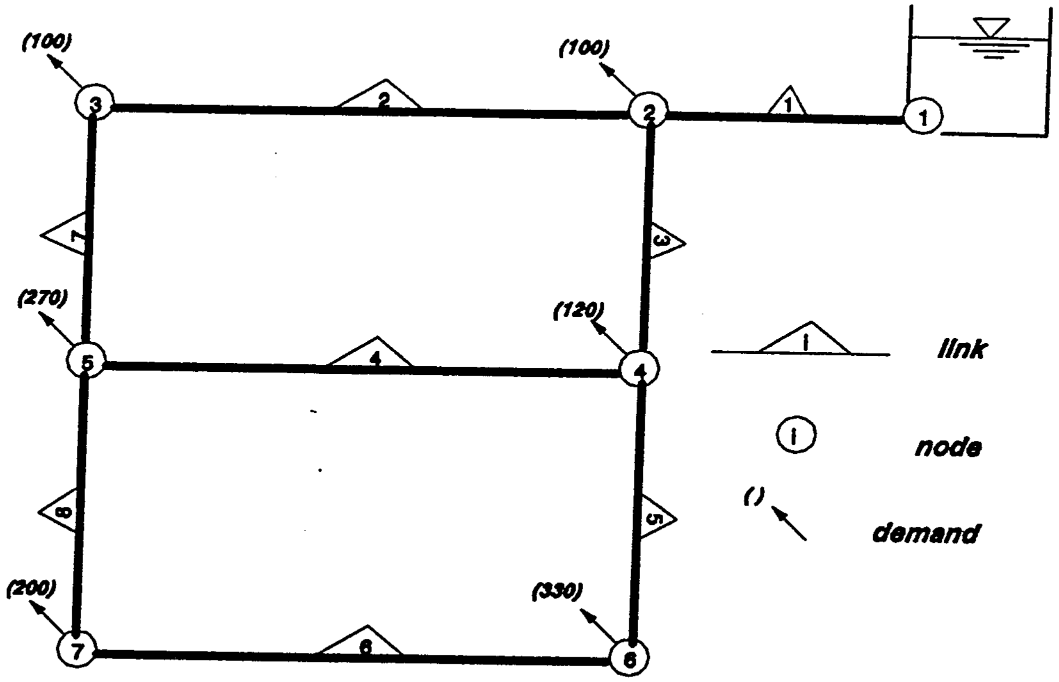


Figure 4.5: Two-looped water supply system.

$$cost = \sum_{i,j} \sum_{m} c_{ijm} x_{ijm}$$

The objective of the problem is to minimize cost under the following constraints.

$$\sum_{m} x_{ijm} = L_{ij}$$

$$x_{ijm} \geq 0$$

$$\Delta H_{ijm} = J_{ijm} x_{ijm}$$

$$J = \alpha (Q/C)^{1.852} D^{-4.87}$$

$$H_{min_n} \leq H_s \pm \sum_{i,j} \sum_{m} J_{ijm} x_{ijm} \leq H_{max_n}$$

where the first summation is over all links *(i,j)* connecting node s and node n, and the second summation is over all segments m in each link.

The nomenclature is as follows:

x_{ijm} is the length of a pipe segment of the m^{th} diameter in the link connecting nodes i and j;

c_{ijm} is the unit cost of the m^{th} diameter in the link connecting nodes i and j;

L_{ij} is the length of the link connecting nodes i and j;

ΔH_{ijm} is the head loss of the m^{th} diameter in the link connecting nodes i and j;

J_{ijm} is the hydraulic gradient of the m^{th} diameter;

Q is the discharge;

C is the Hazen Williams constant;

D is the pipe diameter;

H_s is the head at node s;

H_{min_n} and H_{max_n} are the lower and upper head constraints at each node n; and

α is a coefficient $= 1.526 * 10^{10}$ when Q is in m^3/s and D in cm.

Input data for the pipe system are given in Table 4.3 and Table 4.4. The flow in each pipe for a given set of pipe diameters and this input data was solved using the conventional Hardy Cross relaxation method.

The least cost solution for looped pipe network problems demands that some links will have two sections with different diameters [Orth, 1986]. Therefore, each link is assumed to consist of two different sized pipes i.e. $m = 2$. The sizes of the pipes in a link are to be consecutive sizes available in the market. The problem has 16 pipe diameter variables,

Node	Demand	Elevation	Head	Link	Length
1	-1120	210	210	1	1000
2	100	150	—	2	1000
3	100	160	—	3	1000
4	120	155	—	4	1000
5	270	150	—	5	1000
6	330	165	—	6	1000
7	200	160	—	7	1000
				8	1000

Table 4.3: Input data for the pipe network

Diameter (in)	Unit cost ($/ft)	Diameter (in)	Unit cost ($/ft)
1	2	12	50
2	5	14	60
3	8	16	90
4	11	18	130
6	16	20	170
8	23	22	300
10	32	24	550

Table 4.4: Available pipe diameters and unit prices

which take discrete values, and another 16 continuous pipe length variables. Therefore, the problem has a total of 32 decision variables.

Observations:

Optimal solutions cited in the first source literature and those found with LSP are given in Table 4.5 and Table 4.6 respectively. It is reported in [Alperovits & Shamir, 1977] that the problem was solved after 19 iterations, which took 4.05 seconds CPU time on IBM 370/168 machine. LSP found an optimal solution different from the one cited in the literature. The solution found with LSP showed about 12.91% improvement on the objective function value over the solution cited in [Alperovits and Shamir, 1977]. In the second source literature, [Fujiwara et al., 1987], a minimum cost of $ 415, 271 is reported.

Unfortunately no details of the solution are given in the literature, hence no comparison could be made with the LSP results. In the third source literature, Loganathan and Greene found a minimum cost of $ 405,301, which is an improvement over the solution found with LSP. A close examination of this solution revealed that the minimum head requirements at two nodes were violated.

Link	Total length	Bigger pipe		Smaller pipe	
		Length	Diameter	Length	Diameter
	(m)	(m)	(in)	(m)	(in)
1	1000	255.97	20	744.00	18
2	1000	996.37	8	3.61	6
3	1000	999.98	18	0	0
4	1000	319.38	8	680.62	6
5	1000	1000.00	16	0	0
6	1000	784.94	12	215.06	10
7	1000	999.99	6	0	0
8	1000	990.91	6	9.06	4
Total cost = $ 479,525					

Table 4.5: Optimal design for the pipe network given in ASCE, 1977

Link	Total length	Bigger pipe		Smaller pipe	
		Length	Diameter	Length	Diameter
	(m)	(m)	(in)	(m)	(in)
1	1000	0	22	1000	20
2	1000	0	12	1000	10
3	1000	632.13	16	367.87	14
4	1000	232.30	2	767.70	1
5	1000	294.89	16	705.11	14
6	1000	0	12	1000	10
7	1000	860.36	10	139.64	8
8	1000	114.14	2	885.86	1
Total cost = $ 417,607					

Table 4.6: Optimal design for the pipe network found with LSP

4.3.3 10-member truss design [Problem 10-2-0-40]

Source:

1. Venkayya, V. B., "Design of optimum structures" in Computers and Structures, Vol 1, pp 265-309, 1971.
2. David E. Goldberg and Manohar P. Samtani, "Engineering optimization via genetic algorithm", in Electronic computation, pp 471-482, Kenneth M Will ed 1986.

Methods of optimization used in the source literature: Venkayya used a method based on an energy criteria and a gradient search procedure for design of structures subjected to static loading. A Genetic Algorithm was used in [Goldberg & Samtani, 1986].

Problem formulation:

The geometry and loading system of an indeterminate structure for a 10 member truss is given, as shown in Figure 4.6. The cross sectional area of all members to give the minimum weight of the structure is required. The optimization problem is formulated as follows.

$$Minimize\ Weight = \sum_{j=1}^{10} \rho\, L_j\, A_j$$

where

ρ is the specific weight of the material used

W is the total weight of the members

A_j is the cross sectional area of the member j in square inches

L_j is the member j length in inches

The problem is constrained by the allowable stresses and the bounds on member sections.

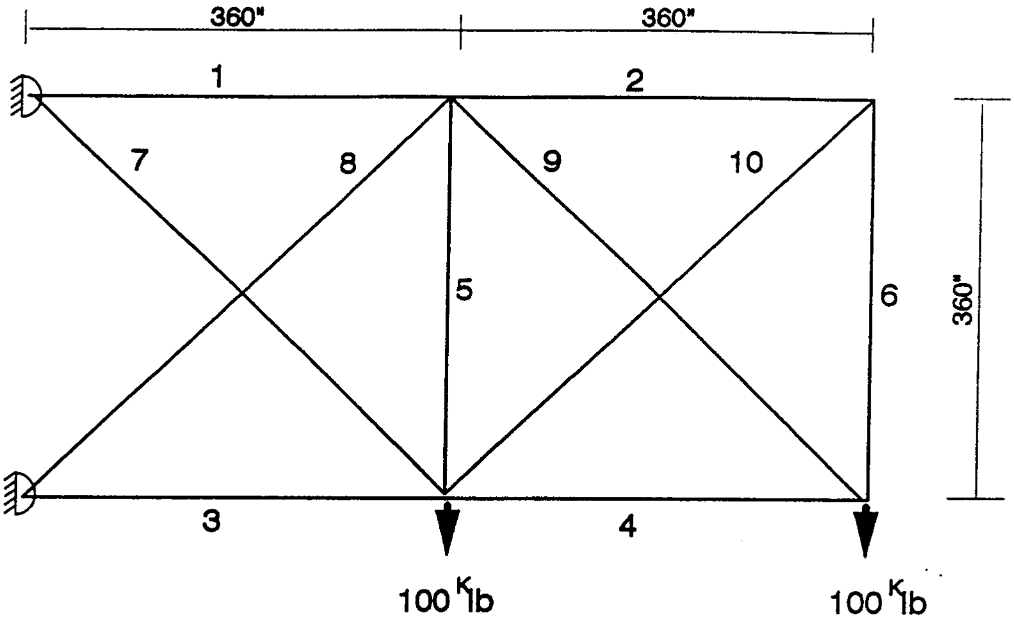


Figure 4.6: 10-member truss.

$$\sigma_{min} \leq \sigma_j \leq \sigma_{max}$$
$$A_{min} \leq A_j \leq A_{max}$$

where

σ is member stress

σ_{min} and σ_{max} are the minimum and maximum stresses.

A_{min} and A_{max} are the minimum and maximum areas allowable, which are given as $0.1\,in^2$ and $10\,in^2$ respectively.

The material to be used is aluminum which has maximum stress of 25,000 psi (compression and tension), Modulus of elasticity 10^7 psi and specific weight of $0.1\,lb/in^3$. Both loads shown in Figure 4.6 are 100^K lb each. A stiffness analysis computer program developed by Fleming [Fleming, 1986] was used with LSP to solve the structural analysis.

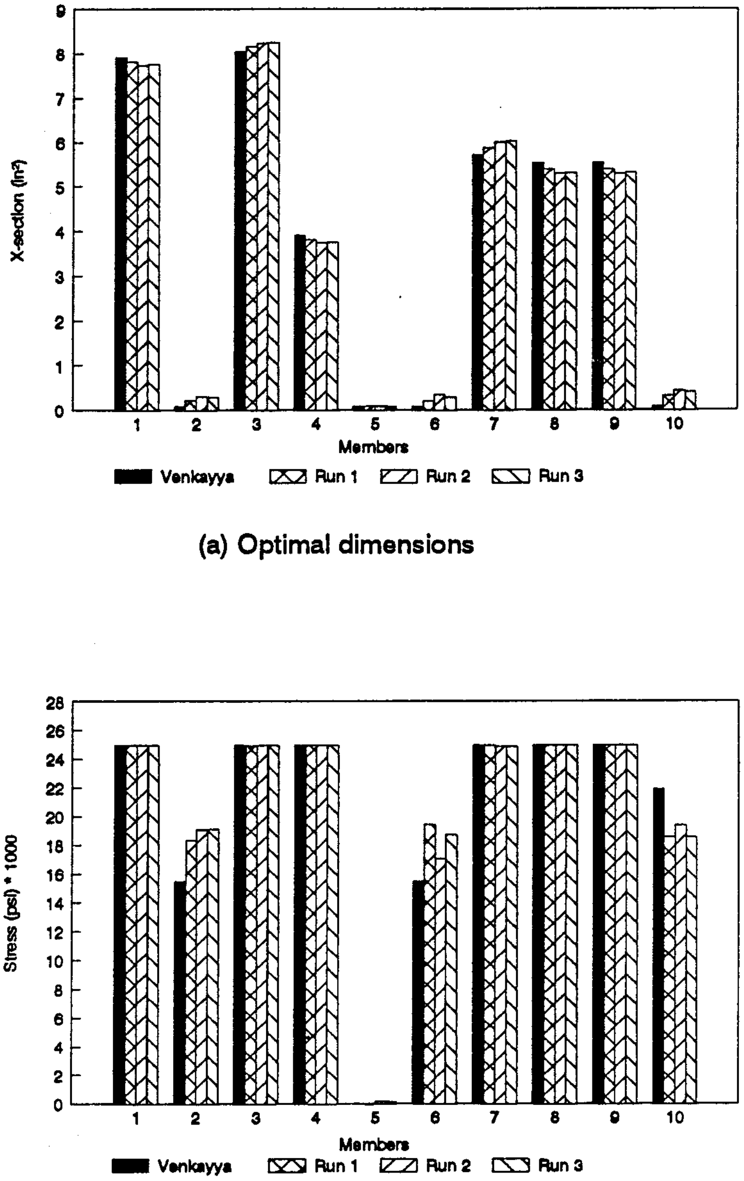
Observations:

This problem has been solved by many authors using different methods. Goldberg and Samtani [Goldenberg & Samtani, 1986] are those who addressed the problem to demonstrate the use of the "Genetic Algorithm" for solving engineering problems. They compare their result with Venkayya's [Venkayya, 1971] result. Assuming Venkayya's confirmed feasible solution as the 'true' solution, their three best-of-run results show that the Genetic Algorithm results deviated from the 'true' solution by 0.82%, 1.32% and 2%. Unfortunately the best run cited in [Goldenberg & Samtani, 1986] violates six of the ten stress constraints, which is acknowledged as 'minor excursions beyond the stress constraint'. Figure 5 of the original source paper shows the violations. It cited that 6,400 points were explored to reach the solution in all the three runs.

A number of LSP runs were made to compare results with the 'true' solution results. The three best LSP feasible results showed 0.6%, 0.9% and 1.00% deviation from the results given by Venkayya. LSP found a set of well separated near optimal points, which suggests that the problem has multiple global optima. The LSP results given in Table 4.7 were found, on the average, after 73,800 function evaluations per each run which took 5 minutes and 15 seconds on a 80386-33Mhz microcomputer. The detailed results cited by Venkayya and those found with LSP are given in Table 4.7. The three different solutions found with LSP and Venkayya's optimum solution are shown in Figure 4.7. Figure 4.7(a) shows the optimal decision variables, whereas Figure 4.7(b) shows the stress in each member of the optimal solution.

LSP's failure to identify the optimal solution in this instance can be attributed to the fact that it lies on the boundary of the original search domain. Further improvement in the LSP solution might be achieved through local refinement using a gradient method but was not attempted.

For a general comparison, a summary of minimum objective function values for the truss problem found with the two methods mentioned above, and LSP, are given in Table 4.8.



(a) Optimal dimensions

(b) Optimal member stresses

Figure 4.7: Detailed results for Venkayya's 10-member truss problem.

Member	Cross sectional area (in^2)			
	Venkayya	LSP		
		run 1	run 2	run 3
1	7.938	7.826	7.753	7.778
2	0.100	0.236	0.321	0.289
3	8.062	8.179	8.251	8.223
4	3.938	3.827	3.754	3.779
5	0.100	0.100	0.100	0.101
6	0.100	0.223	0.359	0.295
7	5.745	5.906	6.028	5.992
8	5.569	5.411	5.308	5.343
9	5.569	5.411	5.309	5.343
10	0.100	0.330	0.447	0.421
Truss weight (lb)	1,593.2	1,602.6	1,609.6	1,607.2

Table 4.7: Detailed results for the 10-member truss design

Venkayya		Goldberg	LSP
1,593.2	run 1	1,606	1,602.6
'True'	run 2	1,614	1,609.6
solution	run 3	1,625	1,607.2

Table 4.8: Comparison of truss weights for the 10-member truss design

4.3.4 38-member truss design [Problem 38-1-0-77]

Source: Andrew B. Templeman, "Discrete optimum structural design", Computer and Structures, Vol 30, No 3 pp 511-518, 1988.

Method of optimization used in the source literature: Heuristic methods for finding discrete optimum solution.

Problem formulation:

Consider the 38-member truss shown in Figure 4.8. With the given loading system, the truss is to be designed in such a way that the tip displacement does not exceed 10 millimetres. Members should be chosen from the five different cross section bars available, that is 5.0, 10.0, 20.0 40.0 and 75.0 $* 10^3 \, mm^2$. All bars are of the same material with mass density, $\rho = 7.85 * 10^{-6} \, Kg/mm^3$ and Young's Modulus, $E = 210 KN/mm^2$.

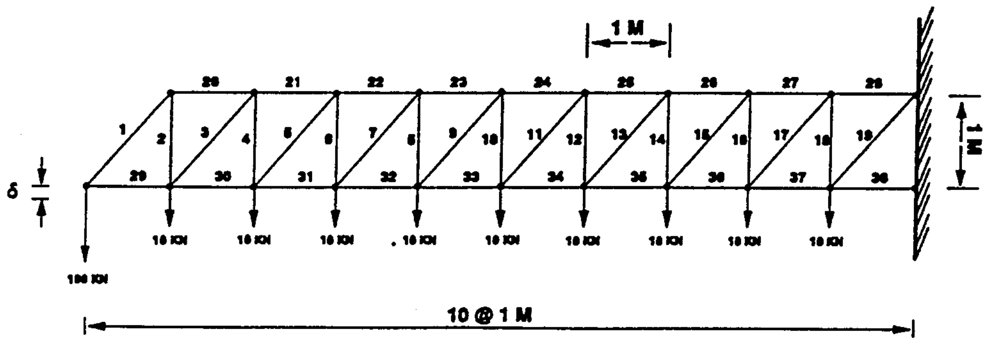


Figure 4.8: 38-member truss.

The total mass of the truss is used to represent the cost so the optimization problem is formulated as

$$Minimize\ Mass = \sum_{i=1}^{38} \rho \, L_i \, A_i$$

$$\text{Subject to } \delta \leq 10\,mm$$

where L_i and A_i are the length and cross section of the i^{th} member, and δ is the displacement at the tip of the truss. L_i are known quantities and the A_i are discrete variables.

Observations:

The result given in the literature is reported to be the global discrete optimum design. Even though the difference is small, LSP achieved a slight improvement in the objective function. The most interesting aspect is that LSP identified multiple global optima. Over 40 different global optimal solutions were identified, of which only ten are given in Table 4.9 along with the single solution given in the source literature.

Fleming's [Fleming, 1986] computer programme for plane trusses was used with LSP to solve the structural analysis problem using the stiffness method. It took about 75 hours to meet the modified variance convergence criterion using an 80386-33Mhz microcomputer, but the total number of function evaluations was less than 500,000. The reason for such a long computation time is the effort expended on the determination of the deflection for the many designs evaluated in the LSP search.

As was pointed out by the external examiner Dr. A. Templeman, there exists a unique algebraic equation for the tip deflection of this truss. While use of the equation would have significantly reduced the computational time needed for constraint evaluations the number of function evaluations would remain unchanged.

	Member areas ($1000mm^2$)											
Member	Literature	LSP solutions										
1	10	10	10	10	10	10	10	10	10	10	10	
2	5	10	5	5	5	10	5	5	5	5	5	†
3	10	10	10	10	10	10	10	10	10	10	10	
4	10	5	10	10	10	10	5	10	10	5	10	†
5	10	10	10	10	10	10	10	10	10	10	10	
6	10	10	5	10	5	10	10	10	10	10	10	†
7	10	10	10	10	10	10	10	10	10	10	10	
8	10	10	10	10	10	10	5	5	10	10	10	†
9	10	10	10	10	10	10	10	10	10	10	10	
10	10	5	10	10	10	5	10	10	10	10	10	†
11	10	10	10	10	10	10	10	10	10	10	10	
12	10	10	10	10	10	10	10	10	10	10	5	†
13	10	10	10	10	10	10	10	10	20	10	10	⋆
14	10	10	10	10	10	10	10	10	10	10	10	
15	10	10	10	10	10	10	10	10	10	10	10	
16	10	10	10	10	10	10	10	10	10	10	10	
17	10	10	20	20	10	10	10	20	10	10	10	⋆
18	10	10	10	10	10	10	10	10	10	10	10	
19	10	20	10	10	20	20	20	10	10	20	20	⋆
20	5	5	5	10	5	5	5	10	5	5	5	†
21	20	10	10	10	10	10	10	10	10	20	10	⋆
22	20	20	20	20	20	20	20	20	20	20	20	
23	40	40	40	40	40	40	40	40	40	40	40	
24	40	40	40	40	40	40	40	40	40	40	40	
25	40	40	40	40	40	40	40	40	40	40	40	
26	75	75	75	75	75	75	75	75	75	75	75	
27	75	75	75	75	75	75	75	75	75	75	75	
28	75	75	75	75	75	75	75	75	75	75	75	
29	5	5	5	5	5	10	10	10	10	5	5	†
30	20	20	20	10	20	10	20	10	10	10	20	⋆
31	20	20	20	20	20	20	20	20	20	20	20	
32	40	40	40	40	40	40	40	40	40	40	40	
33	40	40	40	40	40	40	40	40	40	40	40	
34	40	40	40	40	40	40	40	40	40	40	40	
35	75	75	75	75	75	75	75	75	75	75	75	
36	75	75	75	75	75	75	75	75	75	75	75	
37	75	75	75	75	75	75	75	75	75	75	75	
38	75	75	75	75	75	75	75	75	75	75	75	
Mass (Kg)	8489.16	8482.42										
† takes either 5 or 10		⋆ takes either 10 or 20										

Table 4.9: Alternate optimal designs for the 38-member truss

4.3.5 Unit Hydrograph ordinate determination . [Problem 19-1-21-0]

Source: Larry W. Mays and Cheng-Kang Taur, "Unit Hydrographs via Nonlinear Programming", Water Resources Research, Vol 18, No 4 pp 744-752, 1982.

Method of optimization used in the source literature: Large scale Generalized Reduced Gradient Technique, where the output from a linear programming version of the problem is used as the starting point for GRG2.

Problem formulation:

The optimal unit hydrograph from a set of known rainfall and runoff data is required. The unit hydrograph should be able to produce either a minimum sum of deviations or a minimum sum of squares of deviations between observed and derived runoff hydrographs. Unlike the traditional approach, determination of rainfall excess does not need to be defined in advance. Therefore rainfall excess values are considered as part of the set of decision variables. The relationship between rainfall excess and total rainfall is shown in Figure 4.9.

At any time much of the rainfall is lost as ground water flow and evapotranspiration and only the rest contributes to the surface runoff. The rainfall excess, which contributes to the surface runoff, is expressed as

$$P_{i,n} = R_{i,n} - H_{i,n} \qquad i = 1, 2, ..., I \qquad n = 1, 2, ..., N_i$$

where

$R_{i,n}$ is the total rainfall intensity of event i at time n.

$H_{i,n}$ is the rainfall loss of event i at time n.

$P_{i,n}$ is the rainfall excess of event i at time n.

I is the total number of observed hydrographs.

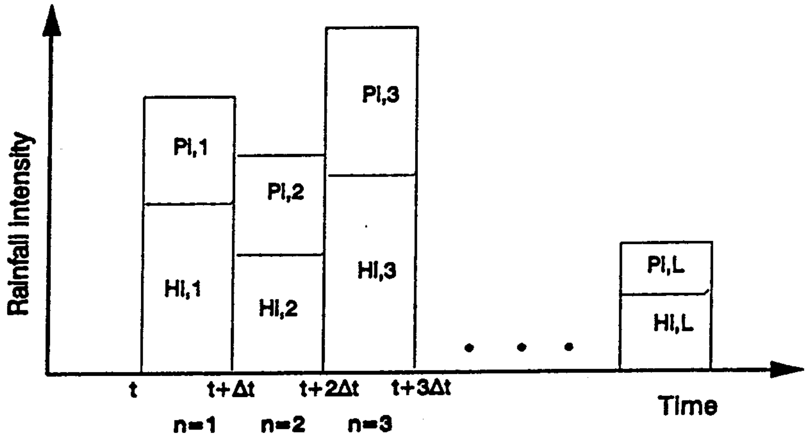


Figure 4.9: Rainfall components.

N_i is the total number of ordinates of observed hydrograph from the i^{th} event.

The objective of this problem is to minimize the sum of absolute deviations between the observed runoff and the runoff derived from the unit hydrograph. The optimization problem is formulated, in its general form, as

$$Minimize \quad \sum_{i=1}^{I}\sum_{n=1}^{N_i} \mid F_{i,n} - Q_{i,n} \mid^r$$

Subject to:

$$Q_{i,n} = P_{i,n}U_1 + P_{i,n-1}U_2 + ... + P_{i,n-m+1}U_m$$

$$\text{for } i = 1, 2, ..., I \text{ and } n = 1, 2, ..., N_i$$

$$\sum_{n=1}^{L_i} P_{i,n} = D_i$$

$$D_i = K \sum_{n=1}^{N_i} Q_{i,n}$$

$$K \sum_{m=1}^{M} U_m = 1$$

$$U_m \geq 0 \qquad m = 1, 2, .., M$$

where

M is total number of unit hydrograph ordinates.

U_m is the value of the m^{th} unit hydrograph ordinate.

$F_{i,n}$ is observed direct runoff from the i^{th} hydrograph at the n^{th} ordinate.

$Q_{i,n}$ is derived direct runoff from the i^{th} hydrograph at the n^{th} ordinate.

D_i is the direct runoff volume for the i^{th} rainfall event.

L_i is the number of periods of rainfall excess for the i^{th} event.

r is a constant which takes a value of 1 or 2 depending upon
the definition of the objective function.

K is a constant expressed as

$$K = \frac{12(3600)}{(5280)^2}\frac{\Delta t}{A}.$$

where Δt is the interval in hours between hydrograph ordinates, and A is the drainage area of the watershed in $miles^2$.

LSP solved the Little Walnut Creek watershed in Austin, Texas, hydrograph using the data given in the literature. Table 4.10 shows details of the data recorded on 07/19/79. The watershed has a drainage area of 5.57 $miles^2$. The problem has 9 excess rainfall variables and 10 unit hydrograph ordinates which makes the total number of decision variables to be 19.

In the source literature this problem was solved using different approaches. The best result was claimed to be the minimization of the sum of absolute deviations using GRG2. It is reported that optimal solution was reached after 7.8 seconds on CDC Cyber 170/750 B computer system. The results cited for the 'best' result were used to make a comparison with the LSP results and are given in Table 4.11. The optimal unit hydrographs suggested in the literature and the one found with LSP are given in Figure 4.10.

Time (hr)	Total rainfall (in)	Observed runoff (cfs)
1845	0.02	0.4
1915	0.36	3.0
1945	0.19	12.5
2015	0.16	62.
2045	0.49	466.00
2115	0.87	1750.00
2145	0.13	985.00
2215	0.04	647.00
2245	0.01	330.00
2315		216.00
2345		97.00
0015		52.00
0045		43.00
0115		33.80
0145		24.60
0213		19.60
0245		18.80
0315		17.90

Table 4.10: Rainfall-Runoff data for Little Walnut Creek

Observations:

The results with LSP show a considerable improvement in the objective function value over that is cited in the literature. The total sum of deviation between observed and calculated runoff cited in the literature is 54.7 whereas the LSP results show the sum of deviations to be 50.57. This is about an 8% improvement on the optimum value. But an equally significant improvement from a hydrological modelling standpoint is that the unit hydrograph derived from the LSP results is superior to the hydrograph result cited in the literature. The recession side of the hydrograph derived from LSP results shows the smooth monotone decay expected in a unit hydrograph. Figure 4.10 contrasts the unit hydrographs derived from the two results.

Literature			LSP		
Rainfall Loss (in)	Unit Hydrograph Ordinates (cfs)	Calculated Runoff (cfs)	Rainfall Loss (in)	Unit Hydrograph Ordinates (cfs)	Calculated Runoff (cfs)
0.020	4445.00	0.40	0.0199	3638.42	0.37
0.359	841.90	3.00	0.3592	1506.41	3.00
0.187	881.70	12.50	0.1869	964.16	12.47
0.147	271.30	62.00	0.1445	399.77	62.03
0.388	347.30	466.00	0.3693	335.04	466.01
0.498	70.00	1750.00	0.4436	124.20	1749.96
0.000	50.50	985.00	0.0698	58.79	985.10
0.000	68.60	647.00	0.0148	64.24	647.30
0.000	36.10	330.00	0.0042	52.10	330.05
	86.30	216.00		45.79	216.06
		97.00			97.05
		52.00			51.99
		43.00			43.00
		33.80			33.81
		40.00			24.62
		12.35			4.44
		3.80			1.45
		0.90			0.27

Table 4.11: Optimal hydrograph for the Little Walnut Creek

4.3.6 Air pollution control design [Problem 4-11-0-40]

Source: Wang, Bi-Chong and Luus, Ruin, "Reliability of optimization procedures for obtaining global optimum", AIChE Journal Vol 24, No 4, pp 619-626, 1978.

Method of optimization used in the source literature: Direct search method based on random sampling and search domain contraction.

Problem formulation:

The maximum ground level concentration of a pollutant resulting from the emission of multiple sources is required. Holland's plume rise equation and Gifford's dispersion

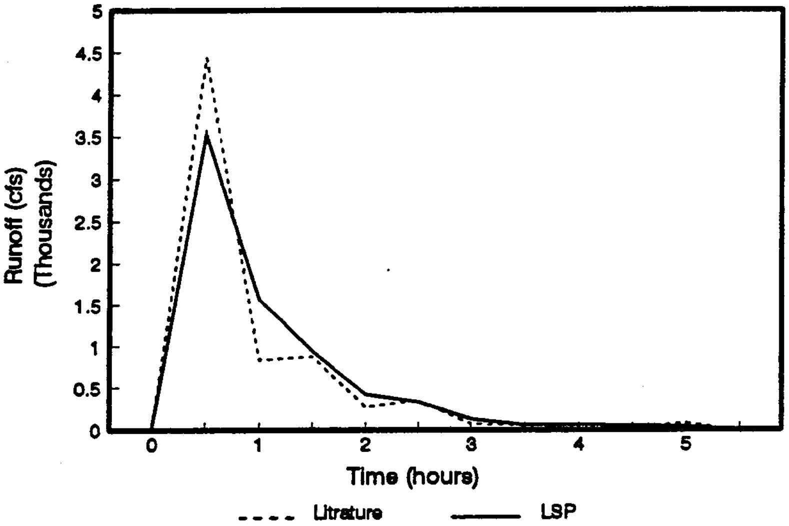


Figure 4.10: Optimal unit hydrographs for Little Walnut Creek.

equation are used for estimating the ground level sulphur dioxide concentration [Turner, 1973]. Under adiabatic conditions, the ground level concentration of sulphur dioxide is given by

$$C = \frac{0.8}{\pi u}\sum_{i=1}^{10}\frac{Q_i}{\sigma_{yi}\sigma_{zi}}exp[-.5(\frac{Y_i}{\sigma_{yi}})^2 - .5(\frac{H_i}{\sigma_{zi}})^2]$$

where

$$Y_i = sin\theta(x - a_i) + cos\theta(y - b_i)$$

$$X_i = cos\theta(x - a_i) - sin\theta(y - b_i)$$

$$\Delta H_i = \frac{V_{si}d_i}{u}[1.5 + 2.68\frac{T_{si}-T_a}{T_{si}}d_i]$$

$$H_i = H_{si} + \Delta H_i \qquad \text{for } i = 1, 2, ..., 10$$

$$\sigma_{yi} = \begin{cases} 0.9591 & X_i \leq 10 \\ 0.1136 X_i^{0.9265} & 10 < X_i \leq 2 * 10^3 \\ 0.1385 X_i^{0.9015} & 2 * 10^3 < X_i \leq 10^4 \\ 0.2030 X_i^{0.8600} & 10^4 < X_i \leq 10^5 \end{cases}$$

$$\sigma_{yi}\sigma_{zi} = \begin{cases} 0.07925 & X_i \leq 10 \\ 4.828 * 10^{-5} (ln\, X_i)^{8.8766} & 10 < X_i \leq 200 \\ 3.108 * 10^{-6} (ln\, X_i)^{10.5295} & 200 < X_i \leq 10^3 \\ 1.808 * 10^{-7} (ln\, X_i)^{11.9998} & 10^3 < X_i \leq 5 * 10^3 \\ 1.892 * 10^{-9} (ln\, X_i)^{14.1284} & 5 * 10^3 X_i \leq 10^5 \end{cases}$$

and the values of all constants are given in Table 4.12.

a_i	b_i	H_{si}	d_i	T_{si}	Q_i	V_{si}
-3000	-2500	183.0	8.0	413	2882.6	19.245
-2600	-300	183.0	8.0	413	2882.6	19.245
-1100	-1700	160.0	7.6	413	2391.3	17.690
1000	-2500	160.0	7.6	413	2391.3	17.690
1000	2200	152.4	6.3	413	2173.9	23.404
2700	1000	152.4	6.3	413	2173.9	23.404
3000	-1600	121.9	4.3	413	1173.9	27.128
-2000	2500	121.9	4.3	413	1173.9	27.128
0	0	91.4	5.0	413	1304.3	22.293
1500	-1600	91.4	5.0	413	1304.3	22.293

Table 4.12: Stack and Emission data

where

a_i = x coordinate of stack i, (m)

b_i = y coordinate of stack i, (m)

C = average ground level sulphur dioxide concentration based on 30 minute sampling time, (g/m^3)

d_i = exit diameter of stack i, (m)

H_i = effective height of stack i, (m) H_{si} = height of stack i, (m)

Q_i = emission rate of stack i, (m/s)

T_a = air temperature, taken as (283^oK)

T_{si} = exit gas temperature at top of stack i, (^oK) u = wind velocity, (m/s)

V_{si} = exit gas velocity of stack i, (m/s)

X_i = downwind distance from stack i, (m)

Y_i = crosswind distance from the plume centre line of stack i, (m)

ΔH_i = plume rise from stack i, (m)

θ = wind direction measured clockwise from x axis, (radian)

The optimization problem is to find the coordinates on the ground $x \sim y$, the wind velocity u, and the wind direction θ for which C is maximized. From physical considerations, the following constraints are imposed.

$$-20000 \leq x \leq 20000$$

$$-20000 \leq y \leq 20000$$

$$0 \leq u \leq 12.5$$

$$0 \leq \theta \leq 2\pi$$

It is reported in the literature that this 4-variable problem has numerous local optima. It is also mentioned that one of the local optimal values which is $C = -7.5316$ is close to the global optimum value. The closeness of these objective function values makes the search more challenging for both gradient and LSP methodologies.

Observations:

The global solution cited in the source literature and that found with LSP are given in Table 4.13.

Variable	Literature	LSP
x	-8039.6	-8038.06
y	9369.2	9369.70
u	5.6371	5.641006
θ	3.996	3.9966
C	$-7.7114 * 10^{-4}$	$-7.7092 * 10^{-4}$

Table 4.13: Optimal design cited in the literature and that found with LSP

The result found with LSP is similar to the result cited in the source literature. The convergence criterion was set by trial at $V_M \leq 1 * 10^{-19}$ to achieve comparable accuracy to the cited solution. LSP reached convergence after 28,220 function evaluations which took 3 minutes and 35 seconds on a 80386-33Mhz microcomputer.

4.3.7 Irrigation system design [Problem 6-7-1-4]

Source:

1. Holzapfel, Edward A., Marino, Migual A. and Chavez-Morales, Juses, "Surface Irrigation Optimization Model", ASCE, Irrigation and Drainage Engineering Journal, Vol 112 No 1, pp 1-19, Feb 1986.

2. Holzapfel, Edward A. and Marino, Migual A., "Surface Irrigation Nonlinear Optimization Models", ASCE Journal of Irrigation and Drainage Engineering, Vol 113 No. 3 pp 379-392, August 1987.

Methods of optimization used in the source literature: The first authors, [Holzapfel et al., 1986], solved this problem using linear approximations of the nonlinear objective function and constraint equations, then they used linear programming packages. The second authors, [Holzapfel et al., 1987], used MINOS 5.0 to solve the same problem.

Problem formulation:

An irrigation system is to be designed so that the maximum profit from the crop can be obtained by identifying the most efficient furrow irrigation system. The example adopts the field data and profit yield relationship for the Chillan project in Chile. A surface irrigation system is designed on the basis of soil, crop, topography, size and shape of the irrigable area, and the availability of farm equipment. For this particular case the six design variables involved in the problem formulation are: 1) the flow discharge; 2) the length of run in the direction of the flow; 3) the time of irrigation cutoff; 4) the number of furrows per set; 5) the number of runs per set; and 6) the number of sets per day.

The objective of the design is to maximize profit (PRO) which is defined as total revenue minus total cost, expressed as

$$PRO = \gamma P_c - (W_c + L_c) - OC$$

where

PRO =Profit in dollars per hectare

γ = Yield in metric tons per hectare

P_c = Price of crop in dollars per ton

W_c = Water cost in dollars per hectare

L_c = Seasonal labour cost in dollars per hectare

OC = Other costs (cost of crop, fertilizer, pesticides, harvesting, etc). Since OC does not change with alternative irrigation systems, it can be dropped from the analysis.

It was reported in the source literature that, in all the studies they made earlier, a good correlation was found between profit and requirement distribution efficiency (RDE). The relationship of PRO and RDE is given as

$$PRO = C_l RDE + B_l$$

where C_l and B_l are regression coefficients with known values.

The mathematical relationship between RDE and the design variables considers the water distribution in the soil and involves the following problem variables.

The mathematical expressions are

$$RDE = MQ^{a}L^{-b}T_{co}^{c}$$

therefore

$$PRO = C_{l}MQ^{a}L^{-b}T_{co}^{c} + B_{l}$$

where

Q = inflow discharge in millilitre per second

L = Length of run in meters

T_{co} = Time of irrigation cutoff in minutes

N_{fu} = Number of furrows per set

N_{r} = Number of runs

N_{s} = Number of irrigation sets per day

Q_{T} = Total discharge available in millilitres per second

Q_{maxf} = Maximum discharge in furrow in millilitres per second

L_{T} = Total length in meters

L_{min} = Minimum run length in meters

A_{min} = Minimum area irrigated per day in square meters

F_{s} = Furrow spacing in meters

S_{t} = Maximum allowable number of irrigation sets

T_{max} = Maximum time per day to irrigate minimum area

EFF_{m} = Minimum required distribution efficiency allowable in%

$K_{Q}, M, a, b, c, u, w, C_{l}$ and B_{l} are constants.

Since C_{l}, M and B_{l} are constant values, taking them out of the objective function equation would not change the optimal solution. The reduced form of the objective function would then be

Maximize $Q^a L^{-b} T_{co}^c$

The design variables are restricted within specified limits depending upon given field conditions, and fulfil certain physical conditions. Finally the model formulation is

Maximize $Q^a L^{-b} T_{co}^c$

Subject to:

$$Q \le Q_{maxf}$$

$$Q L^{-u} T_{co}^w \ge K_Q$$

$$Q N_{fu} \le Q_T$$

$$L N_r = L_T$$

$$L N_{fu} N_s \ge A_{min}/F_s$$

$$T_{co} N_s \le T_{max}$$

$$Q^a L^{-b} T_{co}^c \le 100/M$$

$$Q^a L^{-b} T_{co}^c \ge EFF_m/M$$

$$L \ge L_{min}$$

$$N_r, N_{fu} \ge 1$$

$$N_s \le S_t$$

All necessary data for the design is given in Table 4.14.

Q_T	32,000 ml/s	K_Q	25
Q_{maxf}	1,300 ml/s	M	0.129
L_T	800 m	a	0.706
L_{min}	50 m	b	0.645
A_{min}	20,000 sq. m	c	0.809
F_s	1 m	u	1.02
T_{max}	1200 min	w	0.26
S_t	6	C_l	9.98
EFF_m	20%	B_l	-143

Table 4.14: Constants for the irrigation system design

The problem involves six variables with one equality and six inequality constraints. Lower and upper bounds on each variable are also specified. In solving the problem with LSP, three of the variables, N_{fu}, N_r and N_s were assumed to take integer values even though it was not clearly stated in the literature.

The problem was solved under two different conditions [Holzapfel et al., 1987]. The first case was where all variables were allowed to take any value within the variable domain, and the second case where the length of the run was restricted to a fixed value. Their solution to the first case was questionable so that only the second case with $L = 200$ was solved using LSP.

Observations:

Results cited in the source literatures and those found with LSP are given in Table 4.15. It took 5,785 function evaluations to meet the convergence criterion of $V_M \leq 0.0001$ with LSP.

Variables	Literature		LSP
	1986	1987	
Q (ml/s)	1,260	1,255	1,280
L (m)	200	200	200
T_co (min)	306	305.9	300
N_{fu}	26	25	25
N_r	4	4	4
N_s	4	4	4
Profit (\$)	525	524	523

Table 4.15: Optimal solution for the irrigation system design

The optimal objective function value found with LSP is slightly inferior, by about 0.19%, to those values given in the literature. But the LSP result is feasible while the other two cited are infeasible. That is

i) $Q * N_{fu} = 32,760 \not\leq 32,000$ (about 2.38% violation) [Holzapfel et al., 1986].

ii) $T_{co} * N_s = 1,224 \nleq 1,200$ (about 1.97% violation) [Holzapfel et al., 1986 and 1987].

If the same constraint violations are allowed with LSP, the objective function value can be improved by about 2.53% over the solutions cited in the literature.

Amongst all engineering fields there has perhaps been the longest sustained interest in practical applications of nonlinear optimization in chemical engineering. The following four test problems are taken from the chemical engineering literature. These particular problems have been included here for the challenge they present to NLP methods and because in recent years they have been used as test problems by some authors. In one case GAMS, which uses MINOS was used to find the optimal solution. This provided a good opportunity to test LSP against a current optimization package. The statements of the problem formulation are taken directly from the sources cited.

4.3.8 Alkylation process [Problem 10-1-7-0]

Source:

1. Rein Luus and T. H. I. Jaakola, "Optimization by direct search and systematic reduction of the size of search region", AIChE Vol 19, NO. 4, pp 760-766, 1973.
2. T. F. Edgar and D. M. Himmelblau. "Optimization of chemical process". McGraw Hill 1988.
3. Morari, Manfred and Grossmann, Ignacio E., "Chemical engineering optimization models with GAMS", CACHE Process design case studies, Vol 6, 1991.

Methods of optimization used in the source literature: Luus and Jaakola used a direct search optimization method with sequential search domain reduction. Edgar and Himmelblau used sequential quadratic programming while GAMS was used in the third source cited.

Problem formulation:

The optimization problem is to determine the optimal operating conditions for the simplified alkylation process shown in Figure 4.11. The variables involved in the problem formulation, along with their upper and lower bounds, are given in Table 4.16. The objective function was defined in terms of alkylate product, or output value minus feed and recycle costs; operating costs were not reflected in the function. The profit per day to be maximized is:

$$f(\mathbf{x}) = C_1 x_4 x_7 - C_2 x_1 - C_3 x_2 - C_4 x_3 - C_5 x_5$$

where

C_1 = alkylate product value ($ 0.063 per octane-barrel)

C_2 = olefin feed cost ($5.04 per barrel)

C_3 = isobutane recycle costs ($0.035 per barrel)

C_4 = acid addition cost ($10.00 per thousand pounds)

C_5 = isobutane make-up cost ($3.36 per barrel)

It was stated in the literature that a regression analysis was first carried out to form the process model. The alkylate yield, x_4, was a function of the olefin feed, x_1, and the external isobutane-to-olefin ratio, x_8. The relationship determined by nonlinear regression, holding the reactor temperatures between 80 to 90°F and the reactor acid strength by weight percent at 85 to 93, was

$$x_4 = x_1(1.12 + 0.13167 x_8 - 0.00667 x_8^2).$$

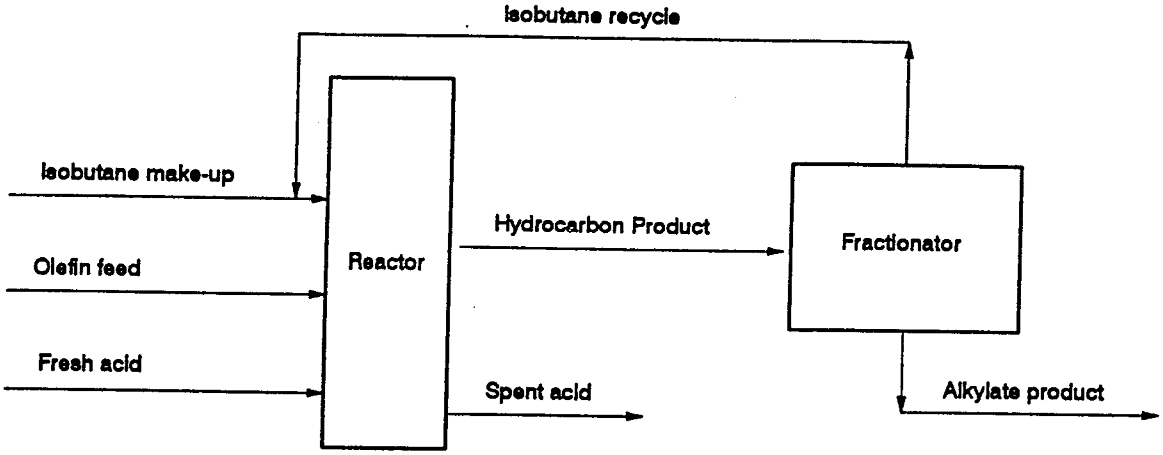


Figure 4.11: Alkylation process.

The isobutane make-up, x_5, was determined by a volumetric reactor balance. The alkylate yield, x_4, equals the olefin feed, x_1, plus the isobutane make-up, x_5, less shrinkage. The volumetric shrinkage can be expressed as 0.22 volume per volume of alkylate yield so that

$$x_5 = 1.22x_4 - x_1.$$

The acid strength by weight percent, x_6, could be derived from an equation that expressed the acid addition rate, x_3, as a function of the alkylate yield, x_4, the acid dilution factor, x_9, and the acid strength weight percent, x_6 (the addition acid was assumed to have acid strength of 98%).

$$x_6 = \frac{98000x_3}{x_4x_9 + 1000x_3}.$$

The external isobutane-to-olefin ratio, x_8, was equal to the sum of the isobutane recycle, x_2, and the isobutane make-up, x_5, divided by the olefin feed, x_1

$$x_8 = \frac{x_2 + x_5}{x_1}.$$

Variable	Description	Lower bound	Upper bound
x_1	Olefin feed (barrels per day)	0	2000
x_2	Isobutane recycle (barrels per day)	0	16000
x_3	Acid addition rate (100 lb per day)	0	120
x_4	Alkylate yield (barrels per day)	0	5000
x_5	Isobutane make-up (barrels per day)	0	2000
x_6	Acid strength (weight percent)	85	93
x_7	Motor octane number	90	95
x_8	External isobutane-to-olefin ratio	3	12
x_9	Acid dilution factor	1.2	4
x_{10}	F-4 performance number	145	162

Table 4.16: Bounds for the variables involved in the alkylation process

The motor octane number, x_7, was a function of the external isobutane-to-olefin ratio, x_8, and the acid strength by weight percent, x_6 (for the same reactor temperatures and acid strengths as for the alkylate yield , x_4)

$$x_7 = 86.35 + 1.098x_8 - 0.038x_8^2 + 0.325(x_6 - 89).$$

The acid dilution factor, x_9, could be expressed as a linear function of the F-4 performance number, x_{10}

$$x_9 = 35.82 - 0.222x_{10}.$$

The last dependant variable is the F-4 performance number, x_{10}, which was expressed as a linear function of the motor octane number, x_7

$$x_{10} = -133 + 3x_7.$$

Observations:

Edgar and Himmelblau solved a modified version of this problem in which some of the constraints were relaxed. The optimal solution for the modified case is not reported here but it was observed that two of the constraints were slightly violated. The constraint violation is so small that it might have arisen from round off error.

The optimal solution cited in [Luus & Jaakola, 1973] was reported to be close to earlier works by other authors. In CACHE, the GAMS package, which makes use of MINOS 5.3, was used to solve the same problem. The results with this approach showed inferior results compared to the earlier published works. The result found with LSP is similar to the one reported by Luus and Jaakola, and is an improvement over the GAMS's results. It took 4,789 LSP function evaluations to obtain the optimal result. The optimal solutions found with the three methods are given in Table 4.17.

Variable	Optimal solution		
	Literature		LSP
	CACHE	Luus et al.	
x_1	1734.410	1728.40	1728.40
x_2	16000.000	16000.00	16000.00
x_3	98.405	98.40	98.40
x_4	3060.992	3056.00	3056.00
x_5	2000.000	2000.0	2000.00
x_6	90.592	90.60	90.60
x_7	94.170	94.20	94.20
x_8	10.378	10.41	10.41
x_9	2.629	2.61	2.61
x_{10}	149.509	149.60	149.60
$f(\mathbf{x}^*)$	1,154.4	1,162.009	1,162.017

Table 4.17: Optimal solution for the Alkylation process

4.3.9 Heat exchanger network configuration [Problem 16-1-15-0]

Source:

1. C. A. Floudas and A. R. Ciric, "Strategies for overcoming uncertainties in heat exchanger network synthesis", Computers and Chemical Engineering, 13(10), pp 1133-1152, 1989.

2. C. A. Floudas and P. M. Pardalos, "A collection of test problems for constrained global optimization algorithms", Lecture Notes in Computer Science, NO. 455, Springer-Verlag, 1990.

3. Morari, Manfred and Grossmann, Ignacio E., "Chemical Engineering Optimization Models with GAMS", CACHE Process design case studies, Vol 6, 1991.

Methods of optimization used in the source literature: GAMS was used in CACHE. Floudas and Ciric used a method where the variable set and the constraint set are decomposed into two sets leading to two subproblems so that each subproblem contains only linear constraints. This decomposition of the original problem induced a special structure in the resulting subproblems.

Problem formulation:

The problem involves determination of the optimal heat exchanger network configuration for a system of two hot streams and one cold stream. A minimum utility consumption analysis has shown that no stream of cooling water is required. The inlet temperatures, outlet temperatures, and heat capacity flow rates for the hot streams and cold streams are given in Table 4.18. The heat duties and overall heat transfer coefficients for the matches that will take place in this network are given in Table 4.19. The objective of the problem is to obtain the optimal configuration of the cold stream so as to minimize the heat exchanger investment cost.

The problem formulation for this 16 variable problem is given as follows:

$$
\begin{aligned}
Minimize\,Cost = \quad & 1300[\frac{1000}{0.05[\frac{2}{3}(\Delta T_{11}\Delta T_{12})^{.5}+\frac{1}{6}(\Delta T_{11}+\Delta T_{12})]}]^{.6} \\
& +1300[\frac{600}{0.05[\frac{2}{3}(\Delta T_{21}\Delta T_{22})^{.5}+\frac{1}{6}(\Delta T_{21}+\Delta T_{22})]}]^{.6}
\end{aligned}
$$

Stream	T in (K)	T out (K)	$FC_p(\frac{kW}{K})$
H1	500	250	4
H2	350	200	4
C1	150	310	10
$\Delta T_{min} = 10°K$			

Table 4.18: Stream data

Match	Q(kW)	$U(\frac{KW}{m^2K})$	$A(m^2)$
H1 C1	1000	0.05	207.357
H2 C1	600	0.05	137.230
Cost of heat exchangers =$1300A^{0.6}$			

Table 4.19: Match data

Subject to:

- Mass balances at the splitting and mixing points

$$f_1^I + f_2^I = 10$$

$$f_1^I + f_{12}^B - f_1^E = 0$$

$$f_2^I + f_{21}^B - f_2^E = 0$$

$$f_1^O + f_{21}^B - f_1^E = 0$$

$$f_2^O + f_{12}^B - f_2^E = 0$$

- Energy balances at mixing points and over the exchangers

$$150f_1^I + t_2^O f_{12}^B - t_1^I f_1^E = 0$$

$$150f_2^I + t_1^O f_{21}^B - t_2^I f_2^E = 0$$

$$f_1^E(t_1^O - t_1^I) = 1000$$

$$f_2^E(t_2^O - t_2^I) = 600$$

- Temperature approach calculations for the heat exchangers

$$\Delta T_{11} = 500 - t_1^O$$

$$\Delta T_{12} = 250 - t_1^I$$

$$\Delta T_{21} = 350 - t_2^O$$

$$\Delta T_{22} = 200 - t_2^I$$

- Minimum temperature approach constraints

$$\Delta T_{11}, \Delta T_{12}, \Delta T_{21}, \Delta T_{22} \geq 10$$

- Bounds of the flow rates and temperatures

$$0 \leq f_1^I, f_2^I, f_{12}^B, f_{21}^B, f_1^O, f_2^O \leq 10$$

$$2.941 \leq f_1^E \leq 10$$

$$3.158 \leq f_2^E \leq 10$$

$$150 \leq t_1^I \leq 240$$

$$250 \leq t_1^O \leq 490$$

$$150 \leq t_2^I \leq 190$$

$$210 \leq t_2^O \leq 340$$

where

f^I = flow rate flowing from the splitter at the beginning of the network to the mixer.

f^E = flow rate flowing through the exchanger

f^B = flow rate flowing from the splitter following the match to the mixer preceding the exchanger

f^O = flow rate flowing from the splitter following the exchanger to the mixer at the end of the network

t^I = inlet temperature

t^O = outlet temperature

ΔT = temperature approach.

Observations:

The decomposition procedure adopted by [Floudas & Ciric, 1989] was not necessary in the LSP solution of the problem.

The solution cited in both sources and the one found with LSP showed similar results. The optimal solution is given in Table 4.20. It took 182,905 function evaluations to get the optimal solution with LSP.

Variable	Optimum value
f_1^I	0
f_2^I	10
f_1^O	10
f_2^O	0
f_1^E	10
f_2^E	10
t_1^I	210
t_2^I	150
t_1^O	310
t_2^O	210
ΔT_{11}	190
ΔT_{12}	40
ΔT_{21}	140
ΔT_{22}	50
f_{12}^B	10
f_{21}^B	0
Objective	56825

Table 4.20: Optimal solution to the heat exchanger configuration

4.3.10 Power generation using fuel oil [Problem 4-11-0-5]

Source: Morari, Manfred and Grossmann, Ignacio E., "Chemical Engineering Optimization Models with GAMS", CACHE Process design case studies, Vol 6, 1991.

Method of optimization used in the source literature: GAMS

Problem formulation:

The two-boiler turbine-generator combination shown in Figure 4.12 is used to produce a power output of 50 MW. It can use any combination of fuel oil and blast furnace gas (BFG). However only 10.0 fuel units per hour of BFG is available. Since this supply of BFG may not be sufficient for the required power generation, fuel oil must be purchased and used. It is desired that we use the minimum total amount of fuel oil in the two generators. For this purpose, fuel requirements for the two generators are expressed as a quadratic function of the MW produced by using nonlinear curve-fitting. Thus if x (in MW) is the power produced and f (in ton/h for fuel oil and in fuel unit/h for BFG) is the amount of fuel used, then $f = a_0 + a_1x + a_2x^2$. The constants a_i $(i = 0, 1, 2)$ depend on the generator and type of fuel used.

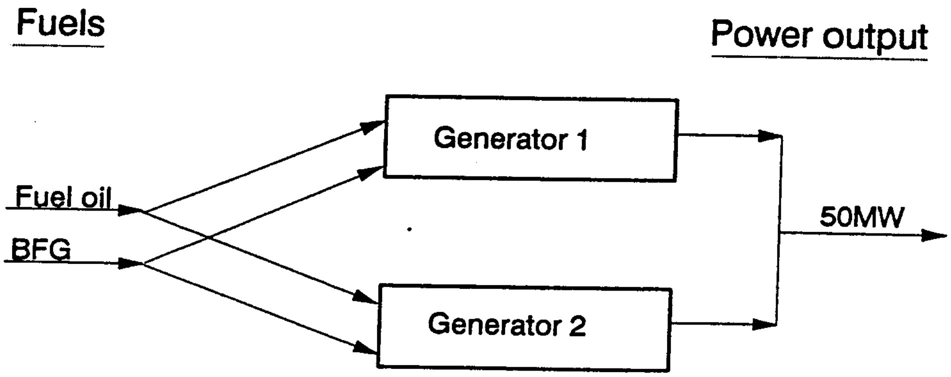


Figure 4.12: Two-boiler Turbine-Generator combination.

Assume that when a combination of fuel oil and BFG is used, the total power generated is given by the sum of the powers generated by each fuel. The ranges of operation

Generator	Fuel type	a_0	a_1	a_2
1	Fuel Oil	1.4609	0.15186	0.001450
1	BFG	1.5742	0.16310	0.001358
2	Fuel Oil	0.8008	0.20310	0.000916
2	BFG	0.7266	0.22560	0.000778

Table 4.21: Constants in fuel consumption equations

of the two generators are [18, 30] MW and [14, 25] MW respectively. It is required to formulate an optimization problem which minimizes the amount of fuel oil purchased and determine the amounts used by each generator.

For operating the generators, define f_{ij} as the amount of fuel type j used by generator i, and x_{ij} the corresponding MW generated. From the curve-fitting data

$$f_{ij} = a_{ij0} + a_{ij1}x_{ij} + a_{ij2}x_{ij}^2 \qquad i = 1,2 \qquad j = 1,2$$

where the a_{ijk} $(k = 0, 1, 2)$ are the constants for generator i and fuel type j (1 = fuel oil, 2 = BFG). The values of the constants are given in Table 4.21.

The power P_i generated by generator i will be given by

$P_i = x_{i1} + x_{i2}; \qquad i = 1,2$ where

$P_1 + P_2 \geq 50$. Define z_j as the total amount of fuel j purchased, then

$$z_j \geq \sum_{i=1}^{2} f_{ij} = \sum_{i=1}^{2} a_{ij0} + a_{ij1}x_{ij} + a_{ij2}x_{ij}^2; \qquad j = 1,2$$

$$z_2 \leq 10$$

$$x_{ij} \geq 0$$

Minimize z_1

Observations:

In the source literature, the optimal power outputs for generator 1 and 2 are 30 MW and 20 MW respectively, and 4.681 ton/h of fuel oil is purchased. BFG generates a power of 36.325 MW and the fuel oil is used to generate 13.675 MW. The optimum objective function value found with LSP is 4.683 ton/h, which is very similar to the solution cited in the literature. One of the important results found with LSP is that the global optimum is accompanied with multiple near optimal solutions. Details of solutions cited in the literature and found with LSP are given in Tables 4.22 and 4.23 respectively. The total number of function evaluations expended to meet LSP's termination criterion was 33,556, which took 50 seconds on a 80386-33Mhz computer.

Generator	Fuel oil	BFG	Power
1	10.114	19.886	30
2	3.561	16.439	20
Total	13.675	36.325	50

Table 4.22: Optimal power generation cited in the literature

Generator	Fuel oil	BFG	Power
1	10.479	19.504	29.983
2	3.254	16.768	20.022
Total	13.733	36.272	50.005

Table 4.23: Optimal power generation found with LSP

4.3.11 Chemical equilibrium [Problem 10-1-3-0]

Source: Morari, Manfred and Grossmann, Ignacio E., "Chemical Engineering Optimization Models with GAMS", CACHE Process design case studies, Vol 6, 1991.

Method of optimization used in the source literature: GAMS

Problem formulation:

From the thermodynamics of chemical reaction equilibrium, the equilibrium state of a closed system at constant temperature, T, and pressure, P, is the state at which its total Gibbs free energy is a minimum. This criterion is used to obtain the equilibrium composition of a given mixture by minimizing its free energy with respect to its composition. An ideal gas mixture of ten chemical species is maintained at $T = 298\ K$ and $P = 750\ Hg$. The species are made up of three atomic elements (e.g. H, O and C). If we denote the three elements as A, B and C, then the species formulas in terms of A, B and C are $A, A_2, A_2C, B, B_2, AB, BC, C, C_2$ and AC respectively for $s = 1, 2, ..., 10$. Being an ideal gas, the Gibbs free energy per mole of species s is given by $G_s = G_{o_s} + RT\, ln(Py_s)$, where y_s is the mole fraction of species s in the mixture. The G_{o_s} are given in terms of $w_s = G_{o_s}/RT$ in Table 4.24. The mixture contains 2 mole A, 1 mole B and 1 mole C.

s	w_S	s	w_s
1	-10.021	6	-18.918
2	-21.096	7	-28.032
3	-37.986	8	-14.640
4	- 9.846	9	-30.594
5	-28.653	10	-26.111

Table 4.24: Free energy constants (w_s)

Define x_s as the moles of species s and F as the total moles of all species in the mixture, that is

$$F = x_1 + x_2 + x_3 + ... + x_{10}.$$

Since the mixture is ideal, G_t is the sum of the free energies of the species in the mixture, where

$$G_t = x_1G_1 + x_2G_2 + x_3G_3 + ... + x_{10}G_{10}.$$

Substituting $w_s\, ln(Py_s)$ for G_s, dividing G_t by RT, and expressing the mole fractions $y_s = x_s/F$ then,

$$G = \frac{G_t}{RT} = \sum_{s=1}^{10} x_s[w_s + ln(Px_s/F)]$$

Let a_{is} denote the moles of element i in one mole of species s. The values of a_{is} are easily obtained from the species formulas. For instance $a_{11} = 1$, since species $A(s = 1)$

has only one atomic element $A(i = 1)$, $a_{21} = 0$, $a_{13} = 2$, etc.. The total number of moles of element i in the mixture, X_{0i}, is given by,

$$X_{0i} = a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + ... + a_{i10}x_{10}; \qquad i = 1, 2, 3.$$

The optimization problem is to minimize the total Gibbs free energy, G_t, with respect to the mixture composition. But since the mixture T is maintained constant, minimizing G_t is equivalent to minimizing G. Then the optimization formulation is

$$\text{Minimize } G$$

The solutions for this 10 variable problem cited in the literature and found with LSP are given in Table 4.25.

Observations:

The LSP results confirm the results cited in the literature. LSP found the solution in 3 minutes and 20 seconds on a 80386-33Mhz microcomputer, which took 83,588 function evaluations to meet the convergence criterion $V_M \leq 1 * 10^{-4}$.

	x_s	
s	Literature	LSP
1	0.007	0.0061
2	0.068	0.0709
3	0.907	0.9038
4	0.0004	0.0005
5	0.491	0.4902
6	0.0005	0.0008
7	0.018	0.0183
8	0.003	0.0034
9	0.015	0.0154
10	0.042	0.0436
G	-43.495	-43.4942

Table 4.25: Optimal solution to the Chemical equilibrium problem

4.4 Model fitting

Regression analysis is used to fix the parameters of a model from a set of experimental data. The experimental data will have a set of observed points with known values for the independent variables and the corresponding values of the dependent variable. The relationship between the dependent and independent variables can be expressed in a variety of ways. For example, as a simple linear model

$$y = \beta_0 + \beta_1 x;$$

or as a multiple linear model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n;$$

or as a polynomial model

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_n x^n;$$

or as a general nonlinear model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1 x_2 + \beta_3 x_2^2 + ...$$

In all the above forms the regression analysis determines the values of the parameters, i.e. the β vector, in the model. The methods used to estimate these parameters are known as simple, multiple linear and nonlinear regression analysis depending upon the formulation of the model. The equations used to define a certain model normally reflect the physical processes involved and models dealing with the physical world are predominantly nonlinear in their formulation. On the other hand estimating parameters for linear models is easier than for nonlinear models and therefore the equations of nonlinear models are often linearized so that linear regression methods can be used. Such transformations may not be possible with highly nonlinear models.

Three best fit criteria used in the estimation of parameters of a linear model are: minimization of the sum of the squares of deviation between observed and model generated values; minimization of sum of absolute deviations; and minimization of the maximum absolute deviation [Narula, 1982] .

Suppose a set of data is collected and the parameters for the linear model $y^j = \beta\mathbf{x}^j$ are to be estimated. To fit a model from m experimental points using the least squares, or the absolute deviations, as the model criterion, the objective function can be expressed as

$$Minimize \sum_{j=1}^{m} \mid y^j - \beta\mathbf{x}^j \mid^p . \tag{4.2}$$

Subject to typical constraints with as many constraints as there are data values as

$$y^j = \beta_0 + \beta_1 x_1^j + \beta_2 x_2^j + ... + \beta_n x_n^j; \quad j = 1, ..., m$$

where

y^j represents the experimental value of the dependent variable at point j.

$\mathbf{x}^j$ is the coordinate of the independent variable vector at point j.

β is the vector of model parameters.

If p takes a value of 2, the objective would be to minimize the sum of squares of the deviations. When p takes a value of 1, then the objective would be to minimize the sum of absolute deviations.

LSP can be used to optimize the function given in Equation 4.2 where the constant p takes the value of 1, 2 or any real value. LSP finds a solution to such problems without the need for linearization of any of the functions. It also provides designers with more freedom when deciding on the severity of the criterion for model fitting. In addition, LSP can easily estimate parameters for both linear and nonlinear regression models and, in fact, provides new perspectives on the parameter values found.

The type of sensitivity information on the model parameters, provided by LSP, is unique and has considerable potential value to an engineer seeking a representative model. The sensitivity information is provided by LSP in the form of the acceptance sets as the optimal solution is approached. This information is provided without any extra computational overhead and is also easily understood. More discussion on sensitivity analysis with LSP in connection with model fitting is discussed in Chapter 8.

Examples of regression models were taken from the literature and solved using LSP. In almost all the cases LSP confirmed the results in the literature. Test problems [Problem 3-17-0-0 and Problem 4-11-0-0, Appendix A] are representative of some of the nonlinear regression models solved with LSP. A 4% improvement over the solution cited in the literature was found with LSP's solution to [Problem 3-17-0-0, Appendix A]. While it took 19,842 function evaluations to meet the convergence criterion of $V_M \leq 1 * 10^{-5}$, this still took only 3 minutes with an 80386-33Mhz personal computer and this time is of little practical significance. Only 2,128 function evaluations were expended to meet a similar convergence criterion for the four variable problem [Problem 4-11-0-0, Appendix A].

Chapter 5

EXPERIENCE WITH LSP

Considerable experience was gained with LSP when the many test problems, both mathematical and engineering in nature, were solved during the course of this research. This experience showed that the output obtainable from LSP as the search progresses can give important information concerning the effectiveness of the search, possible adjustments to the search strategy when difficulties are experienced, as well as insights into the topology of the problem being solved. It has also helped to give a clearer indication of the influence of certain LSP parameters, such as the number of points in a level set on the number of function evaluations required for solution, on the reliability and efficiency of the overall search.

While the basic LSP algorithm with the recommended default parameter values is an effective global optimizer, it is natural to seek further improvement in performance. This chapter describes a variety of enhancements and adjustments which have been beneficial in solving some of the test problems and would be implemented under the direct control of the user.

5.1 Intermediate output, its presentation, and interpretation

One of the important contributions of this research in implementing LSP is the information which is displayed graphically on the computer monitor after every iteration. This feature was not anticipated but, as its potential value became apparent, it was explored and exploited in this research. Three principal types of output displays were investigated;

scatter diagrams of points in an acceptance set on two variable planes; plots of level set value-c against iterations; and the cumulative number of function evaluations against iterations. These can all be generated rapidly with little computational overhead and were provided without difficulty by the LSP software developed. These plots and their significance are described in detail in the next three sections. All of the diagrams used in this chapter are either derived from these displays or represent actual screen dumps of the displays.

5.1.1 Plots of x_i, x_j pairs

A set of two dimensional $x_i \sim x_j$ plots of the confirmed points in the current acceptance set are displayed on the computer monitor at each iteration. When the axes of these plots span the initial variable bounds, the cluster formed by these points indicate the size and location of the current acceptance set in the decision domain. The approximate acceptance set boundaries can be visually inferred from the cluster of points displayed. The existence of multiple local optima is suggested in these $x_i \sim x_j$ plots when the displayed points form distinct clusters with open spaces in between. It is quite common to observe that, after a few more iterations and an accompanying improvement of the level set value-c, the displayed points form a smaller single cluster. The existence of distinct clusters at some stage of the search and their disappearance at a later stage is indicative of local optima and LSP's ability to reject them. If there is some special reason to explore a particular local optimum, LSP can be rerun within a new initial cuboid enveloping the cluster of points of interest. Clusters may however exist in n dimensional space but not necessarily be evident in many or even all of the $x_i \sim x_j$ plots due to overlapping and the limited perspective of each of the two dimensional views. Thus while visual detection of clusters is sufficient to confirm their existence it is not necessarily the case that they will be detectable in this way. In section 5.1.2 other graphical evidence of the existence of multiple clusters is described.

Two plots derived from a single LSP run, but at different iterations, are shown in Figure 5.1 for a two variable problem and indicate the existence of multiple optima. The points form two distinct clusters, which indicates that there are at least two local optima. At this stage there is nothing to suggest whether the two clusters will lead to a single global optimum or to two or more equal global optima. After a few additional iterations the points around the upper left corner of Figure 5.1(a) completely disappear and all points in the acceptance set cluster around a single point, as shown in Figure 5.1(b). The disappearance of those points which had formed a distinct cluster at an earlier iteration implies the existence of a local minimum in that vicinity. As the LSP search progresses the displayed points coalesce about the solution and ultimately occupy only a single pixel in the graphical display.

Similar scatter diagrams plotted with axes which span the sides of the current cuboid, that is where the full length of the axes correspond with the current cuboid side, can also be displayed at any stage of the search. Such a plot gives a closer look at the pattern of points in the current acceptance set. These patterns may reveal useful information concerning, for example, the distribution in the decision space of near optimal solutions and the mutual relationships of each pair of the variable set. The significance of such indications will be discussed under the subject of sensitivity in Chapter 8.

5.1.2 Plots of cumulative function evaluations-N_f against iteration number-I

Many different schemes for counting the number of function evaluations incurred in nonlinear optimization appear in the literature. The count adopted in this research reflects both objective function and constraint function evaluations. When a point is generated in LSP it is first checked for feasibility against the set of constraints including the bounds on each variable. If the point is feasible then it is retained, or if infeasible, it

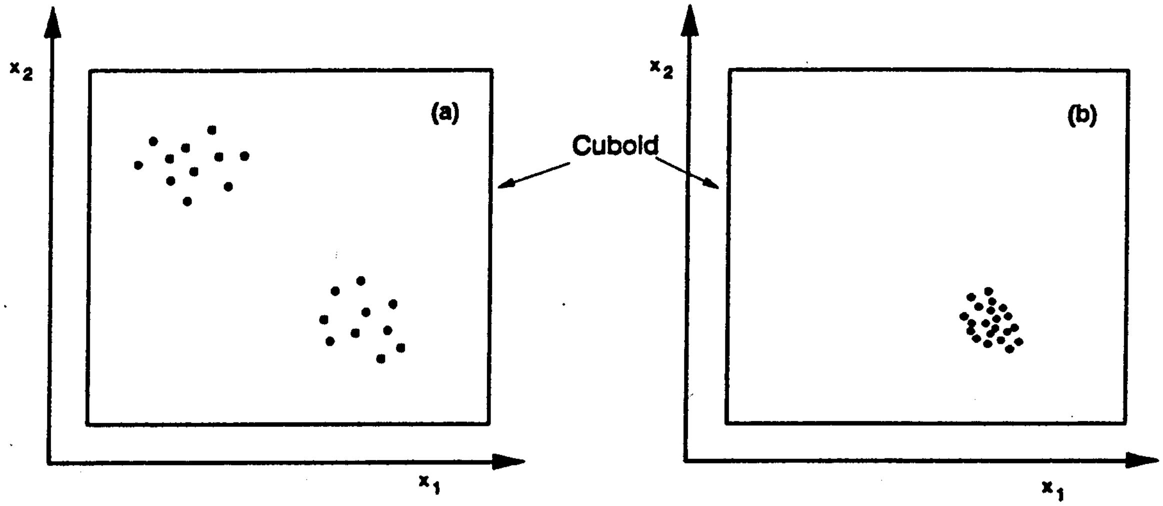


Figure 5.1: Local optima suggested by the disappearance of a point cluster.

is rejected. In either case this is counted as one function evaluation. At every feasible point the objective function value is calculated and this is also counted as a function evaluation. Consequently every confirmed point in an acceptance set will count as two function evaluations.

A variety of shapes of the $N_f \sim I$ curve can be observed. From experience with the test problems these curves have been categorized into three types: linear, exponential-like and sigmoidal curves. The implication of each type of curve is explained next.

<u>Linear curve</u>

The number of function evaluations at the first iteration gives an indication of the ratio of the feasible region volume and the initial cuboid volume. The number of function evaluations at the first iteration for an unconstrained but bounded problem should be equal to N_{keep}, that is equal to the number of points which are to be confirmed in the acceptance set.

For the ideal of an unconstrained, but bounded, problem with convex objective, the $N_f \sim I$ plot has been confirmed through the test problems to be almost a straight line.

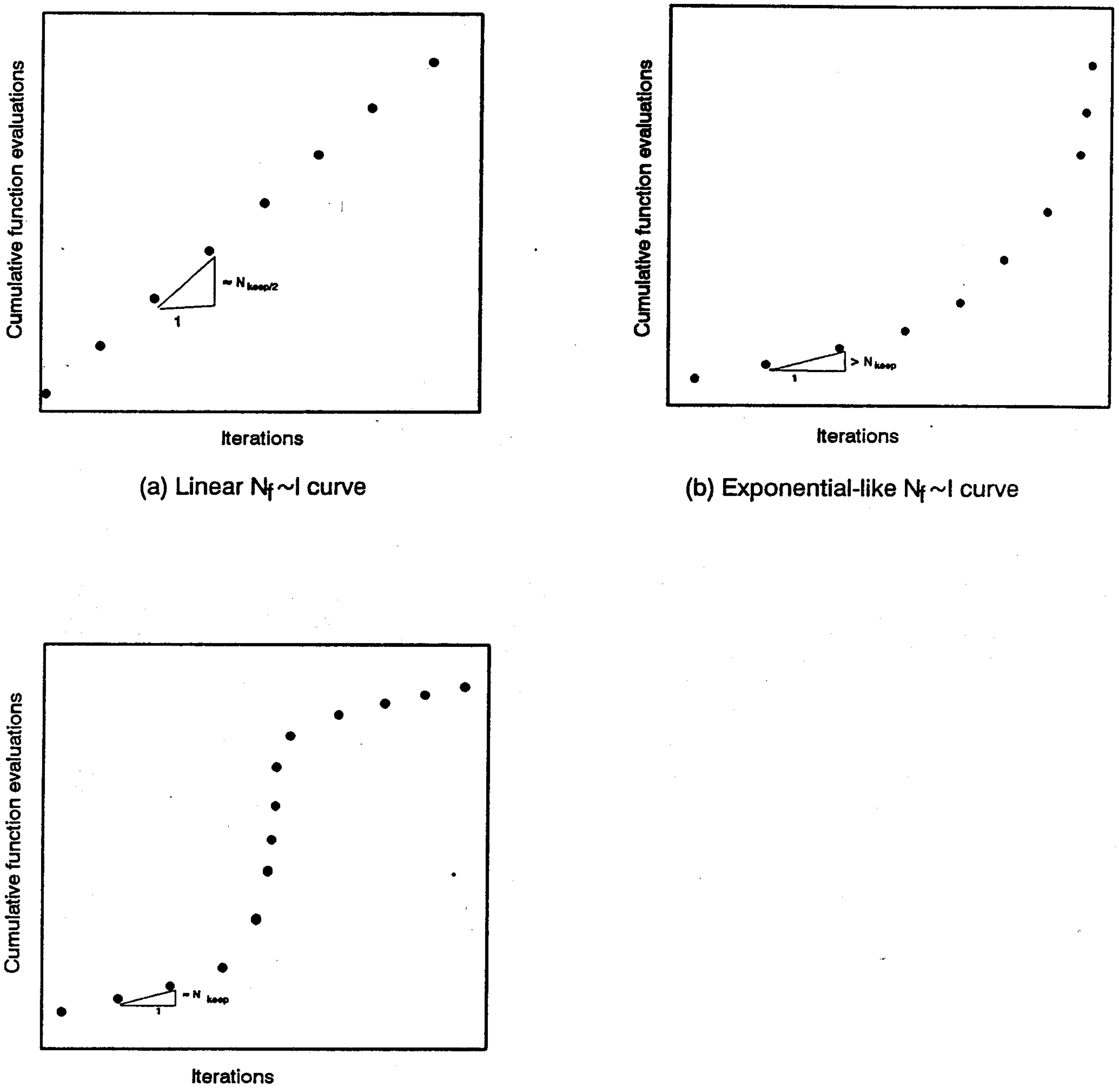
The slope of this line, the number of function evaluations per iteration, is only slightly larger than $N_{keep}/2$, which is shown in Figure 5.2(a). The reason is simply that, in this rather ideal case, only about half of the points in a level set are lost with the lowering of the level set value-c to $M(f, c)$ at each iteration. These lost points must be replaced by new ones at every iteration and each new point needs one function evaluation.

Exponential-like curve

When a problem has constraints in addition to the variable bounds then the number of function evaluations at the first iteration will be greater than N_{keep} as a result of the diminished feasible space within the initial cuboid. For constrained problems, using the uniform random trial point generation scheme described in Chapter 2, Section 2.3.2, the number of function evaluations increases as the ratio of the feasible region volume to the cuboid volume gets smaller. A progressive plot of the cumulative number of function evaluations against iteration number, $N_f \sim I$, indicates the effectiveness of the search at each iteration and any change in this efficiency as the search progresses.

For constrained problems the $N_f \sim I$ plot usually deviates from a straight line, and for a typical "well behaved" LSP problem, progressively steepens reflecting that generating new acceptable points gets more difficult with the number of iterations. Figure 5.2(b) is a plot for a typical constrained problem.

The inefficiency of acceptable point generation is indicated by the degree of deviation of the $N_f \sim I$ plot from the straight line of slope N_{keep}, each point requiring one function evaluation for feasibility; and one function evaluation for eligibility in the level set. This can occur when the problem has widely spaced multiple optimal points which produce a disjoint acceptance set or when the geometry of a connected acceptance set produces a large cuboid volume with small acceptance set volume, for example when the acceptance set is distributed around a cuboid diagonal.



(a) Linear N_f~I curve

(b) Exponential-like N_f~I curve

(c) Sigmoidal N_f~I curve

Figure 5.2: $N_f \sim I$ plots.

Sigmoidal curve

For some problems the $N_f \sim I$ plot may be linear for the initial iterations, then adopt an exponentially increasing shape for a few more iterations, and finally decline and continue as a straight line, as shown in Figure 5.2(c). Such a plot shows that generating points in the acceptance set was efficient at the earlier stages of the search, then lower efficiency was experienced for a while, but that the efficiency was restored in the later stages when approaching the final convergence of c on c^*.

A common explanation for such a curve is that, at the early stage of the search, the level set requirement is easily met and feasibility dominates acceptance set point generation. At this stage all points in the acceptance set constitute a single cluster, indicating that feasible region is not disjoint. Therefore the $N_f \sim I$ plot is close to a straight line with slope approximately equal to N_{keep}. At the intermediate stages the acceptance set becomes fragmented, the points forming distinct separate clusters each surrounding a local optimum. Since acceptance set point generation is inefficient with multiple optima, their existence is reflected in the $N_f \sim I$ curve by a steepened slope. But in the later stages the local optima are discarded and the search is performed on a single connected acceptance set. Such a curve shape is most pronounced when the objective function values at widely dispersed local optima are close to the global optimum objective function value. If the objective function value differences at all local and the global optimum are large, the influence of the local optima on the $N_f \sim I$ curve is small.

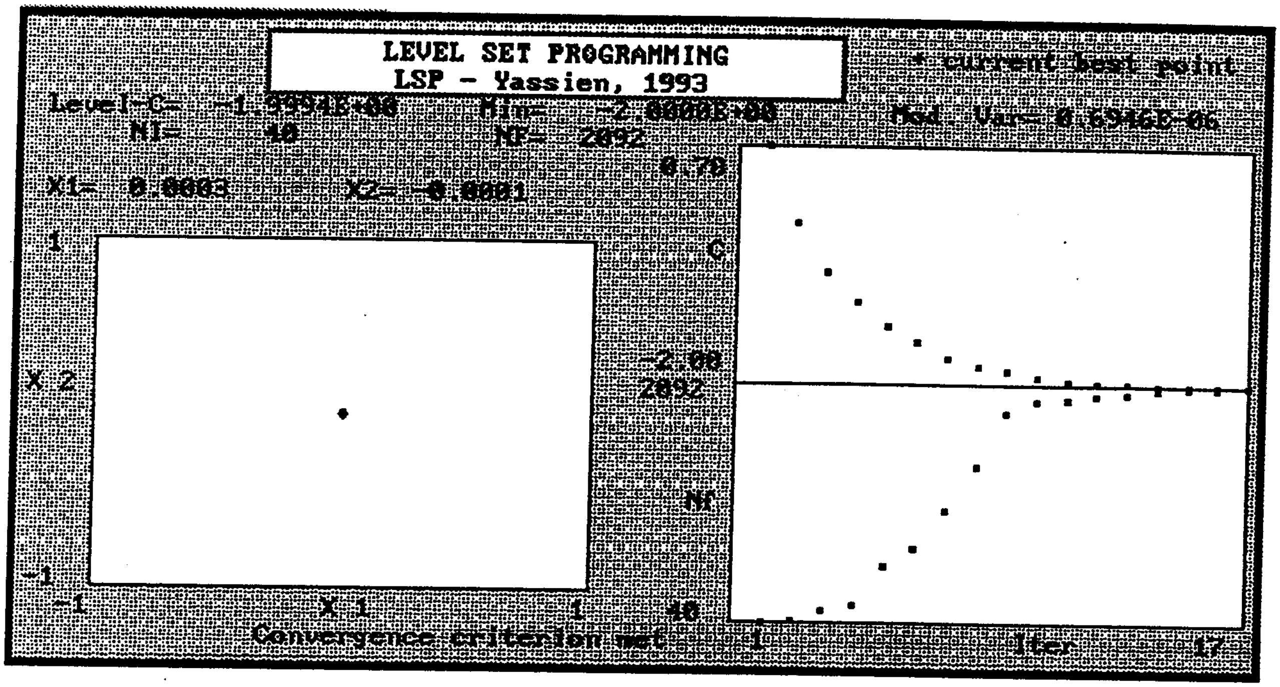
The sigmoidal curve can be demonstrated with an LSP solution of the Rastrigin function, [Problem 2-16-0-0, Appendix A], a two variable problem with 50 local optima but only one global optimum. This problem's formulation is

$$\text{Minimize} \quad f(\mathbf{x}) = x_1^2 + x_2^2 - cos18x_1 - cos18x_2$$

Subject to:

$$-1 \leq x_1 \leq 1$$
$$-1 \leq x_2 \leq 1.$$

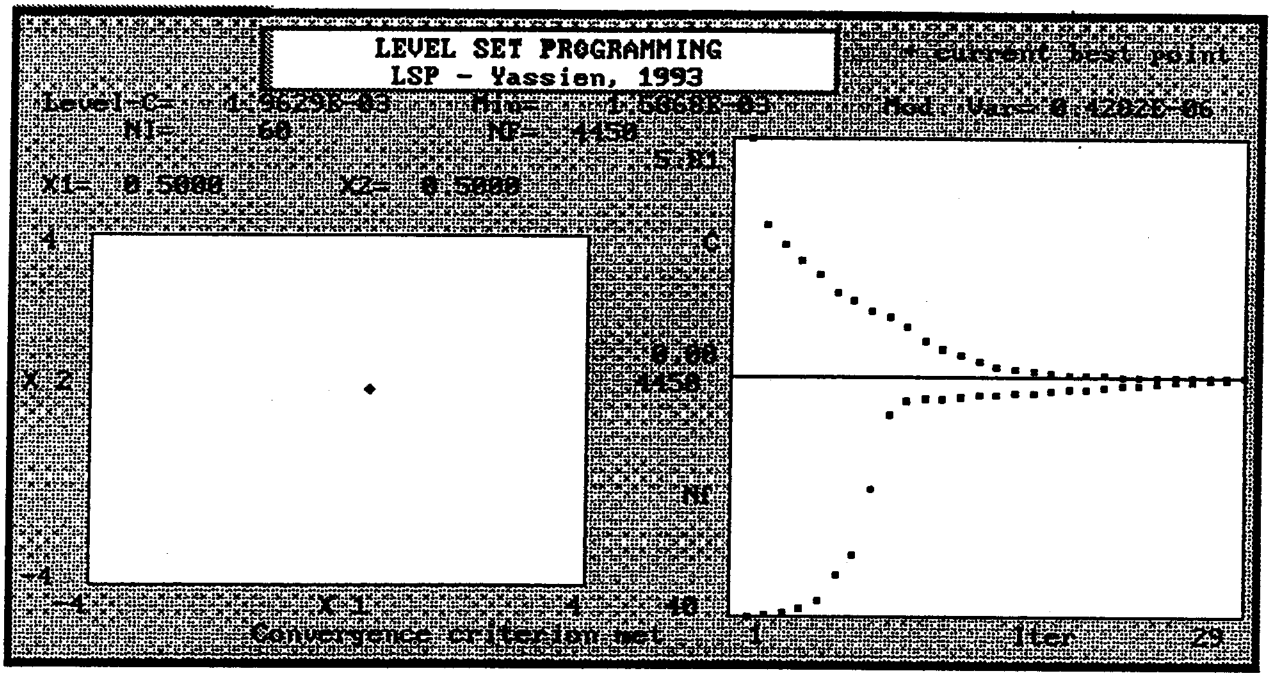
Figure 5.3 shows the screen display at convergence obtained from the LSP solution.



```
LEVEL SET PROGRAMMING
LSP - Yassien, 1993
                                                      * current best point
Level-C=  -1.9994E+00     Min=   -2.0000E+00        Mod. Var= 0.6946E-06
     NI=     40             NF=  2092
X1=  0.0003        X2= -0.0001
Convergence criterion met
```

Figure 5.3: Rastrigin function-Implied local optima by $N_f \sim I$ plot. (Actual screen dump).

A second possible reason for observing the characteristic sigmoidal $N_f \sim I$ curve arises when a fissure exists in the objective function surface in the feasible region at the global optimum. For a problem with more than one local optimum, points in the acceptance set can form clusters around the individual local optimal points but not necessarily include a point in the fissure. The ratio of the acceptance set and cuboid volumes may be small under these conditions and the search inefficient. Provided the cuboid at this stage includes the fissure region, then there is still a possibility of generating points at or near the global optimum. Once a single point is generated in the fissure, the search tends to converge rapidly on the global optimum as more and more points are generated within the fissure, so that the efficiency of the search is restored. The Road Runner function test problem [Problem 2-22-0-0, Appendix A] screen display is shown in Figure 5.4, and is an example where the $N_f \sim I$ plot exhibits the sigmoidal shape due to the presence of a fissure.



```
LEVEL SET PROGRAMMING
LSP - Yassien, 1993                        ◆ current best point
Level-C=   1.9629E-03   Min=   1.5668E-03       Mod. Var= 0.4202E-06
     NI=   60               NF=  4450
X1=  0.5000          X2=  0.5000
Convergence criterion met
```

Figure 5.4: Screen display for the Road Runner function, two dimensional case. (Actual screen dump).

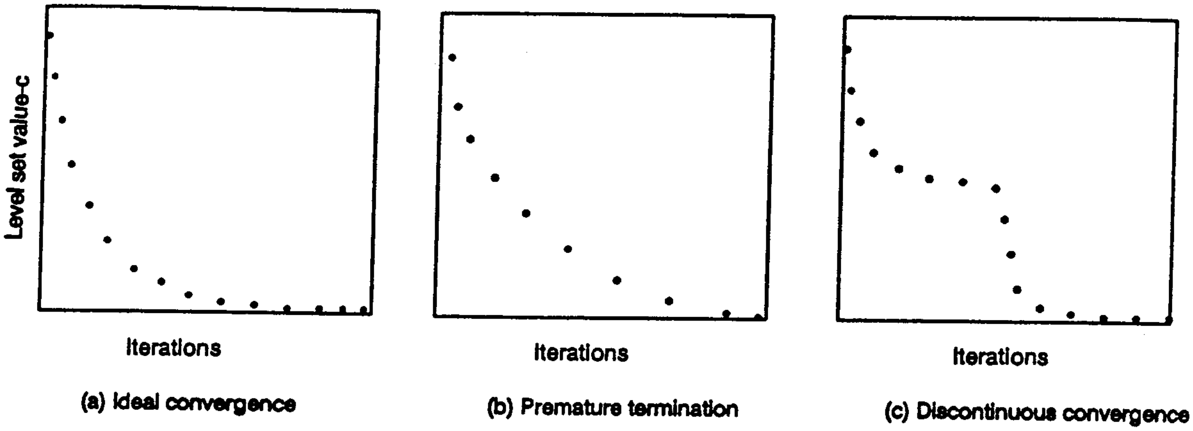
5.1.3 Plots of level set value-c against the iteration number-I

The progressive plot of level set value against the number of iterations, $c \sim I$, gives feedback on the strength of the convergence on c^*. In this research a pre-specified value of the modified variance was adopted as a convergence criterion. This value is compared with the modified variance V_M after each iteration. Ideally the modified variance should reach zero at the global optimum, but a small value is used as a tolerance for practical implementation purposes.

Because the modified variance is not dimensionless and is also sensitive to the absolute value of the objective function, a strong convergence criterion value for one case may be weak for another case. It can therefore be difficult to specify a convergence criterion in advance of obtaining a solution of the problem. The convergence question can be judged from the evolving $c \sim I$ curve and based on the difference of level set value between consecutive iterations, as was discussed in Chapter 2, section 2.3.5.

Ideally $c \sim I$ would be a smooth and monotone decreasing curve, and strongly asymptotic to the global minimum. The advancement of c on c^* resembles a binary search in some respects and therefore often shares similar convergence characteristics. As the curve becomes nearly horizontal it indicates that the difference between the level set values at consecutive iterations is very small, and that further iterations would probably bring even smaller improvement to the objective function per iteration, see Figure 5.5(a). But if the search stops before the horizontal part of the curve is well developed, it may suggest that the convergence criterion adopted is too large and has lead to a premature termination, as in Figure 5.5(b).

In less ideal cases the level set value versus iterations plot might show some discontinuity as in Figure 5.5(c). The causes for the discontinuity can be either a discontinuity in the objective function surface, or that the global optimum lies in a small fissure and a first point has been generated in that fissure just prior to the discontinuity. When a low point is generated in the fissure it focuses the search around the new point and a major correction occurs in the level set boundary estimates. As more points are then generated in the fissure the level set value drops rapidly compared to previous iterations, hence an abrupt change in shape is introduced into the $c \sim I$ curve. An example of the



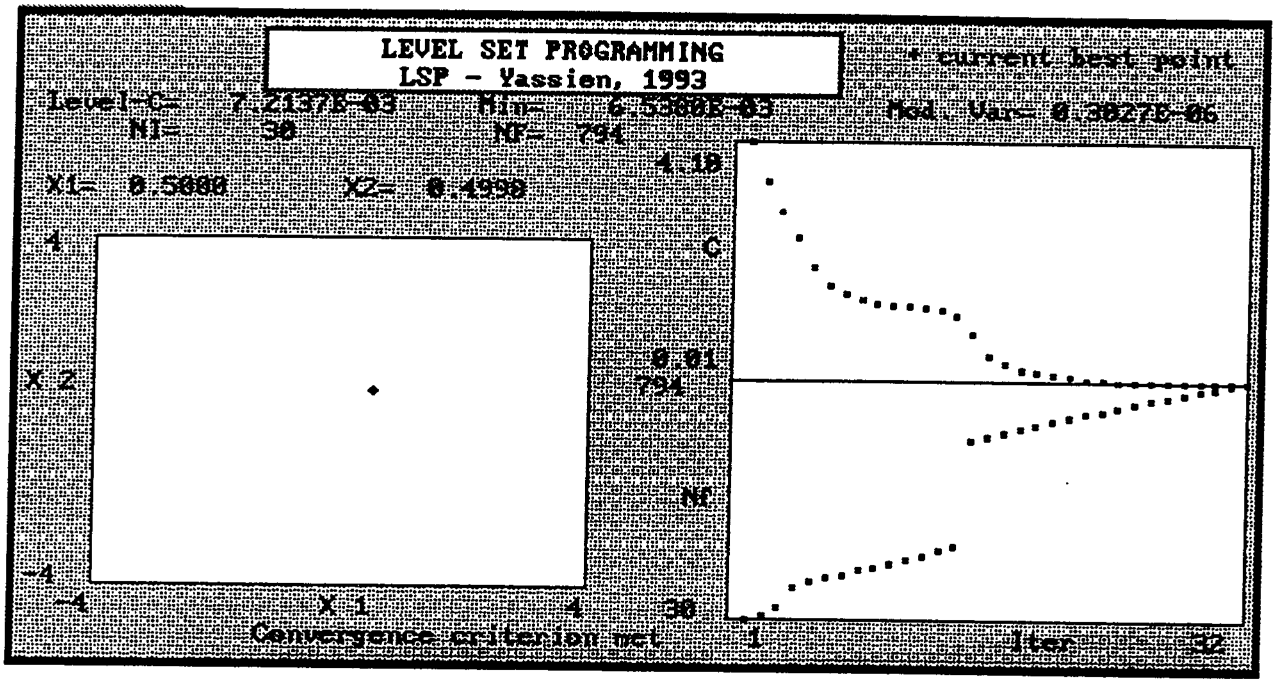
(a) Ideal convergence
(b) Premature termination
(c) Discontinuous convergence

Figure 5.5: Level set value-c versus iteration-I.

above is given in Figure 5.6, where the Road Runner Function is used to demonstrate a discontinuity in the $c \sim I$ plot. In this instance N_{keep}, the number of points maintained in the level set, was reduced to half of its recommended value. The curve exhibits the initial tendency to converge on a non optimal point, as discussed above, and then the abrupt slope change leads to the final convergence on the global optimum.

5.1.4 Further generalization of the $N_f \sim I$ and $c \sim I$ plot interpretations

The $N_f \sim I$ and $c \sim I$ plots are considered to be an important by-product of the LSP search and their interpretation is unique to LSP. Experience with these plots when solving problems has revealed some distinct and frequently observed patterns. These patterns have been linked to the known features of the many test problem solved and has lead to the following generalizations. While these are explained with the aid of a one dimensional problem, their key features can easily be extended to the multidimensional equivalent cases. It should be stressed that, regardless of the dimensionality of the problem being solved, the $c \sim I$ and $N_f \sim I$ plots will remain two dimensional. Interestingly, the characteristic plots shown in the following pages were clearly observed in many multidimensional problems. This suggests that the types of features reflected in



LEVEL SET PROGRAMMING
LSP - Yassien, 1993

Level-C= 7.2137E-03 Min= 6.5300E-03 Mod. Var= 0.3027E-06
NI= 30 NF= 794

X1= 0.5000 X2= 0.4990

Convergence criterion met

Figure 5.6: Low N_{keep} search for the Road Runner function. Demonstrates discontinuity in $c \sim I$ plot. (Actual screen dump).

the two dimensional examples are often also dominant in more complex problems.

The top two shaded diagrams in Figure 5.7 show plots of the objective function against the variable x for two single dimensional problems. The user would not have knowledge of these curves, but their general nature is revealed by the LSP screen displays of the $c \sim I$ and $N_f \sim I$ plots shown in the lower sections of Figure 5.7.

The plots of the level set value-c versus iterations are both simple monotone decreasing. This suggests for example that no fissure like feature in the objective function surfaces have been detected during LSP search.

In the unimodal objective function case of Figure 5.7, the $N_f \sim I$ plot, shows a linear relationship between N_f and I. The slope of the curve is close to N_{keep}, which is indicative of good search efficiency. This curve, together with the smooth monotone decreasing

$c \sim I$ curve, leads to the conclusion that the optimization problem being solved has only one global optimum and an objective function surface which is essentially unimodal.

In the multimodal case of Figure 5.7 the $N_f \sim I$ plot increases exponentially. This decline in efficiency is indicative of a progressive decrease in the volume ratio of the level set to the current cuboid which often arises when the problem being solved has multiple global, or near global, optima.

Figure 5.8 shows three sets of more complex $c \sim I$ and $N_f \sim I$ plots reflecting more complex objective functions. The first set, Figure 5.8(a), shows a smooth monotone decreasing $c \sim I$ curve and an almost constant slope of the $N_f \sim I$ curve except for a slightly higher slope at the initial stage of the search. These two curves are characteristic of problems with a set of local optima, and a single global optimum lying in a relatively broad depression in the objective function surface. Moreover the objective function value at the local points is significantly inferior to the value at the global optimum. If the objective values of the local optimal points are extremely inferior to the value at the global optimum, then the influence of the local optimal points on the $N_f \sim I$ curve is so weak that the higher slope around the initial stages may not be distinct. The objective function of Figure 5.8(a) is representative of this situation.

Figure 5.8(b) shows a smooth monotone decreasing $c \sim I$ curve but the $N_f \sim I$ plot has a characteristic sigmoidal shape. The likely indication from these two curves is that the problem has multiple local optima, but only one is a global optimum, and the difference between the values of the objective function at the local and global optima is quite small. This kind of objective function is depicted in the top figure of 5.8(b).

Figure 5.8(c) shows a pronounced discontinuity in the $c \sim I$ curve and the characteristic three part sigmoidal $N_f \sim I$ curve. Such curves are generally obtained with problems which have a fissure in the objective function surface. The fissure is contained within the cuboids throughout the search but no point is sampled inside the fissure at the earlier stages of the search. The type of objective function manifesting these plots is depicted at the top of Figure 5.8(c) for a single variable case.

5.2 Adjusting LSP parameters

With LSP there are a number of parameters which control key aspects of the search. These can remain fixed at recommended default values, be initialized at other values, or be adjusted as the search progresses in the light of the outputs discussed in section 5.1. The default parameter values recommended in sections 5.2.1 to 5.2.5 have proven to be effective in many cases, but when search difficulties are indicated by the $c \sim I$ and $N_f \sim I$ curves, adjustments to these values may be warranted. Each of the parameters impacts directly on quantities in the domain of the engineer's real world problem as opposed to being mathematically abstract.

The important LSP parameters are: the number of confirmed points in the acceptance set N_{keep}; the value of the criterion for convergence V_M; the value of the clustering criterion for partitioning the problem into subregions; the criterion for initiating skewness and the adjustment parameters; the change of the penalty parameters per iteration; and the tightening of any relaxed constraints per iteration. These parameter values and their adjustment, which were investigated and exploited throughout the test problems, are discussed in the next sections.

5.2.1 Number of confirmed points in the acceptance set-N_{keep}

The required number of confirmed points in an acceptance set depends primarily on the number of variables involved in the problem formulation. Generally, increasing N_{keep} improves reliability but reduces the search efficiency. The ideal relationship between N_{keep} and the number of variables in the problem is not necessarily linear, though a

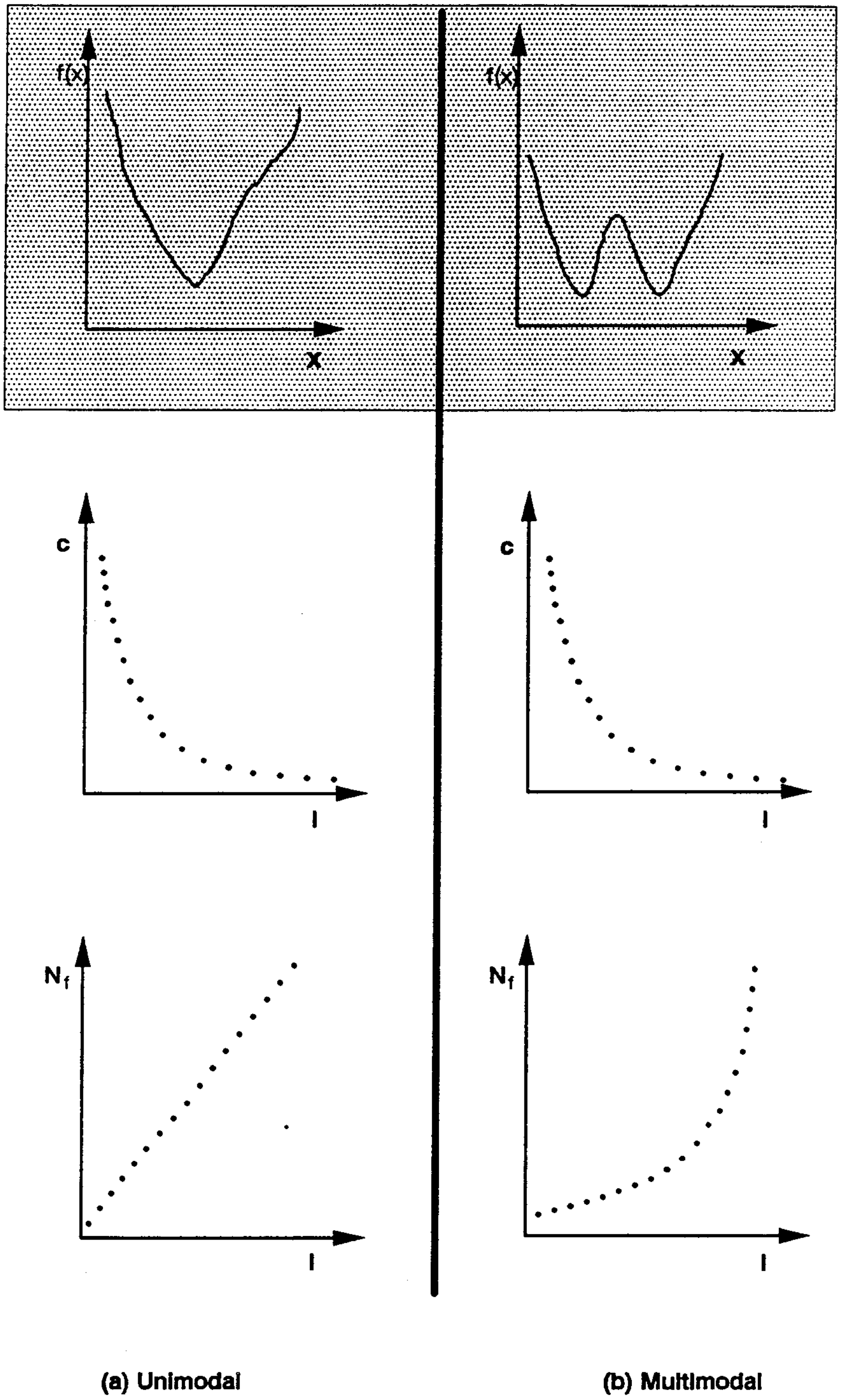


Figure 5.7: Interpretation of simple $c \sim I$ and $N_f \sim I$ curves, unimodal and multimodal objective functions.

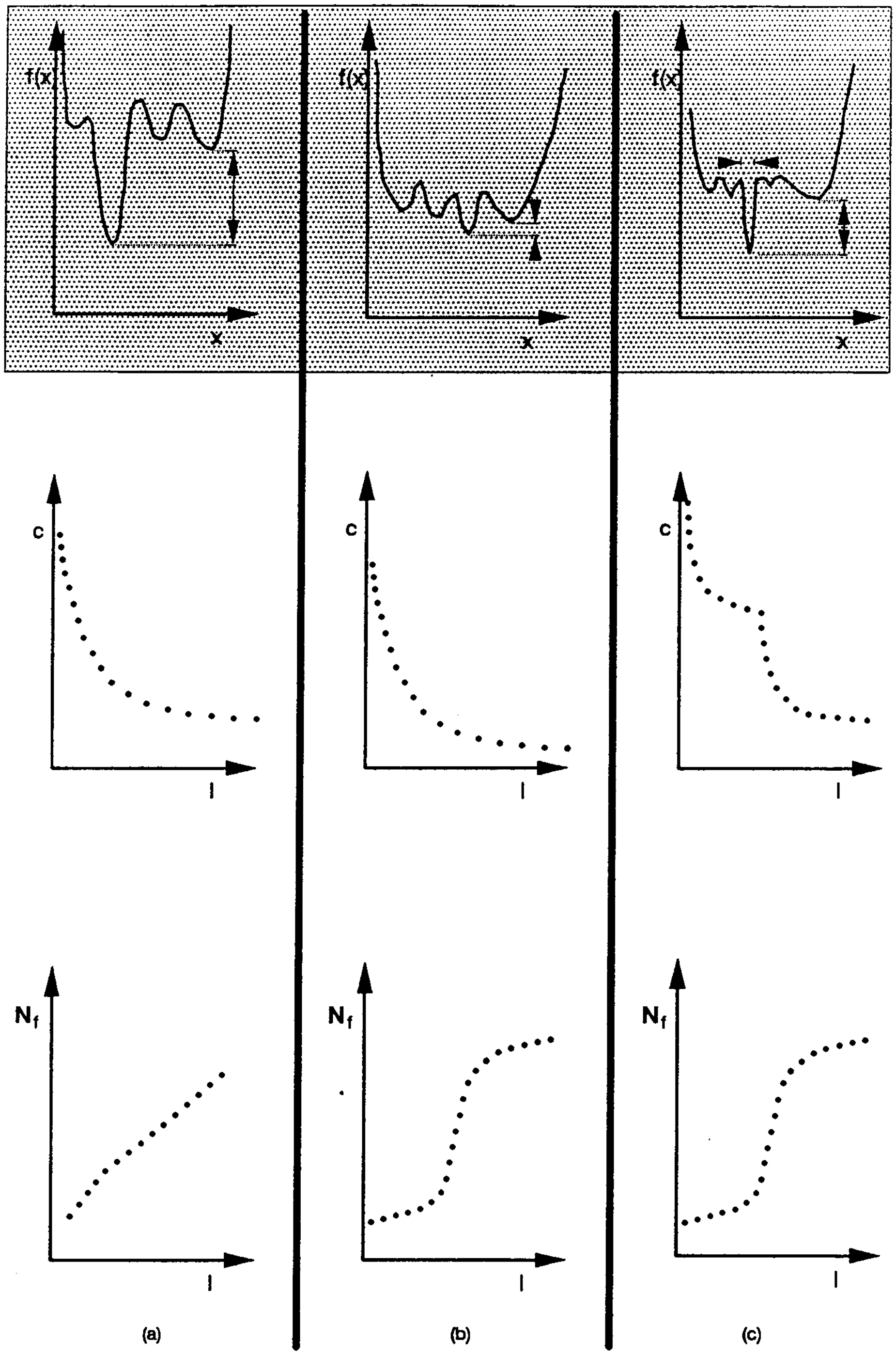


Figure 5.8: Interpretation of more complex $N_f \sim I$ and $c \sim I$ plots.

linear relationship appears to work well for lower dimension problems.

From the experience gained during this research, N_{keep} = (10×number of variables) was found to be adequate for most problems up to 6 variables. With larger dimension problems substantially less than the 10×number of variables was often satisfactory. For example, the maximum value of N_{keep} utilized was 160 for Templeman's problem, discussed in Chapter 4, section 4.3.4, involving 38 variables and yet the solution found with LSP improved (very slightly) on the published results for the same problem as well as finding multiple optima in a single run.

5.2.2 Termination criterion-V_M

Since the convergence criterion used in LSP is based on the variance of the objective function values in the acceptance set, it is expressed in the same units as the objective function and is prone to scaling problems. Therefore a specific value of the convergence criteria performs best for a specific range of objective function values. A similar scaling problem is common with gradient optimization convergence criteria. One gradient method, GRG2, recommends that the objective function and constraint functions be scaled to have absolute values greater than 10^{-4} and less than 100 for successful operation of its convergence criterion [Lasdon, 1982]. Convergence criterion values ranging between $1 * 10^{-3}$ to $1 * 10^{-8}$ were adopted with LSP. The higher values were used for problems with high absolute values of the objective function at their global optimum.

The modified variance of the objective function values in the level set is reported at every iteration. Its value can also be assessed in the light of the information on the progress of the search provided by the $c \sim I$ and $N_f \sim I$ curves as discussed in Chapter 2, section 2.3.4 and earlier in this chapter.

5.2.3 Cluster criterion

Subdividing the cuboid into a set of smaller cuboids when multiple optima are indicated by the formation of multiple point clusters can be an important strategy in achieving a computationally efficient and successful solution with LSP. A simple clustering analysis routine is adequate to identify the clusters and hence their boundaries. Any overestimation and even overlap of cluster boundary estimates does not threaten the success of the search. Once the bounds of each cluster and hence their cuboids are established, the search proceeds independently within each of these individual new cuboids.

The standard cluster analysis routine adopted in this research involves the construction of a dendrogram. At some distance from the branch ends, the dendrogram is cut by a line perpendicular to the branches so that the initial dendrogram tree is divided into a set of smaller trees. Points corresponding to each of the smaller trees are considered to form individual sets and are identified as clusters if the necessary clustering criterion is met. The clustering criterion was specified in Chapter 3, section 3.1.3 as the ratio of the minimum distance between two clusters and the maximum distance between two points within a cluster. When this ratio is above a certain predefined value, the region occupied by the acceptance set is subdivided into smaller subregions. Based on the test problems, experience showed that the useful range of the clustering criterion is between 2 and 8.

Even though clustering analysis methods can be used for any n dimensional problems, their application in this research was confined to problems involving up to 4 variables. This was largely because of the difficulties which occur in cluster investigation in larger dimensional problems. Furthermore, there are dangers in over generalizing the experience obtained with a limited number of test problems to high dimensional problems that might be encountered in engineering.

5.2.4 Cuboid stretching parameters

At every iteration the cuboid defined by the confirmed points in the acceptance set is stretched on each side by $\frac{1}{n_x}$ of its total length, where n_x is the number of points from the previous iteration remaining in the acceptance set after points which do not fulfil the current $f(\mathbf{x}) \leq c$ condition have been discarded. Details of the stretching parameters are given in Chapter 3, section 3.4.

The stretching of cuboids obviously increases the cuboid volume, but the efficiency of the search is not noticeably reduced by this extra volume when the stretch is small. Experience with the test problems showed that no significant difference in the total number of function evaluations exists for cases with and without cuboids being stretched. But the increased reliability benefits from cuboid stretching were significant, especially when optimal points were close to the cuboid boundaries.

5.2.5 Skewness adjustment parameters

The LSP parameters used for skewness adjustment have already been discussed in Chapter 3, section 3.5. The choice of value for each parameter is made so as to balance the increased speed of convergence against the increased risk of missing the global optimum. The parameters help reduce the risk of missing optimal points located at the boundaries, and in particular the corners, of the cuboid.

From the experience gained in this research, the best values for the three skew parameters range from 0.3 to 0.4 for δ_0, 0.10 to 0.25 for δ_1 and 0.05 to 0.15 for δ_2.

When skewness adjustments were used, the default values $\delta_0 = 0.4$, $\delta_1 = 0.1$ and $\delta_2 = 0.05$, were adopted. For a one variable problem with a unit length of cuboid, these figures give a maximum elongation (stretch) of 5% when $\delta = 1$ and a minimum elongation of 2% when $\delta = \delta_0$. For cases where $|\delta| \leq \delta_0$ there is no adjustment. With the default parameter values adopted, the shift of the cuboid would be a maximum of 10% when $\delta = 1$ and a minimum of 4% when $\delta = \delta_0$. Therefore for a cuboid of unit length the stretch introduced due to skewness adjustment will be in the range $[0.02, 0.05]$ and the consequent shift will be in the interval $[0.04, 0.10]$.

The above mentioned default parameter values produced satisfactory results throughout the test problems. As a test of response to a highly skewed feasible region, the initial bounds for the Rosenbrock function, where the solution is at $x_1 = 1$ and $x_2 = 1$, were set at $0 \leq x_1, x_2 \leq 20$. Two sets of runs were conducted, one without, and the other with, skewness adjustment. In the first case, out of the 10 different LSP runs without skewness adjustment, only 3 runs converged on the true optimum point. But, in the second case, where skewness adjustment was used, all the 10 different runs converged on the global optimum point. These examples, as well as a number of the other test problems, demonstrated that the skewness adjustment significantly improves the reliability of convergence on the true optimum point with only a small addition in computational effort.

5.2.6 Penalty and constraint relaxation and tightening parameters

Details of penalty parameter modification are given in Chapter 3, section 3.2.1. Experimental work showed good results for the penalty coefficient reduction within a range of 0.7 to 0.9 at every iteration. A minimum value of 100 was found to be satisfactory for problems whose absolute value of objective function evaluated at the optimal point is less than 100. Reductions outside this range encountered difficulties which commonly involved convergence on a non optimal point. This is also discussed in Chapter 3, section 3.2.1.

The relaxation parameter value was usually determined by trial and error while running LSP. In some instances, with heavily constrained problems, very little progress towards establishing the initial set of N_{keep} points occurred even after the expenditure of considerable computational effort. The constraints were then relaxed before completion of the first iteration. But, if possible, it is suggested that at least one iteration is completed so that an indication of N_f^1, the number of function evaluations needed to establish N_{keep} confirmed points, is obtained. If $N_f^1 \geq 20 \times N_{keep}$ one might think of relaxing the constraints. Generally the value of the relaxation parameter is related to the complexity of the constraint functions. The merits of relaxing constraints are discussed in Chapter 3, section 3.3.

The six variable problem [Problem 6-1-0-2, Appendix A] demonstrates the advantage of constraint relaxation. Without relaxing the constraints it took over 100,000 function evaluations to generate a single acceptable point. But with constraint relaxation, it was possible to find the optimal solution with 272,225 function evaluations.

From experience with the test problems, values between 0.5 and 0.85 were found to be reasonable for tightening relaxed constraints. A lower value can be used for problems where the search efficiency indicated by N_f after the first iteration is high indicating that the feasible region volume is close to the current cuboid volume.

5.3 Use of a rhombohedron shaped search domain

A rhombohedron can be used as a more compact search domain in place of a cuboid. The efficiency of LSP then increases in proportion to the volume ratio of the minimal cuboid to the minimal rhombohedron enclosing the same acceptance set. This is discussed in Chapter 3, section 3.6. This efficiency improvement increases with increasing linearity in the relationship between pairs of variables. The greatest efficiency gain occurs when the correlation coefficient approaches $+1$ or -1.

As the search becomes confined to a very small region of the decision domain, relationships which are highly nonlinear will often approximate locally to a linear relationship so that the rhombohedron can be exploited frequently. However, it is important to note that LSP does not impose an actual linear relationship on the variable pair so that no approximation or distortion is being introduced into the problem being solved.

The rhombohedron was used for a number of test problems which had shown distinct evidence of linearity between some of the variables in the $x_i \sim x_j$ plots at some stage in the search. The three variable flywheel design problem [Problem 3-15-0-2, Appendix A], and the four variable power generation problem [Problem 4-11-0-5, Chapter 4], were used to investigate the reduction in the number of function evaluations when the rhombohedron was used. Each of these test problems was solved in two ways. In the first case, a cuboid was used throughout the search while in the second case shifts were made between cuboid and rhombohedron as the need arose. The shift to the rhombohedron was made only when the absolute value of the correlation coefficient between any two variables was greater than or equal to 0.9. Both test problems were run three times for each case and the average number of function evaluations required for convergence is given in Table 5.1. The same termination criterion and N_{keep} values were adopted in all of the cases.

	Number of function evaluations	
Problem	Cuboid	Rhombohedron
3-15-0-2	7,432,678	507,558
4-11-0-5	1,431,317	84,653

Table 5.1: Function evaluations using cuboid and rhombohedron

In multidimensional problems more than one pair of variables can exhibit high linear correlation. In such cases the rhombohedron defined might have more than one non-rectangular face. Even though it is technically possible to make further modifications to the cuboid in response to multiple high correlation coefficients, only a single variable pair was accommodated at any one time in this research. When more than one pair of variables had correlation coefficients greater than 0.9 the cuboid was replaced with a rhomboid only in the plane of the variable pair with the highest correlation. At this present stage of implementing the use of rhombohedron both conceptualization and visualization difficulties preclude the use of a rhomboid for more than one variable pair.

The rhombohedron also provides benefits in other situations. When a problem has multiple global optima, the number of function evaluations can be very large before the convergence criterion is met, as discussed in section 5.1.4. If there are only two global optimal points, or there are more than two optimal points lying approximately on a straight line in the decision space, adopting a rhombohedron can dramatically improve the search efficiency. Therefore in some cases, the use of the rhombohedron can be a substitute for cluster analysis. For example, a rhombohedron search domain was used to solve the two variable, two global optima problem discussed in Chapter 3, section 3.1.3, using $N_{keep} = 20$ and $V_M \leq 0.001$. The convergence criterion was met and both optimal points identified after 234,866 function evaluations. Comparison of the result with the 8,018,142 function evaluations cited in Table 3.1 in Chapter 3 indicates that the efficiency gain is substantial.

5.4 Relationship between initial cuboid volume and its location, and N_f

When optimization problems are bounded but bound estimates are not known, the designer has to guess the possible upper and lower limits for each variable in order to define the initial cuboid. A generous guess is preferred since precision on the bounds is not required and it is important that the global optimum not be excluded. The size of the initial cuboid was found not to have a significant influence on the total number of function evaluations required to solve the problem.

A two variable unconstrained problem, [Problem 2-24-0-0, Appendix A], demonstrates the influence of the initial cuboid can have on the total number of function evaluations.

This problem was run with two different sized initial cuboids. In the first case the bounds of the variables used were $0 \leq x_1 \leq 4$ and $0 \leq x_2 \leq 4$, which gives a volume of 16 units. In the second case, where the bounds used were $0 \leq x_1 \leq 20$ and $0 \leq x_2 \leq 20$, the volume of the initial cuboid was increased to 400 units, i.e. 25 times that of the volume in first case. Ten separate runs for each of these cases were conducted and the average value of number of function evaluations was found to be 2,369 for the first case and 2,725 for the second, larger initial volume, case. These numbers suggest that there is only a weak relationship between initial cuboid volume and the total number of function evaluations required for solution.

Inappropriately skewed initial estimates of bounds relative to the global optimal point(s) can result in a high number of function evaluations for solution. The skewness adjustment generally modifies the location of the cuboids so that clusters of points in the vicinity of a cuboid boundary are not on the boundaries of subsequent cuboids. Even if the skew adjustment is high at the earlier stage of the search, it can take many iterations to shift the cuboid until the global optimum is included in the current cuboid. Such extra iterations introduce computational inefficiency. Of course the inefficiency can be reduced with a different set of skewness parameter values, but adjusting these parameters for each particular problem is not a feasible option as there will rarely be any *a priori* information on which to base the adjustment.

5.5 Some specific difficulties encountered while implementing LSP

Computational inefficiency and convergence difficulties were encountered by LSP in just four of the 200 published mathematical test problems attempted. LSP can fail to find the true solution to a problem when a cuboid at any stage of the search erroneously excludes a portion of the level set containing the global optimum and this is not subsequently recovered through cuboid stretching or skewness adjustment. Evidence of this difficulty was provided by a test problem taken from Subrahmanyam, [Problem 2-12-0-4, Appendix A]. This two variable, four constraint problem is shown in Figure 5.9. Its key features are a narrow crescent shaped feasible region formed by two parabolas and an almost parallel objective function surface to one of the variable axis. The formulation is

$$\text{Minimize } f(\mathbf{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

Subject to:

$$-x_1 + 13 \leq 0$$
$$-(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$
$$(x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$$
$$x_2 \geq 0$$

Subrahmanyam, [Subrahmanyam, 1989], solved this problem using the "Extended simplex method applied to constrained nonlinear optimization". He states that convergence was achieved at 833 function evaluations, though it is not clear if this number includes the evaluation of constraints or is only the number of objective function evaluations.

Attempts to solve the above problem with LSP, using the default parameter values, always converged on non optimal points. Since the global optimum lies at an acute corner of the feasible region, the cuboids at successive iterations excluded the global point for the reason discussed in Chapter 3, section 3.5. Using skewness values within the range recommended in this thesis, the skewness adjustment could not shift the cuboid far enough to include the optimum point before the convergence criterion was met. The skewness parameter values were easily tailored to solve this particular problem, values of $\delta_0 = 0.25$, $\delta_1 = 0.22$ and $\delta_2 = 0.15$ produced the correct global optimum. Such values would not be appropriate for general LSP use because of the severe reduction in efficiency that would result with most types of problems.

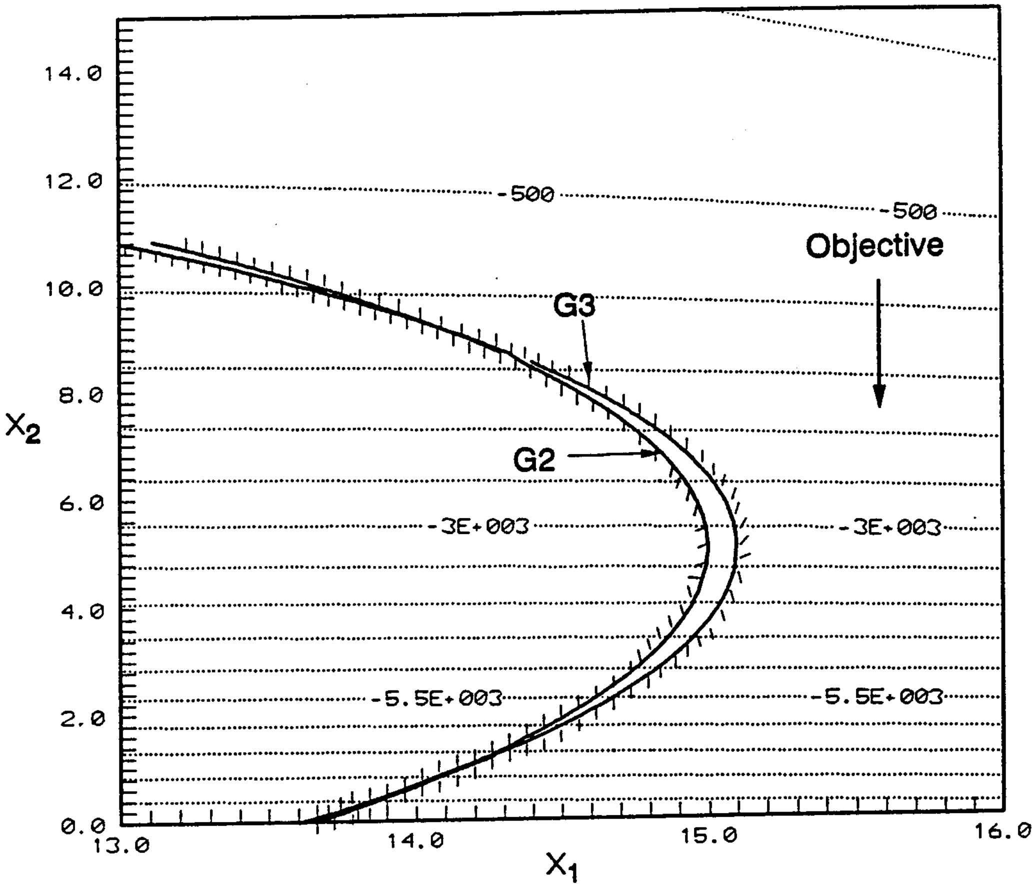


Figure 5.9: The crescent shaped feasible region of Subrahmanyam's problem [Problem 2-12-0-4, Appendix A].

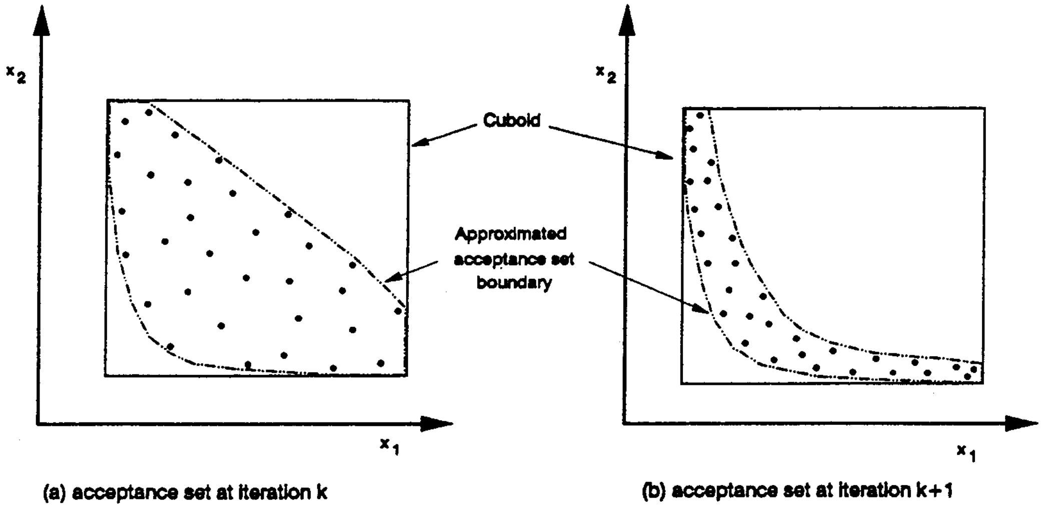
Subrahmanyam's problem suggests an even simpler test problem. As the only critical ingredient is the crescent shaped feasible region, the objective function can be replaced by a simple linear function with a mild gradient. It appears that, with its elegant simplicity, this modified version of Subrahmanyam's problem might play a similar role for level set methods in general as that played by Rosenbrock's valley problem for gradient search methods. A sample objective function for a modified Subrahmanyam's problem can be stated as

$$f(\mathbf{x}) = 245x_1 + 4292x_2.$$

Another particular difficulty arises with LSP when the shape of an active constraint surface is similar to the shape defined by the objective function surface in the vicinity of the global optimum value. An elementary example of this occurs in linear programming when the objective function is parallel to an active constraint. There, all points on the line joining the two neighbouring optimal vertices constitute a set of global optima. When solving nonlinear problems having this characteristic, the cuboid volume reduction between consecutive iterations can, from a practical point of view, cease. But the volume of the level set inevitably decreases at every iteration, even though the magnitude of this decrease may be small. Thus the ratio of the acceptance set volume to the cuboid volume gets progressively smaller with each iteration. As this ratio declines, the chances of generating acceptable points decreases, and the search becomes increasingly inefficient.

Another situation where a progressive reduction of the acceptance set volume occurs while the volume of the cuboid remains almost constant is shown in Figure 5.10 for an unconstrained problem. The figure shows that the cuboid volume, which corresponds to area in this two dimensional example, remains nearly the same at iterations k and $k+1$, while the volume of the acceptance set is substantially reduced. In this case the efficiency of generating acceptable points at the $k+1$ iteration declines proportionally with the acceptance set volume. Such a situation can, however, often be detected visually in the $x_i \sim x_j$ displays and remedied by partitioning.

Efficiency problems also arise with LSP when a problem has multiple global optima, and the global optimal points are approximately distributed along an elongated path. An example is provided by the problem shown in Chapter 3, Figure 3.5 when solved as a maximization problem, so that the problem becomes



(a) acceptance set at iteration k

(b) acceptance set at iteration k+1

Figure 5.10: Acceptance set volume change after an iteration while cuboid remains constant.

$$\text{Maximize } f(\mathbf{x}) = 100 - (x_1 + x_2 - 10)^2$$

Subject to:

$$0 \leq x_1 \leq 10$$

$$0 \leq x_2 \leq 10$$

The solution is $f(\mathbf{x}^*) = 100$, and all points on the line connecting (0,10) and (10,0) on the x_1-x_2 plane are members of the solution set. The final cuboid enveloping these points is the same as the original cuboid.

Under normal circumstances the on-screen plot of number of iterations versus the level set value, $c \sim I$, shows a monotone decreasing curve, but in some cases discontinuities in the curve are observed. This indicates that there is an abrupt change in the rate of improvement of the level set value. This phenomenon may arise from the fact that there is a fissure in the objective function surface in the feasible region which gives lower values of the objective function and no point was sampled in the fissure in any iteration prior to the discontinuity. There could of course be more than one fissure in the objective surface and consequently any discontinuity in the $c \sim I$ curve should signal caution about the validity of any presumed global solution.

5.6 Observations and recommendations

Prior information about a nonlinear optimization problem, such as the nature of the objective function surface or the shape of the feasible region, can be of help in the search for the global optimum. Realistically though, such *a priori* information is seldom available, even with engineering problems of modest complexity.

The information provided by LSP as the search progresses, as discussed earlier in this chapter, is simple in nature but gives important clues to the likely nature of the difficulties being encountered. As a result, possible remedies to overcome the difficulty, like relaxation of constraints or adjustments of other LSP parameters, can be suggested as the search progresses and before unnecessary computational effort is expended or the search abandoned.

In most of the test problems, the volume of the cuboid reduces and the density of points (i.e. points per unit volume in the decision space) in the acceptance set increases from one iteration to the next. In order to maintain a more or less constant density of points, the number of confirmed points in the acceptance set can be reduced at every iteration without impairing the convergence properties of LSP. A recommended refinement is that, unless the computational effort for a single function evaluation is very large, a higher initial N_{keep} value be used than that suggested in section 5.2.1. Initially, N_{keep} can be raised by about 30% of the recommended value and then be gradually reduced, by one or two, at every iteration, the reduction to stop when N_{keep} reaches about two thirds of the recommended (10×number of variables) value. When no other difficulties are experienced, the total number of function evaluations per iteration is proportional to the number of points in the acceptance set so that, while this strategy improves reliability the overall computational effort expended is unchanged.

The other heuristic adjustments discussed are modification of the penalty parameter and tightening of constraints following relaxation. These systematic adjustments are intended to increase the efficiency of generating acceptable points within a cuboid. There were instances where problems with feasible regions which were too small relative to the initial cuboid to successfully generate any acceptable points could only be solved using these approaches. Examples are [Problem 3-7-0-2, Appendix A] and the ten member truss problem in Chapter 4, section 4.3.3.

In LSP the number of function evaluations expended at each iteration are displayed in the $N_f \sim I$ graph and used to interpret certain phenomena. In general the $N_f \sim I$ plot serves as an indication of search efficiency as the search progresses. High numbers of function evaluations at the initial iteration indicates difficulty in generating points in the feasible region. Continuing the search under those circumstances, without modifying the LSP parameters, may result in an overall inefficient search or even an unsuccessful search. Therefore it is recommended that, in response to a high N_f value in the first or early iterations, the search be halted and the technique of constraint relaxation and tightening actuated in the LSP program. The fact that the need for such modification is usually indicated quite clearly by the output from the first iteration is of significant practical value as it limits wasteful expenditure of computational effort.

When the plot of the level set value versus iterations, $c \sim I$, shows a discontinuity at some stage of the search, the reasons for the discontinuity can usually be attributed to the detection of a narrow and deep fissure in the objective surface in the region containing the global optimum point. No point has been sampled in the fissure prior to the $c \sim I$ curve discontinuity but the acceptance set (so also the cuboid) still includes the global optimum. The subsequent generation of a single point in the fissure, with substantially lower objective function value, triggers the discontinuity in the rate of reduction in c over the current and following iterations. When such a phenomenon is observed, N_{keep} might be raised, say by about 50%, and LSP rerun so that the chances of generating points in any similar fissures, if they exist, is increased. The increase in N_{keep} will help to increase the chance of converging on the global optimum, but at the expense of extra computational effort.

In some cases, and especially with low dimension problems, the $x_i \sim x_j$ scatter diagrams of points in the acceptance set clearly indicate the existence of more than one cluster of points. In that case LSP can be halted and separate LSP searches initiated within cuboids bounding the individual subregions indicated by the clusters. The bounds of the subregions can be estimated visually from the scatter diagrams. Some additional graphical identification techniques are discussed in Chapter 7, section 7.5.2.

Finally it should be emphasized that, for all of the parameter adjustments discussed, much of the triggering information concerns just c, N_f and I. The magnitude of c is related directly to the problem being solved while N_f and I are easily understood quantities related to the search. The influence of these quantities on the search strategy is therefore not likely to be difficult for an engineer practitioner, as opposed to the numerical analysis specialist, to understand. The combined influence of a complicated set of constraints and a complicated objective function surface on the LSP search, even when a high number of dimensions are involved, can be interpreted from the set of two dimensional screen plots discussed in sections 5.1.1 to 5.1.3 far more easily than one might expect. In addition, the two dimensional $x_i \sim x_j$ scatter diagrams have the potential to provide an engineering practitioner with useful new knowledge concerning the topology of his design problem. The convergence criterion can be judged and adjusted in the context of the specific problem being solved and in light of the progress indicated in these plots. The plots also provide important clues concerning the modification of certain LSP parameters which can enhance computational efficiency and reliability while the search is in progress.

Chapter 6

EVALUATION OF NLP PERFORMANCE

During the course of this research into the implementation of a level set optimization scheme it became evident that it was evolving into an optimization tool with many features which might be considered attractive. At the same time it was recognized that, with the large number of solution points needed to be generated at each iteration, LSP might be considered to be computationally extravagant when compared with most gradient methods as well as certain direct search methods. Some kind of objective evaluation of LSP's overall performance and characteristics seemed desirable and the literature was searched for a suitable evaluation procedure.

There are a growing number of nonlinear optimization methods and many have been developed within very specialized fields of application. Diverse testing procedures and evaluation criteria have been presented but the problem remains of determining which test problems and evaluation criteria are relevant to an individual user needs. While some users are concerned primarily with speed and accuracy of convergence, others are more interested in reliably obtaining the true global solution.

No established universal criteria set for evaluating nonlinear optimization methods was found. As well, most of the test problems which appear in the literature, and have been used for evaluating performance, were developed to demonstrate the special strengths of one specific NLP method.

The performance criteria which have been offered in the literature are geared almost entirely to evaluating local optimization methods. The reason being that local optimization methods are by far the most prevalent. Since very few NLP methods have been developed for global optimization, evaluation criteria for global optimization methods are rare. Unfortunately, the criteria sets which have been proposed for local NLP methods do not consider some of the important features of global optimization techniques and the pitfalls of global optimization problems.

In this chapter some of the existing criteria for assessing global optimization methods are reviewed and a revised set proposed which, in particular, attempt to address the current needs of engineering practitioners. An assessment of LSP's performance under the recommended performance criteria set is also demonstrated in the last section of this chapter.

6.1 Existing evaluation criteria

Most of the existing performance indicators or performance criteria focus on the following factors: efficiency, expressed as the CPU time required to obtain a solution; the number of function evaluations for solution; and the numerical accuracy of the final results. The ability of a particular method to solve a wide variety of optimization problems is seldom considered and quantitatively measured since in the past efficiency has been the dominant consideration [Reklaitis et al., 1983]. Sandgren set as a criterion the ability to solve a large number of problems in a reasonable amount of time [Sandgren, 1980]. He ranked the algorithms on the basis of the number of problems solved within a specified CPU time limits. The limits were based on a fraction of the average CPU time for all algorithms to solve each of a set of problems.

Most of the criteria sets to date have been tailored to a specific algorithm. Regardless of the type of test problems used, the following performance indicators have consistently appeared in the literature and are therefore considered to be the most general.

- *Efficiency* (CPU time). A measure of the central processor time for termination of a successful search for a specific problem with a specified degree of precision. This indicator is highly dependant on the type of computer used and the coding of an algorithm.

- *Robustness.* The ability to solve a large variety of problems in a reasonable time with a specified precision.

- *Number of function evaluations.* A count of how many times the objective function and/or constraint functions are called during execution. There is not even consistent agreement as to how the number of function evaluations required to find a solution should be counted so the numbers quoted are often not comparable. This is an alternative measure of efficiency, which is independent of the type of computer used in the test.

- *Number of Iterations.* The number of iterations to reach convergence for iterative algorithms. This indicator is not influenced by the computer used, but it is not necessarily comparable between different algorithms.

- *Basic operation count.* How many basic computer numerical operations (or flops) are performed. Will be influenced by the code and may even depend on the type of computer used.

- *Numerical accuracy.* The deviation of the attainable solutions from the "true" or analytical solution. With dimensionally large and complex test problems the true solution may not be known.

- *User friendliness.* Simplicity, convenience of input format, ease of understanding of the output both during the search and at termination. A crucial component for practitioners, usually overlooked, is the time required by an occasional user to familiarize himself with the theoretical basis of the methodology and (re)acquire implementation skills.

- *Reliability.* The ratio of the number of successful solutions found to the total number of problems attempted. Wrong solutions involve either convergence on a non-optimal point, violating constraints, failure to meet the convergence criterion due to excessive computation time, or search termination because of computational fatal errors or numerical overflow.

- *Problem dimensional capability.* Expressed as the maximum number of variables and equality and inequality constraints which can be handled without substantially reducing the performance indicators, while the computational effort required remains within acceptable or feasible bounds.

- *Complexity.* The ability to handle ill conditioned, indefinite and degenerate problems.

- *Sensitivity to starting point.* Effect of choice of starting point on the success of the search.

Most attempts to combine the above criteria have been purely qualitative. Schittkowski produced one of the rare attempts to give a quantitative interpretation to the relative weight of each criterion [Schittkowski, 1980]. He adopted nine criteria and assigned the three sets of weighting factors given in Table 6.1 below. The choice of the criteria set depended on the relative importance of the various criteria to three distinct types of users. The actual nature of these types of users was not described but the emphasis on ease of use for type II users suggests that this is most applicable to the non-specialist.

Performance criteria	Alternate Weight		
	I	II	III
Efficiency (speed)	.32	.18	.14
Reliability	.23	.18	.36
Global Convergence	.08	.08	.20
Ability to solve degenerate problems	.05	.03	.03
Ability to solve ill conditioned problems	.05	.06	.03
Ability to solve indefinite problems	.03	.03	.03
Sensitivity to slight problem variation	.03	.03	.06
Sensitivity to position of the starting point	.07	.06	.06
Ease for use	.14	.35	.09

Table 6.1: Schittkowski's NLP performance criteria

6.2 Limitations of Schittkowski's performance criteria

A principal shortcoming of Schittkowski's performance criteria is that they do not address some of the currently important issues which impact upon the appeal and practicability of nonlinear optimization to the practising engineer, while considering others which are of little relevance. Limitations also arise from the fact that the criteria sets were designed to evaluate local optimization methods with continuous functions only. Therefore these sets of criteria and weights need to be extended and the weighting of each criterion needs to be adjusted to reflect contemporary needs and economic realities. Schittkowski's criteria weights were proposed when mainframe computers were the only computing resource and the cost of computation, which was directly proportional to CPU time, was the dominant expense. The development of microcomputers has substantially lowered the cost of computation. Furthermore, the marginal cost of computing on microcomputers can approach zero on machines which primarily serve other purposes and are severely under utilized over a 24 hour day. Thus the weighting of efficiency should be considerably lower today.

6.3 Recommended set of evaluation criteria

The purpose of a criteria set should be to serve as an aid to identifying the best optimization method for a particular class of user. Therefore, as many of the relevant practical considerations as possible should be addressed by the criteria set. The factors considered important in this thesis for NLP performance evaluation are as follows.

Reliability:

One of the most important qualities of an optimization method is its ability to reach a successful solution. A solution is successful when the final solution is achieved without numerical failure and without any violation of the constraints. Reliability is defined as the ratio of the number of successful solutions obtained to the total number of problems attempted. High reliability makes a method applicable, with confidence, to the widest possible range of problems, hence a high weight is given to reliability in the set of criteria.

Global convergence:

Many civil engineering optimization problems are nonlinear in their formulation which can lead to multiple local optima. Since the difference between local and global optimal solutions can be considerable, there is always the need to identify the global optimum. Therefore, a considerable weight is assigned for the capacity to guarantee finding the true global solution.

Multiple global solutions:

A practitioner would definitely be interested in identifying all possible global optimal points or at least being given some indication of their possible existence. These points may give designers and decision makers the flexibility to choose from significantly differing alternate optimal decisions. Consequently, the ability to identify multiple global optima is desirable and should have high weight amongst the performance criteria.

Near optimal solutions:

In many cases nonlinear problems have an undulating objective function surface which results in multiple local optima. All of the local optima may not necessarily be of great interest to a designer but those local points which produce objective function values close to the global optimum can, from a practical standpoint, be of equal interest to the global optimum. Therefore, a considerable weight should be assigned to near optimal solution identification.

Discrete variables:

Many civil engineering problems involve discrete variables as well as mixtures of discrete and continuous variables, so that the capacity to handle integer variables should be given some weight.

Ease of use:

Ease of use and interpretation of the output are major factors for choosing and using an optimization method. The general NLP user cannot be expected to know the meaning of all of the terminology used in numerical analysis. Terms like Kuhn Tucker conditions and Hessian matrix are often alien to him and there is a natural reluctance to adopt a method if the important features of the output data are not immediately understood. A friendly input-output format in a language the designer can understand is appealing. In practice, ease of use is related to expenditures of an engineer's time and hence to cost of implementation. This includes the time an engineer needs to learn and familiarize himself with the theory and implementation of the method and in the interpretation of the results. In this thesis, the highest weight in the performance criteria set is dedicated to user friendliness.

Speed of convergence:

As discussed in section 6.1, speed of convergence should no longer be considered as a major factor. It used to be important when the cost of computation was dominated by CPU time on a mainframe computer. The fact that some NLP methods can be run on personal computers has changed the relationship between cost and computation time. Microcomputers, which are often under utilized, can be left to run for the whole day and the computational cost is not far from the electrical power used. Moreover the speed of computing has continued to increase. This phenomenon has further diminished the importance of speed or efficiency as an indicator of NLP method performance. However, the number of function evaluations, which does not depend on the type of machine used, can serve as a relative indication of speed of convergence.

After considering the above factors, a new proposal for an NLP performance evaluation criteria set along similar lines to those originally proposed by Schittkowski is given in Table 6.2. This table is an extended version of Table 6.1. The criteria set and weights recommended are based on personal experience gained in using a variety of NLP techniques in this research. Each figure in the last column indicates the author's view of the relative importance of the various criteria for the present day needs of a practitioner engineer doing only occasional optimization.

6.4 Evaluation example: LSP versus GRG2

The performance of LSP under the proposed criteria set was not investigated in any systematic or formal way over the more than 200 test problems investigated in this research. Only a limited example of performance evaluation is given here, comparing

	Mainframe			**PC.**
	1980 Alternate Weight			**1993**
Performance criteria	I	II	III	
Efficiency (speed)	.32	.18	.14	**.03**
Reliability	.23	.18	.36	**.21**
Global Convergence	.08	.08	.20	**.15**
Ability to solve degenerate problems	.05	.03	.03	⇑
Ability to solve ill conditioned problems	.05	.06	.03	negligible
Ability to solve indefinite problems	.03	.03	.03	importance
Sensitivity to slight problem variation	.03	.03	.06	⇓
Sensitivity to position of the starting point	.07	.06	.06	**.03**
Ease for use	.14	.35	.09	**.25**
Integer handling	—	—	—	**.12**
Near optimal solution identification	—	—	—	**.08**
Multiple optima identification	—	—	—	**.13**

Table 6.2: Recommended NLP performance criteria

the performance of LSP against a leading gradient method GRG2 using the proposed criteria set. A single test problem, [Problem 4-12-1-5, Appendix A], is used in this example. This test problem was used by Schittkowski to evaluate the performance of a variety of NLP methods including GRG2 [Schittkowski, 1980]. In Schittkowski's work, the performance under each of his criteria was evaluated on the basis of 10 different runs. The same problem was run ten times using LSP. The results showed that all of the LSP runs converged at a single global optimum. The performances and scores of the two methods derived from the test problem are given in Table 6.3. The performance for each criterion is assigned values between 0.0 and 1.0, and scores are calculated as (weight × performance). The weights adopted here are those cited in Table 6.2, section 6.3. The total score, which can take a value between 0.0 and 1.0, is a comparative indicator of the overall performances of the two methods.

In Table 6.3, a performance value of 0.0 is assigned for efficiency to LSP and a value of 1.0 to GRG2. This is because the number of function evaluations to reach search termination with LSP is quite high compared to that reported for GRG2. On the other hand, it is assumed that the time an engineer needs to acquire a working knowledge of the gradient method and interpret its output is about four times the time required with LSP. Therefore, the performance values for ease of use are 1.0 and 0.25 for LSP and GRG2 respectively. The performance values for reliability and global convergence for GRG2 are from [Schittkowski, 1980].

Performance criteria	Weight	Performance		Score	
		GRG2	LSP	GRG2	LSP
Efficiency (speed)	0.03	1.00	0.00	0.03	0.00
Reliability	0.21	0.867	1.00	0.182	0.21
Global Convergence	0.15	0.654	1.00	0.098	0.15
Sensitivity to position of the starting point	0.03	0.00	1.00	0.00	0.03
Ease for use	0.25	0.25	1.00	0.065	0.25
Integer handling	0.12	—	1.00	—	0.12
Near optimal solution identification	0.08	—	1.00	—	0.08
Multiple optima identification	0.13	—	1.00	—	0.13
Total score				0.435	0.97

Table 6.3: Comparison of GRG2 and LSP using proposed criteria set

In general, with the proposed criteria set and weighting scheme, LSP showed superior performance over GRG2. The main factors for LSP's high performance rating are as follows.

- It is particularly easy to understand the output from LSP as it is expressed almost entirely in terms of the problem variables and their values (see Chapter 7 on LSP computer implementation).
- Almost all attempted problems were solved to a reasonable accuracy with LSP but GRG2 failed to find solutions in about 13.3% of the cases.
- Multiple optima are clearly identified by LSP while GRG2 identifies only a single solution at each run.

- A set of near optimal points always accompanies the global optimum solution with LSP but only one local optimum is provided by GRG2 at the end of each search.

Because efficiency still dominates many people's view of the value of an NLP method, and even though LSP was given a zero score in this category in the above evaluation, the number of function evaluations with LSP for most of the cases, were of the same order of magnitude as those solved using GRG2. Only a short time was required to solve these small sized problems. For example, with LSP, the time taken to solve the two variable Rosenbrock function is about 2.86 seconds [Problem 2-7-0-0, Appendix A], a 3 variable fuel allocation problem [Problem 3-14-0-1, Appendix A] took about 17.53 seconds, and about 90 minutes was taken to solve a 15 variable problem with 29 constraints [Problem 15-1-0-29, Appendix A]. These times were on an 80386-33Mhz microcomputer. The times were averaged over five different runs.

Because of the difficulties in finding representative test problems and finding universal performance measures, a single quantitative assessment of the performance of an NLP method does not appear to be possible. In addition, ease of use, which is an important factor in determining if an NLP package is ever to be adopted by non-specialist users, is necessarily a subjective issue.

This chapter has provided a reassessment of Schittkowski's set of qualitative performance criteria to reflect the present day computing environment and the needs of an engineer practitioner. Although no systematic and complete evaluation of a number of NLP packages was attempted in this research it is evident that LSP has characteristics which make it worthy of serious consideration as a tool for nonlinear optimization.

Chapter 7

LSP COMPUTER IMPLEMENTATION

LSP was written in Quick Basic. This language was adopted for convenience in the early stages of the research with the assumption that a more powerful language would be adopted when the need arose. It transpired that LSP does not place any special programming or computational demands so that, for developmental purposes, there was no incentive to change from Quick Basic. While it eventually became apparent that the graphical interface could be a unique and attractive factor of LSP it also presented minimal demands on the graphics capabilities of Quick Basic. The only justification for reprogramming would simply be to maximize computational speed and refine the appearance of the graphics displays.

LSP executes the optimization process iteratively, giving results at the end of every iteration. This intermediate output is available as numerical data and also in the form of graphical displays. When the convergence criterion is satisfied, the program terminates and gives the points in the acceptance set and their corresponding objective function values. All of the intermediate and the final results can also be saved in a file.

Normally LSP does not require any adjustment to its parameters, it performs the whole search automatically and outputs the intermediate and the final results. Modification of penalty terms, tightening of relaxed constraints and adjustments to the number of points in the acceptance set are usually done without user interference. The predominant user interaction is in the form of requests for specific information during the search, such as displaying particular types of plots or sending information to a printer or to a file, or when the search experiences difficulties.

7.1 Subroutines

The LSP package subroutines are categorized into three main groups. The first group consists of the problem definition routines, which contain the objective and constraint function code and the LSP search parameter values, and also process user instructions for any LSP parameter changes. The second group consists of the routines which perform the optimization. The third group consists of the output routines which generate the graphics displays and numerical solutions at various stages of the search.

LSP lends itself to programming as a set of short subroutines. The purposes of each subroutine are discussed in more detail below. A schematic representation of the general functions and operations of each set of the subroutines is also given in Table 7.1.

Problem definition subroutines:

The user is required to furnish information concerning the problem being solved. The subroutines where this information is coded or entered are:

CONSTANTS: All LSP parameters and constants describing the problem size are stored in this subroutine. These include the number of points in an acceptance set, the value of the termination criterion, the value of the clustering criterion, values of the skewness adjustment parameters, and the number of variables in the problem formulation.

INITBOUNDS: The minimum and maximum possible values for each variable are coded into this subroutine and on execution these values are passed to the main program to define the initial cuboid.

PROBLEM DEFINITION

- formulate objective function
- formulate set of constraints
- specify variable bounds

OPTIMIZATION

Initialization

- generate feasible points
- evaluate moments
- specify level set value

Algorithm

- check for termination
- revise level set value
- discard bad points
- define current cuboid
- generate new feasible points

OUTPUT

- Display:
    - points in the current cuboid
    - current level set value
    - current cuboid boundaries
    - computational effort expended
    - scatter plots
      (in both initial and current cuboid)
    - efficiency and search progress curves
- send information to a printer and/or a file

Table 7.1: LSP's problem definition and operations

CONSTRAINTS: All constraints are coded in this subroutine. It receives the $\mathbf{x}$ vector for a point and returns a binary value indicating feasible or infeasible. Constraint evaluation at a point is terminated when the first violated constraint is detected so that generally, on average, only half the constraints will need to be checked before rejection occurs.

OBJECTIVE: The objective function is coded in this subroutine. It receives the $\mathbf{x}$ vector and returns the value of the objective function.

Optimization subroutines:

This set of subroutines perform the actual optimization task.

INITKEEP: In this subroutine trial points are generated, points are checked for feasibility by calling the CONSTRAINTS subroutine and the objective function is evaluated by calling the OBJECTIVE subroutine. The number of confirmed points generated is controlled by the value set for N_{keep}.

UPDATEMSD: The mean and variances of the objective function values at the points in an acceptance set are evaluated. A new level set value-c is assigned for the next iteration.

SWAPSKEW: Points with objective function value greater than the new level set value-c are discarded. A new cuboid, which contains those points fulfilling the level set condition, is defined. Modifications of the new cuboid, in the form of stretching and skewness adjustments are done when the necessary criteria are met.

FILLKEEP: More points are generated in the new cuboid to replace those points discarded at the new level set value-c. The generated points are checked for feasibility and against the new level set value-c before they are accepted.

Output subroutines:

Some of the subroutines give the intermediate and final results as an output. The output is given in the form of numerical values and/or in the form of graphical displays. The user has the option to supply the format required for the output.

REVIEWKEEP: This subroutine prints intermediate and final numerical results on the screen. The current level set value-c, the best point $\mathbf{x}$ vector found so far and its objective function value, the total number of function evaluations and the modified variance of the points in the acceptance set are displayed after every iteration.

KEEPOUT: This subroutine prints the decision variable values and objective function values of the confirmed points in the acceptance set on the screen at any time requested. This subroutine is activated at any time, while the search is in progress, by pressing a function key.

DATA: This subroutine sends the decision variable values and the objective function values of points in the current acceptance set, the bounds of the current cuboid, the number of function evaluations expended so far and the current level set value-c to a file whenever requested. Data transfer is activated by pressing a function key.

PLOTS: Generates the various types of screen plots discussed in this thesis. Some of them are prompted by pressing a specific function key, and others are displayed automatically after each iteration. The $N_f \sim I$ and the $c \sim I$ plots are always displayed after every iteration.

SCRGRAB: This subroutine sends graphical information into a specified file by saving the currently displayed screen. It can be invoked either by pressing a function key or automatically after every iteration.

CLUSTER: The clustering analysis subroutines calculate the dendrogram using the confirmed points and displays the distance coefficients on the screen. The user is asked to specify the number of clusters extracted on the basis of the dendrogram. Once the clusters are identified the confirmed points are reclassified into the different clusters and the cuboid bounds for each cluster calculated.

The LSP program framework can readily incorporate auxiliary subroutines which solve major tasks that are associated with a specific type of problem. For example, the cluster analysis routine, the Hardy Cross relaxation routine for solving pipe flow problems, and frame and structural truss analysis routines have all been incorporated into the LSP program in the form of subroutines when solving test problems.

7.2 Confirmed point generation strategy

A point has to be both feasible and produce an objective function value less than or equal to the level set value-c to enter the acceptance set. Guaranteeing the feasibility of a point before evaluating the objective function or guaranteeing the level set condition before checking feasibility leads to two different strategies of confirmed point generation. These different strategies can affect the efficiency of the search depending upon the nature of the optimization problem being solved. Even though it is not necessary to check point feasibility for unconstrained problems it is relevant, in the following discussion, to consider both constrained and unconstrained problems. Constrained problems can yield effectively unconstrained interior solutions as the search progresses. At some stage of the search the current cuboid might remain entirely within the feasible region, consequently from the LSP perspective, the problem has become unconstrained. Confirmed point generation in such a situation is discussed below under the title 'Interior solutions with multiple optima'. Two distinct situations and their recommended point confirmation strategies are discussed here.

- *Interior solutions with multiple optima:*

  How often trial points generated in the current cuboid violate the level set condition is inversely proportional to the volume ratio of the acceptance set to the current cuboid. In the common situation of global solutions lying at interior stationary points, the rejection of trial points due to violation of the level set condition increases as the search approaches the optimum value. The reason being that the volume of the cuboid remains close to constant while the acceptance set volume reduces to small regions surrounding the optimal points. This situation lowers the efficiency of the search.

  The efficiency is therefore often related to closeness of the level set value-c to the optimum value, but is also affected by the dispersion of the optimal points in the decision space. For example, for a two variable problem with three global optimal points, the efficiency is low if these three points are on a straight line and the line is parallel to one of the variable axes. The efficiency decreases further when the line joining the three points sits on a diagonal of the cuboid.

  As an experiment, a problem was chosen with the acceptance set volume at the k^{th} iteration occupying 10% of the current cuboid volume. This meant that, on the average, 10 trial points were being generated to get a single confirmed point. The number of function evaluations necessary to generate 10 confirmed points was then investigated.

  The two strategies were examined. In the first case, generating a feasible point and then calculating the objective function. In the second case, the steps were reversed and a point which fulfils the level set condition was first generated and then its feasibility checked. The same experiment was done again after few more iterations where the volume of the acceptance set had been reduced to only 1% of the current cuboid. The number of function evaluations for the two strategies at these two iterations are given in Table 7.2.

a. When acceptance set is 10% of the cuboid volume			
Strategy	Function evaluations	Strategy	Function evaluations
Check Feasibility	10 × 10	Evaluate Obj. function	10 × 10
⇓ ⇓		⇓ ⇓	
Evaluate Obj. function	10 × 10	Check Feasibility	10
Sum	200		110

b. When acceptance set is 1% of the cuboid volume			
Strategy	Function evaluations	Strategy	Function evaluations
Check Feasibility	10 × 100	Evaluate Obj. function	10 × 100
⇓ ⇓		⇓ ⇓	
Evaluate Obj. function	10 × 100	Check Feasibility	10
Sum	2000		1010

Table 7.2: Comparison of N_f for two different point confirmation strategies

The above examples support the general experience with the test problems that, if feasibility is checked before the level set condition, the number of function evaluations required for search termination is about twice the number required for the reverse strategy. This advantage may not necessarily be enjoyed for all problems solved with LSP but appears to be the best default strategy.

- *Constrained problems - general case:*

  Inefficiency arises when the acceptance set volume to the cuboid volume ratio is low. The acceptance set for a constrained problem is governed by both the set of constraints and the level set condition. Which of the two factors, that is either the feasibility or the level set condition, is the most likely to cause rejection of a trial point depends upon the nature of the objective function and the complexity of the feasible region in a specific problem.

  In general, the minimum total number of function evaluations is always achieved by checking the dominant rejecting factor first. In practice this means checking feasibility first in the case of highly (tightly) constrained problems, and evaluating the objective function first for problems whose objective function surface is very irregular. Unfortunately, identifying the dominant factor at any stage of an LSP search is not always an easy task.

  Consider an example, where at some stage of the search, the feasible region and the level set occupy 30% and 80% of the current cuboid volume respectively. Assume that the acceptance set, the intersection of the two regions, is 20% of the current cuboid. Table 7.3 shows the number of function evaluations needed, in this example, to generate 10 confirmed points using the two different strategies. The strategy which generates feasible points before evaluating the objective function required less function evaluations since the feasible region volume is smaller than the level set volume.

Checking the feasibility of a single point might consume significant CPU time if the constraints are many and complicated. For example, a 4 variable problem with 3

Strategy	Function evaluations	Strategy	Function evaluations
Check Feasibility	5×10	Evaluate Obj. function	5×10
⇓ ⇓		⇓ ⇓	
Evaluate Obj. function	1.5×10	Check Feasibility	4×10
Sum	65		90

Table 7.3: Comparison of N_f for different strategies for a constrained problem

inequality constraint functions [Problem 4-1-0-3, Appendix A] was used to assess the typical time spent for checking feasibility and for evaluating the objective function. The average time taken to calculate the constraint functions at a feasible point was about 2.64 times the time required to evaluate the objective function. However, no matter how long the computation takes, it is simply counted as a single function evaluation in this research (a rationale for this was discussed in Chapter 5, section 5.1.2). Therefore the relationship between CPU time and number of function evaluations is not always linear.

The same test problem was used to compare the total number of function evaluations to meet the convergence criterion with both strategies, i.e. with checking feasibility before evaluating objective function or with the reverse strategy. The problem was run 10 times for each case. The results showed that the strategy which evaluated objective function prior to checking feasibility required 30.7% extra function evaluations than the strategy which generated feasible points before evaluating objective function.

In the discussion above, the superiority of one strategy over the other is based purely on the total number of function evaluations. From the experience gained with the test problems, it was generally found better to check feasibility before evaluating the objective function. This was basically due to the computational effort required to evaluate a set of constraint functions for most of the test problems. It is possible to switch from one strategy to the other without difficulty so that a choice can be made after appraising the relative computational complexity of the various functions involved in a specific problem.

7.3 Intermediate results presentation

After each iteration, the new and improved level set value-c, the best point observed so far, and the bounds of a new cuboid are all displayed on the screen. The results at any iteration will always show some improvement over the previous iteration. Even if the program is interrupted at the middle of the search, the effort expended is not wasted as the intermediate solution can be used to provide initial cuboid data for a subsequent reviewed attempt to find the solution.

There may be justification to modify the values of some LSP parameters on the basis of the display indicating search performance trends. If it is necessary to make any changes the search can be paused, the necessary changes made and the search then resumed.

All points in the acceptance set, the objective function value at each confirmed point, the level set value-c and the number of function evaluations at every iteration are stored in the computer memory. At any stage of the search the results provided at every iteration, from the beginning of the search up to the current stage, can be displayed and reviewed without stopping the search.

A variety of information is given out by the plots at every iteration. The set of scatter diagrams of points on a two variable plane are displayed by pressing a function key. The scatter diagrams are plotted with a choice of axis lengths, either the initial variable bounds or, alternatively, the current cuboid lengths. The $c \sim I$ and $N_f \sim I$ plots, are automatically displayed after every iteration.

If the intermediate results, including the plots, give an indication that there are multiple optima and efficiency is low, it is better to stop the search and divide the current cuboid into subregions so that the search can be carried out in each subregion separately. Such user intervention increases the overall efficiency. LSP could be programmed to automatically make the necessary changes without the need for sophisticated detection routines. It must be stressed that the need for such adjustments can be recognized by only a moderately experienced user.

7.4 Final output

The final output from LSP gives the global optimal point(s) and their corresponding objective function values and a set of alternate near-optimal points whose objective function values do not exceed the final level set value-c. This information is available both numerically and graphically. Certain performance indicators, like the total number of function evaluations and the number of iterations, are given as a supplementary information.

7.5 Alternative presentation of scatter plots

The most informative graphical representation of intermediate and final results is in the form of the scatter plots of confirmed points in the $x_i \sim x_j$ planes. When dealing with problems involving more than two variables, a set of scatter plots are required to represent the points in every plane of the variables. Various ways of presenting these plots have been assessed during the course of this research. Two methods of scatter plot presentations on a computer screen are discussed in the next subsections under static and dynamic plots. One of the methods utilizes the static plot where no logical connection between points is displayed. The alternative dynamic method allows removal of some of the points from the display, a change to the scale of a plot or labelling points having something in common (for example, a point identified in one plot showing up in the same colour in all other displayed plots).

7.5.1 Static plots

For a problem involving n variables there will be $\frac{n(n-1)}{2}$ scatter plots, each showing one $x_i \sim x_j$ plane. These plots are ideally juxtaposed in a single screen frame. A maximum of 14 plots have been displayed at any one time on a standard 14" VGA display. For higher dimensional problems requiring more than 14 plots the user is either expected to choose a subset of the planes of interest or use a higher resolution display. A page display feature could easily be implemented if necessary.

In some cases there may be a practical, problem related, need for identifying a specific confirmed point, or a subset of confirmed points, in one of the scatter plots and tracing them in the other plots. One example arises when one is interested in tracking solution points which lie within a preferred area of the search domain. A second need, which might arise when the level set value-c has fallen to an acceptably low value, is to track points which were once confirmed in the acceptance set and were subsequently discarded in the search process when they did not fulfil the current level set condition.

Tracking a single point with some particular numerical property which can be detected within the computer program was implemented. An example of this was tracking of the current best point in the acceptance set. A special display symbol (+) was assigned to the current best point and this can be seen in Figure 8.3, Chapter 8, for a six variable problem where it is discussed in connection with sensitivity analysis. Experience with this feature and the many test problems showed that, at any stage of the search, the current best point rarely lies close to the centre of the cluster in any of the $x_i \sim x_j$ plots.

The greatest insights to the problem being solved can be provided by the scatter plots of confirmed points at any stage of the LSP search. For example, when dealing with problems involving more than two variables, the user might have preference for solution points within a certain range for one or two of the key variables. His interest will then be to identify the corresponding points in the other planes.

The implementation of point identification schemes becomes more complicated when there is an interest in identifying groups of points at the same time. Some multiple point detection schemes were implemented to investigate the kinds of information which could be inferred from visually identifying points which had been discarded at previous iterations. Points discarded in each of the last 5 iterations, identified by iteration, and those surviving in the current acceptance set were displayed on the same plot. Points discarded at each iteration were thus considered as members of a single group.

Several techniques were tried to visually distinguish between points in each group. The first attempt used different colours for different point groups, and the second used different symbols for points in different groups. Both attempts did not give meaningful results after the first few iterations. In the latter stages of LSP search, points from different groups overlap so that their differences become indistinct.

A slightly modified approach to the above methods where only the points discarded at one iteration are displayed at any time was also implemented. The set of plots for a single group were displayed in a single frame. Each frame remained on the screen for only a short time (about a second) and the display cycled through all the frames. This animated approach did not appear to have any particular value.

To overcome the difficulties mentioned above, and to exploit the benefits of scatter plots, it is better to use dynamic graphs instead of static graphs. In dynamic graphics, a viewer can see changes which have occurred over time. This improved approach to presenting scatter plots is explained in the next section.

7.5.2 Dynamic plots - Alternagraphics

Alternagraphics [Becker et al., 1987] is a technique which allows the viewer to interactively select some points from a set of scatter points and then view the selected point displayed in any number of corresponding two dimensional displays. Another feature is that the displays can be modified rapidly as the user explores the data set so that primitive animation effects are achieved. The purpose of alternagraphics is to convey information which is not easily perceived in static displays. It is, in effect, just another rather specialized form of computer graphical user interface. Because LSP generates a large set of solution points at every iteration as well as in the final solution, alternagraphics is far more exploitable with LSP than with any of the more conventional nonlinear optimization techniques. Alternagraphics can be implemented in three different ways provided that the points plotted on the screen may be categorized into various groups depending on their importance to the viewer or other numerically identifiable characteristics.

The first approach displays points belonging to successive acceptance sets for only a short time, automatically cycling through the most recent sequence of iterations. The second approach displays all data points on the screen simultaneously in a number of two dimensional plots, but highlights the points belonging to a particular set. Points in a series of sets are then highlighted one after the other. The third approach provides the user with the capability of turning any set ON or OFF rapidly. When a set is turned ON, either its points are highlighted and other points remain unchanged, or only those points turned ON are shown and all the other points are erased from the screen. The ON or OFF procedure can be initiated from a displayed menu, pressing a function key or clicking a button on the mouse to activate the changes.

The third approach mentioned above, where the user has the freedom to interactively choose point sets, appears to offer the greatest benefits in engineering practice. For example, in solving engineering problems with LSP, the engineer might be interested in design solutions which can be achieved within a specific value range of one of the design variables. Using the scatter plot of points in any of the two dimensional displays including the variable of interest, the points which lie in that specific range can be selected and turned ON. The display of the corresponding (i.e. same) points in all the other planes will then be highlighted automatically. This helps the engineer to visualize the distribution of the selected solution points in the other design variable domains.

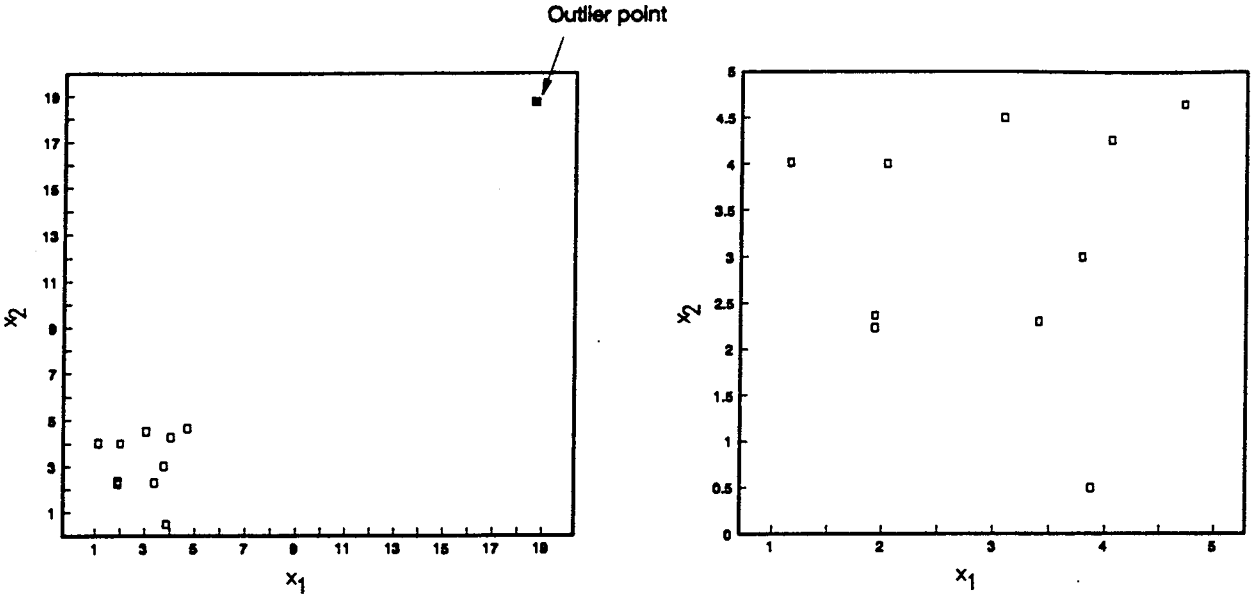
Some of the main features of dynamic graphics discussed extensively in [Becker et al, 1987] are reviewed here to further elaborate on their use and implementation with LSP.

Deletion:

When a single point distorts the scale of a scatter plot, the outlier can be deleted from the plot by simply clicking the mouse cursor at that particular point. Once the outlier is removed, the plot is instantaneously re-scaled and the points appear on the screen with the new axes ranges. Points which had been crammed in a small region are now well dispersed, improving the resolution of the plot. Figure 7.1 demonstrates the improvement of plot resolution by removing the outlier at the upper right corner of Figure 7.1(a).

Linking:

Suppose there are $\frac{n(n+1)}{2}$ scatter plots of points representing the N_{keep} points in an acceptance set of an n variable problem, each plot in an $x_i \sim x_j$ plane. Linking visually connects the corresponding points in the different scatter plots. Certain points are chosen in one of the scatter plots by clicking the mouse on each point and the linking procedure then highlights them on all the other plots. Consider, for example, in a 4 variable



(a) Points crammed in one area

(b) Improved resolution

Figure 7.1: Deletion of a point in dynamic graphics.

problem, the two scatter plots in the planes of $x_1 \sim x_2$ and $x_3 \sim x_4$. Suppose we want to identify the same points in these two scatter plots. Some points are first chosen in the $x_1 \sim x_2$ plane, the same points are then automatically identified in the $x_3 \sim x_4$ plane by highlighting or similar labelling technique. Therefore joining points means visually linking the point $(x_1, x_2)^k$ on the first plot to $(x_3, x_4)^k$ on the second plot, where k indicates the point number. Linking of points is illustrated in Figure 7.2, where the highlighted points in Figure 7.2(a) and (b) represent the same four points in two different planes. Intermediate results of an LSP run for a 4 variable problem, [Problem 4-1-0-3, Appendix A], were used to draw the points in Figure 7.2. In this instance the highlighted points correspond to the current 4 best points which had been selected automatically by the LSP software.

Linking is also a feature of static graphics display and was described in section 7.5.1.

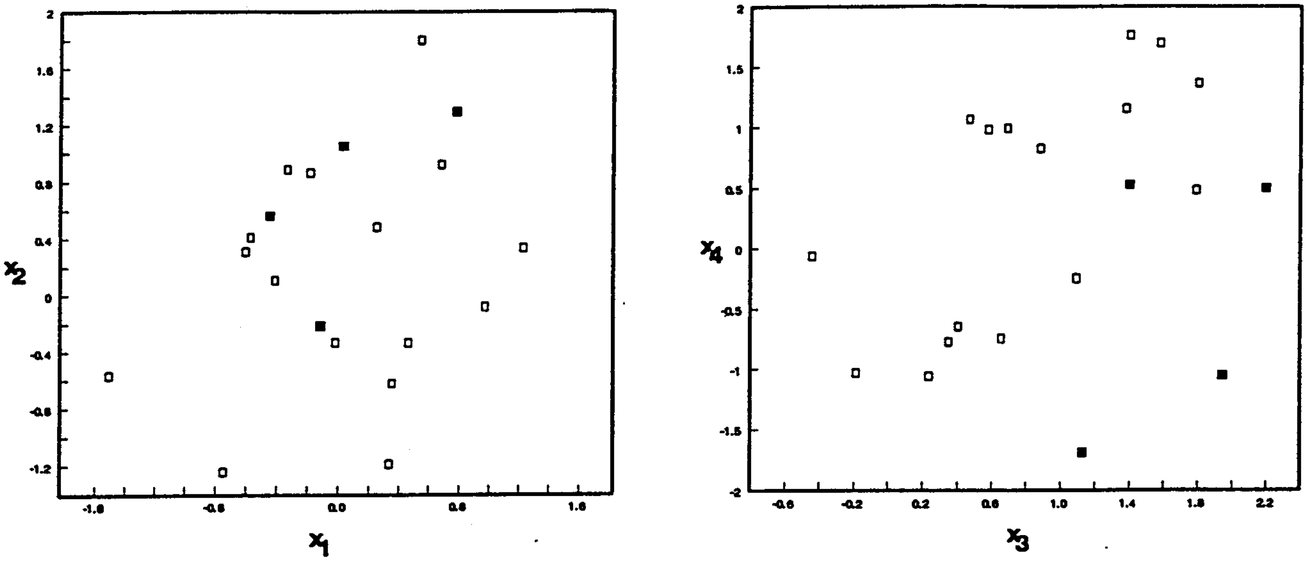


Figure 7.2: Linking of points for a 4 variable problem [Problem 4-1-0-3].

Brushing:

This is a dynamic method in which the user moves a small adjustable rectangle around the screen, with a mouse, in order to identify points within the rectangle. The rectangle is called a brush and each two dimensional plot is called a panel. This technique is used for high dimensional problems which need more than one panel to show points in all planes of the feasible region.

There are two ways of displaying points which lie inside the brush. As the brush moves around the active panel, points within the brush are highlighted in all of the other panels. The second method is performed with all data points plotted on the active panel, but only points in the brush are shown in the other panels. Figure 7.3 is adopted from Becker to demonstrate the second approach of brushing, where only points in the brush are displayed in all the panels other than the active panel [Becker et al, 1987]. In this figure, ozone, radiation, temperature and wind speed designate 4 different variables.

The brushing technique can also be used to delete points by pressing one of the

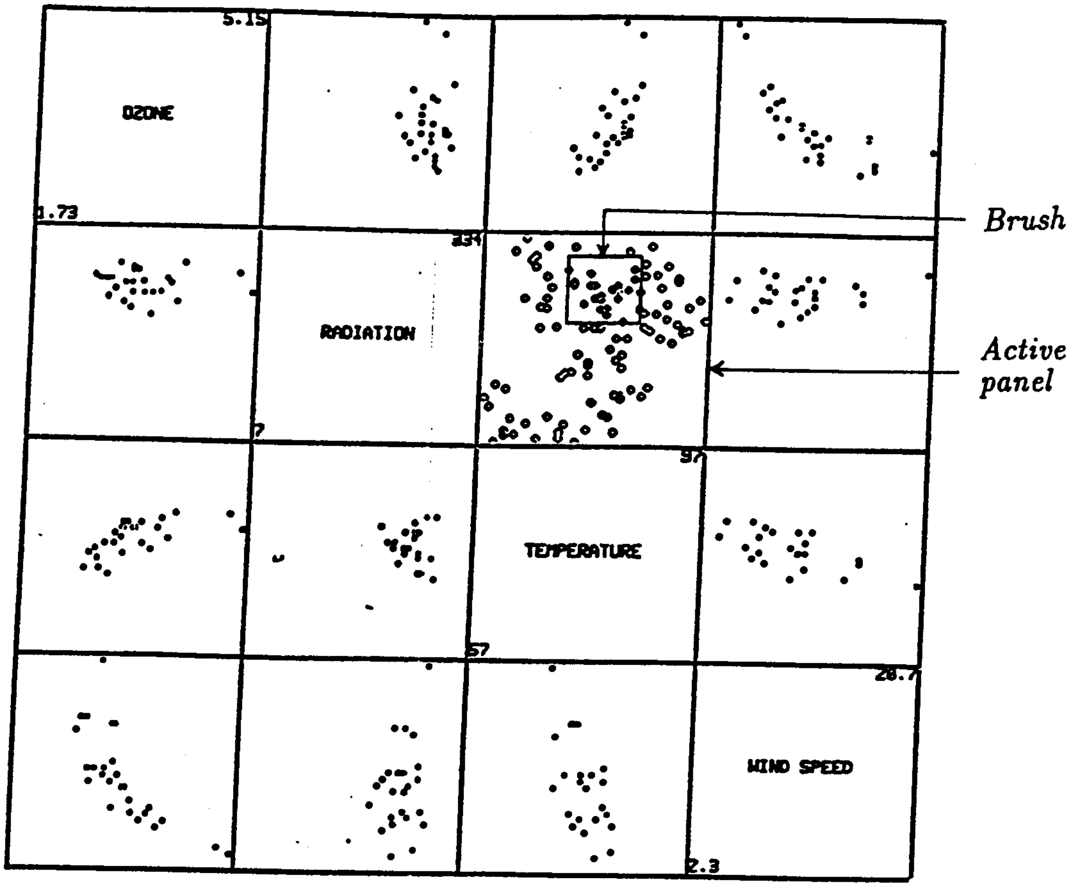


Figure 7.3: Points in the brush highlighted in all the panels [Becker, 1987].

mouse's buttons. Points inside the brush on the active panel are deleted, temporarily, along with corresponding points on the other panels.

The shape and size of the brush can be changed to meet the requirements of the user. When there is a need to examine the effect of only one variable on the others, the brush can be adjusted to produce a long and narrow rectangle. Such a slender brush can also be used to give information on the nonlinear dependence of one variable on the others.

Alternagraphics enhances understanding of the nature of the problem being solved. Implementation of this information in engineering has yet to be fully investigated but, in conjunction with LSP, it appears to have considerable potential for supporting practical engineering applications of nonlinear optimization.

Chapter 8

SENSITIVITY ANALYSIS

Sensitivity analysis is dependant upon numerical information which can be derived directly from the optimal solution and obtained with minimal computational effort. For the established optimization methods this means that the sensitivity output is primarily associated with changes in the optimal solution due to variation in the numerical value of coefficients in the problem formulation, both in the objective function and the right hand side constant coefficients. This information can be important when the coefficients in a problem formulation are poorly defined or can in fact be adjusted.

In LSP, useful sensitivity information which is obtained without any additional analysis is different from that discussed above. It utilizes the acceptance sets with level set values close to the global optimal value to indicate how near optimal solution points are distributed in the search domain. Here the emphasis is not on the coefficients but on the values of the decision variables and the value of the objective function. Therefore the sensitivity information indicates how much one can deviate from the optimal point(s) without the objective function excluding its global optimal value by some prescribed amount. Revealing the distribution of multiple global optima in the decision domain might also be considered as a part of this kind of sensitivity analysis.

LSP sensitivity information is undoubtedly of practical value as it focuses on the influence of those decision, design, or activity variables over which the engineer exercises actual control and can therefore adjust their value. Furthermore, in some important applications of nonlinear optimization, conventional sensitivity analysis results which focus on variation of coefficient values are of no practical value. For example, in model fitting the decision variables are the parameters of the model and the coefficients are the experimental data.

There are a number of options for a model fitting criterion. Consider the case of a simple linear model where the optimization problem is to minimize the sum of the squares of deviations between observed and model generated estimates of a dependent variable z. There are two independent variables x and y and m sets of observations of x, y and z. Each observation is identified by its subscript i. The problem is expressed as follows

$$Minimize \ \sum_{i=1}^{m}(\overline{z}_i - z_i)^2$$

where the model generated data set $\overline{z}_i$ is expressed as

$$\overline{z}_i = a * x_i + b * y_i$$

and the constraints are

$$a * x_1 + b * y_1 = z_1$$

$$a * x_2 + b * y_2 = z_2$$

$$a * x_3 + b * y_3 = z_3$$

$$\vdots \qquad\qquad \vdots$$

$$a * x_m + b * y_m = z_m.$$

The two model parameters, a and b, must be established on the basis of m observations of the independent and dependent variables.

In this problem x_i, y_i and z_i represent the already realized experimental data but also serve as coefficients in the constraint set, whereas a and b are the decision variables, i.e. the unknown model parameters whose values are being established. Because the coefficients in the optimization formulation are observed data sets a sensitivity analysis which is concerned with changing the coefficients has little meaning. In contrast, an indication of the sensitivity of the objective function value to the value of a and b in the region of their optimal values could provide some useful insight as to the robustness of the estimates of a and b. Sensitivity information of this type is readily provided by LSP.

The remainder of this chapter attempts to describe the generally interpretive nature of LSP sensitivity analysis which is derived almost entirely from the graphical outputs. The interpretation of the graphical output is influenced by the finite nature of the point sampling and by topological characteristics of the various functions involved in the problem being solved. The sensitivity conclusions discussed below may therefore not necessarily be the most appropriate for all cases.

8.1 Sensitivity with LSP

Once the LSP convergence criterion is met then all points in the final acceptance set are global optima or provide near optimal solutions. In cases where these points form a distinct single cluster and the difference between any two points is approaching the numerical precision of the computer, then the final acceptance set can be considered to represent a single global optimum point. The size of the final acceptance set is, however, normally determined by the convergence criterion, V_M so that in many cases, the confirmed points in the final cuboid may be scattered over a larger region. Then the difference between any two confirmed points is not simply governed by numerical precision but reflects a considerable difference between solutions yielding optimal or very near optimal solutions. Large distances between points suggests that the problem has either multiple optima or a very flat objective function surface in the region of the acceptance set.

In spite of the interpretations suggested above, and in the remainder of this chapter, one should exercise caution when associating the cuboid size and the distribution of points in the acceptance set with the number of distinct global optima. Even though the final cuboid is assumed to contain all globally optimal points, all confirmed points in the final acceptance set may not produce the global optimum value and, with the discrete sampling involved, some globally optima points may not have been sampled.

The $x_i \sim x_j$ scatter plots can be viewed with axes lengths determined by either the variable bounds established prior to running LSP (the initial cuboid) or within the cuboids at the later or final stages of the search.

8.1.1 Sensitivity interpretation of the confirmed points at convergence plotted in the initial cuboid

The plots of the confirmed points at convergence in the initial cuboid provide another sensitivity perspective on the LSP solution. Usually the scale of these plots, and the display resolution, will obscure the details. They do, however, reveal certain characteristics of the optimum solution from the perspective of the engineer and his problem. Because of resolution limitations at this scale, points which are close together will coalesce to a single point in the display so that the visible points can be assumed to be global optimal points. Problems which have multiple optima often produce distinct scatter plots which can be meaningfully interpreted in various ways. Figures 8.1(a) to (e) are used to demonstrate the interpretation of some typical final scatter plots, within the initial cuboid, for some two variable problems.

Figure 8.1(a): Suggests that there is only a single optimum in the region of interest and the optimum point lies around the centre of the search domain.

Figure 8.1(b): Suggests that there is a single optimum point but two of the variable bounds are probably acting as active constraints. From a practical point of view this

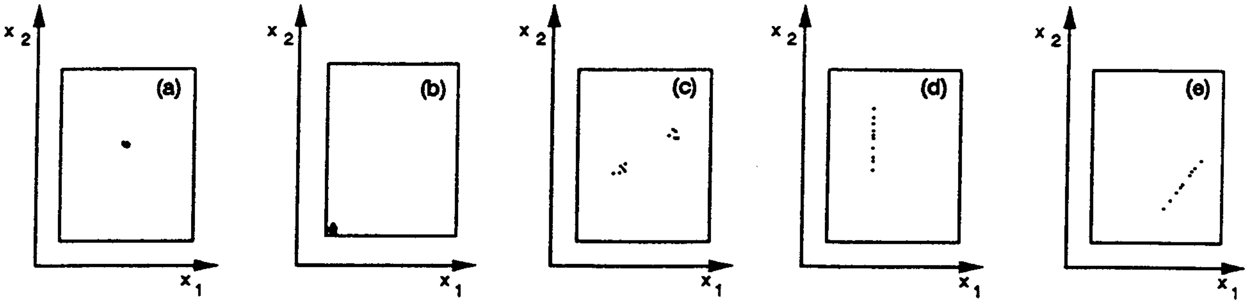


Figure 8.1: Plots of confirmed points after convergence criterion is met.

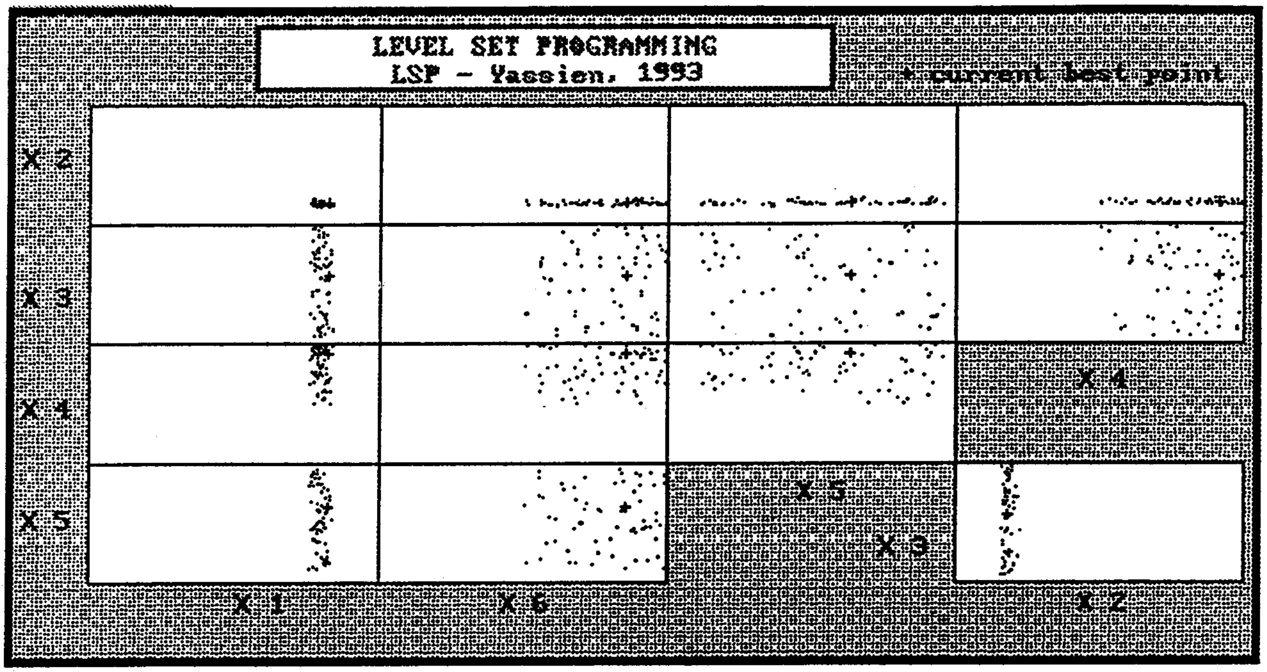
suggests that, if it is possible, a relaxation of bounds might produce a better result.

Figure 8.1(c): The plot clearly suggests the existence of two distinct global optimal points.

Figure 8.1(d): The plot suggests the possible existence of multiple optima, but one of the variables (x_1 in this case) has the same, i.e. constant, value at all of the optimal points.

Figure 8.1(e): Indicates that in a specific region of the search domain, the confirmed points can be approximated by a diagonal line, which implies a linear relationship between the two variables in the region of near optimality.

An actual screen dump for a six variable problem [Problem 6-2-0-6, Appendix A] is given in Figure 8.2 to demonstrate the kind of variability that can occur in the appearance of the 14 individual $x_i \sim x_j$ plots. The plot was generated with a relatively high value of the convergence criterion to make the plots more distinctive. It shows, for example, that fixing the values for variables x_1 and x_2 might be justifiable to simplify the optimization problem as this may not change the optimal solution significantly.



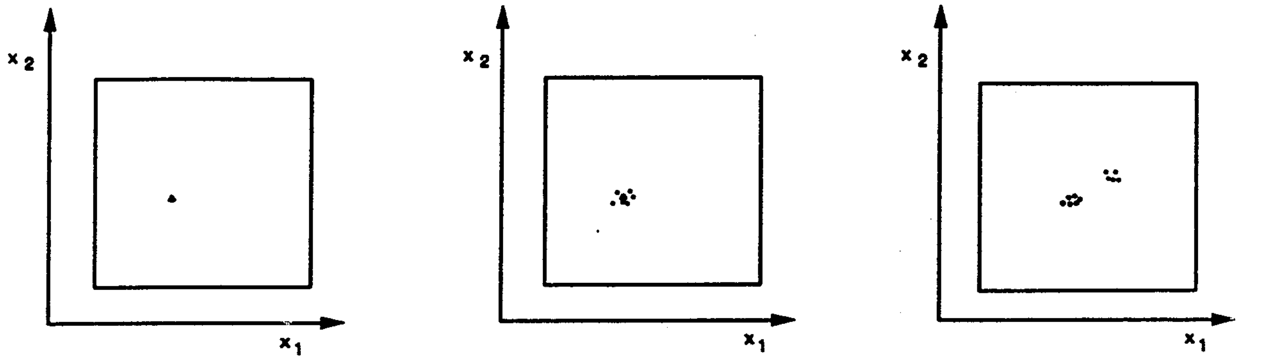
LEVEL SET PROGRAMMING
LSP - Yassien, 1993

Figure 8.2: Plots of confirmed points at convergence within the initial cuboid for a 6 variable problem (Actual screen dump)

8.1.2 Response of acceptance set to changes in the level set value-c

Once LSP is run and an optimal solution found, there may be a value in investigating the sensitivity of the decision variables with respect to the value of the objective function. More specifically determining to what extent an objective function value which is inferior to the global optimum value permits a wider range of decision variable values. This is done by storing the acceptance sets at all iterations. Figure 8.3 shows the kind of effect which might be observed when c is raised above c^* when the acceptance sets are plotted in the initial cuboid.

Actual screen dumps of plots for the test problem [Problem 4-6-1-2, Appendix A] are



(a) c = c*

(b) c > c* (connected level set)

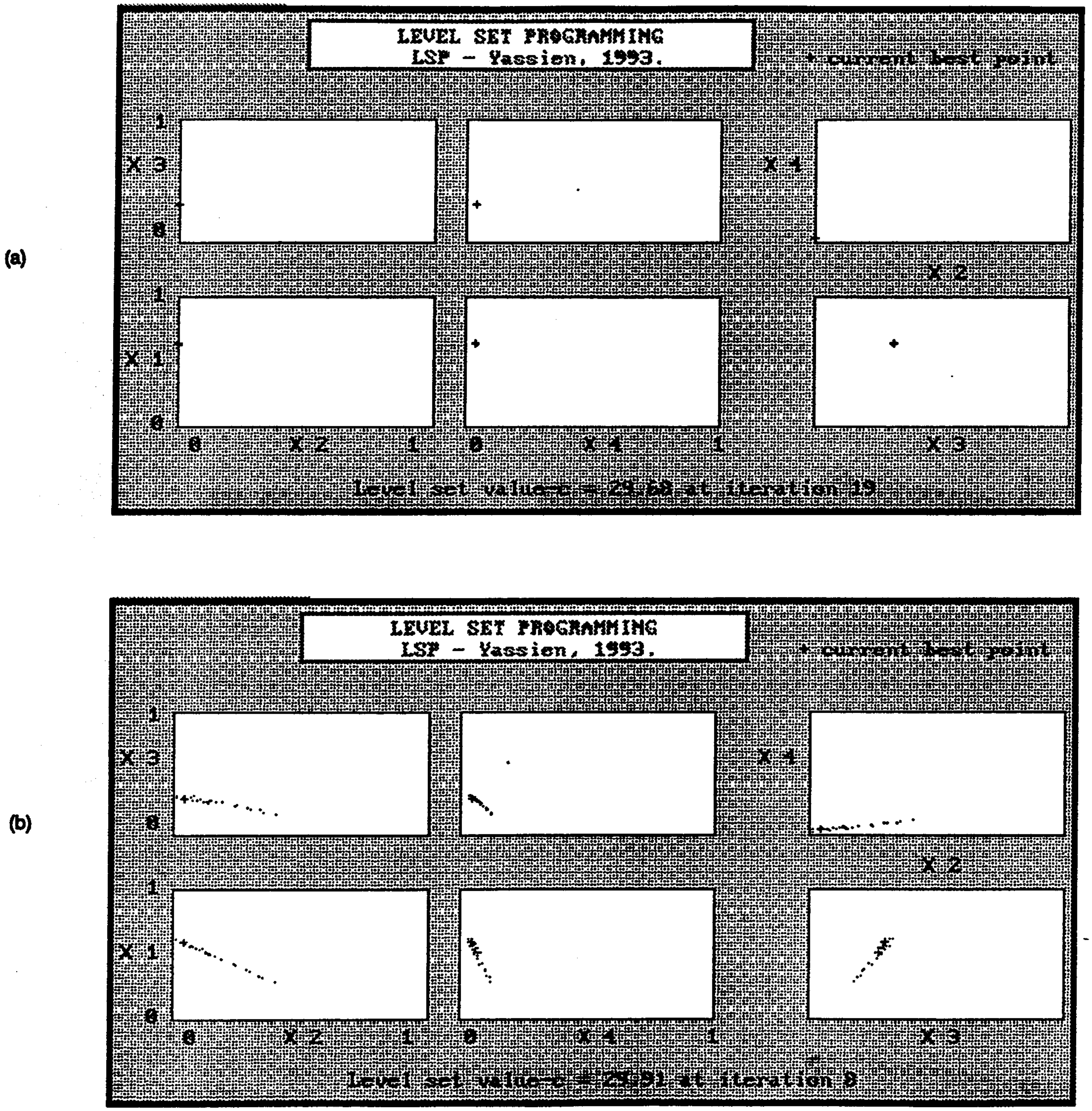
(c) c > c* (partitioned level set)

Figure 8.3: Plots of confirmed points within the initial cuboid for different c values

given in Figure 8.4 and represent outputs at two level set values. Figure 8.4(b) shows points in the acceptance set for a level set value of 29.68, which is the global minimum. Figure 8.4(a) shows points in the acceptance set for the same problem but with the level set value-c raised by 1% to 29.96 which coincides with c at iteration 8. As a result many new points are introduced into the acceptance set. Most of the new points are located far from the global optimum point, as shown in Figure 8.4(a). The small increase in the objective function value has permitted a wide range of possible solution points some of which may offer benefits not measured by the objective function and therefore preferred as practical solutions to the problem.

8.2 Other approaches to obtaining sensitivity information

Information similar to that obtained for coefficient values in conventional sensitivity analysis can also be obtained with LSP, though at some additional computational expense. When the sensitivity can be expressed in the form of a gradient at the optimal solution (e.g. Lagrange multipliers) then only a slight perturbation of a coefficient value is necessary to estimate the gradient by finite difference. A small change would almost guarantee that the acceptance set already obtained just a few iterations before convergence would provide an appropriate starting point for solving the revised problem. Thus only a small additional computational effort would be involved for solving for each perturbed coefficient.



(a)

LEVEL SET PROGRAMMING
LSP - Yassien, 1993.

Level set value-c = 29.68 at iteration 19

(b)

LEVEL SET PROGRAMMING
LSP - Yassien, 1993.

Level set value-c = 29.91 at iteration 9

Figure 8.4: Points in the level set, for $c = c^*$ and $c = 1.01 * c^*$ - [Problem 4-6-1-2] (Actual screen dump)

Chapter 9

CONCLUSION

9.1 Introduction

A global search scheme for nonlinear optimization problems based exclusively on level sets, was first presented in *"Integral global optimization"* by [Chew & Zheng, 1988]. It was this work that provided the starting point for the research in this thesis. The following summary of its shortcomings is not intended to question the value of Chew & Zheng's contribution in any way but to provide a clearer perspective on the contribution of this thesis. As was suggested by the title, they placed considerable emphasis on the theoretical properties of integral expressions for the higher moments of the objective function values associated with solution points in level sets. Much of this theory was found to have little or no bearing on the implementation of a level set based optimization scheme. Although the capability of identifying multiple optima was mentioned, the authors focused on a method for solving problems with a single global optimum only. Thus the potential of the methodology to solve multiple global optima problems as well as identify near optimal solutions was overlooked. A systematic performance assessment over a variety of challenging test problems was not included. The importance of the interpretation of intermediate and final output from a level set search and the potential for a graphical interface were not recognized.

Level sets have been used to augment some global optimization schemes. But the view of some of the authorities in that field has been that the use of level sets as the principal search tool is impractical. The level set method was labelled as being only a "theoretical" iterative scheme in [Horst & Hoang, 1990], and as being "not designed for solving practical problems" in [Torn & Zilinskas, 1988]. A level set based search falls into the class of iterative direct search methods of optimization. Some authors clearly feel that direct search methods are applicable for only low dimensional problems. Edgar & Himmelblau [Edgar & Himmelblau, 1988] suggest that "direct methods ... are not as efficient and robust as many of the modern indirect methods, but for simple two variable problems, they are satisfactory". Also surprising in the light of experience in this thesis is the view that optimization methods based on random sampling are suitable only for low dimensional problems or as devices for generating starting points for more sophisticated methods [Reklaitis et al., 1983].

Computational efficiency has dominated the development of nonlinear optimization methods over the past 30 years. The negative views which have been expressed about direct search methods in general, and level set methods specifically, arise from computational efficiency concerns. As was discussed in Chapter 6 on performance assessment, ease of use is more likely to be the dominant criterion in the future and computational efficiency only of secondary importance.

The research reported in this thesis set out to investigate the use of level sets in practical nonlinear optimization. As the investigation proceeded many new, previously unexploited aspects of level set optimization were revealed and incorporated into the evolving implementation scheme. This implementation was eventually given the name Level Set Programming (LSP) as it appears to have the necessary characteristics, including a theoretically sound global convergence criterion, to qualify as a mathematical programming tool. It uses estimates of only the first two moments of the objective function values at the (discrete) points in the level sets to redefine the search domain at each iteration and to measure convergence on the optimum solution. LSP utilizes approximations of the true level set (or acceptance set in constrained problems) boundaries in the form of cuboids for efficiently generating points in the level sets. The combined effect of the progressively improving level set and the contraction of the search domain makes LSP an efficient and reliable global search procedure.

LSP can handle, without modification, both equality and inequality constraints, and also has the provisions to incorporate existing techniques of optimization, such as the penalty functions. It also makes use of established clustering analysis techniques to improve the search efficiency for problems involving widely separated multiple global optima.

Unlike many other optimization methods, the intermediate results generated during an LSP search provide a global view of the problem being solved. This is in contrast with the single path view provided by say gradient methods. Because a much larger number of point solutions are established at each iteration than is the case with other practical direct search methods, LSP's view of the problem is far more complete than with the established direct search methods.

9.2 Evaluating LSP performance using test problems

A large set of published mathematical and engineering test problems, already solved by a variety of NLP methods, were solved with LSP to evaluate its performance. The test set included unconstrained and constrained problems, continuous and discontinuous functions, and continuous and discrete variables. The dimensions of the problems varied from 1 to 38 variables. The results showed that solutions found with LSP were generally in agreement with the published results, with some improvement in about 5% of the cases. The only aspect of performance to other methods cited in the literature, which shows some superiority over LSP, is in the shorter computational time expended to solve test problems. Even though similar machines are not used for all the cases, the computational time taken to meet the termination criterion reported in the literature was generally lower than with LSP. Still, the time spent with LSP was found to be well within a tolerable time frame from a practical engineering design optimization point of view. The additional computational time can be justified easily when it is weighed against the reliability of LSP as a global optimizer and the extra information it provides during the search.

The nonlinear optimization test problems in the literature have many shortcomings and there are virtually no test problems capable of testing global optimization schemes in any systematic way. To more adequately test the strengths and limitations of LSP, a new mathematical test problem named *'The Road Runner Function'* was developed. It has a single global optimum and several local optima for any dimension n. The function is parametrically adjustable and can be easily extended to any number of dimensions while retaining its key features. It is a challenge to all existing NLP and direct search methods unless an ideal starting point is chosen.

One problem discovered in the literature, that by Subrahmanyam [Subrahmanyam, 1989], was found to have many desirable attributes for a test problem. The formulation and the geometry of its feasible region and objective function surface are simple. It presents convergence difficulties for direct search methods even though the dimensionality of the problem is low. LSP was only able to solve this problem by significant readjustments to its search parameters.

9.3 LSP performance improvements

As a result of the experience gained with the test problems both the reliability and efficiency of LSP were substantially improved over the course of this research. This entailed the refinement of some of the techniques suggested in [Chew & Zheng, 1988], development of some new techniques, the introduction of parameters to control these techniques, and the development of diagnostics to assess progress and guide the modification of these parameters.

One heuristic method investigated involved subdividing the cuboid into a set of smaller subcuboids and then performing searches within each subcuboid. This technique is applied when the intermediate results displayed in the graphical output or cluster analysis indicate the possible existence of a partitioned level set. The division of the search domain avoids inefficient searches in-between the connected regions of the level set, so that the search then concentrates on smaller subregions each of which is fully connected. The sum of the function evaluations to meet the termination criterion for each subregion was found to be considerably lower than the number of function evaluations for the single region with a partitioned level set.

When the feasible region formed by the set of constraints occupies only a very small region of the initial cuboid, feasible point generation can be difficult and result in an inefficient or suspended LSP search. An indication that this difficulty has arisen is provided by the number of function evaluations required at the first LSP iteration. This early feedback permits corrective action to be taken before a large amount of computational effort is wasted. Constraint relaxation was found to improve the efficiency of sample point generation by temporarily increasing the volume of the feasible region. To discourage the existence of infeasible points in the solution set, penalty terms were also added to the objective function. The progressive tightening of the relaxed portion of the constraints in conjunction with the penalty terms on the objective function assures the best continuity between successive acceptance sets and that solutions are found within the feasible region of the original problem.

Due to the random generation of sample points, a global optimum point may be missed if it lies at a boundary of the feasible region. To increase the capability of generating boundary points the technique of skewness adjustment was implemented. This technique modifies the location and boundaries of the current cuboid to ensure that no potentially important regions are excluded from the current cuboid. In addition to this skewness adjustment, reduced bias estimates of cuboid bounds are calculated at every iteration. Both of these adjustments further increase the volume of the cuboid and maximize the probability that the true acceptance set is included within the modified cuboid.

Two other alternatives to the cuboid approach, replacing the cuboid with a rhombohedron and exploiting a linear relationships between a variable pair, were also investigated. These were found to provide significant efficiency improvements under specific but commonly occurring conditions which are detectable from LSP's output.

As with all NLP schemes, equality constraints present significant difficulties. In an ideal case, equality constraints can be used to reduce the number of variables involved, otherwise they must be accommodated using a penalty technique. The approach used with LSP is similar to the classical penalty approach, but the penalty parameter value is reduced at every iteration. This progressive modification of penalty terms throughout the search process was found to significantly improve the chances of reaching the global optimum in the presence of equality constraints.

9.4 Graphical output

The intermediate graphical displays provided by LSP and their interpretation open up a new dimension in engineering optimization. These plots provide entirely new ways to judge the progress of the search as well as insights into the topology of the particular problem being solved.

The first set of plots, the $x_i \sim x_j$ scatter plots, could of course, be obtained with any NLP method, but are far more meaningful with a technique such as LSP which provides a large number of confirmed solution points at each stage of the search. These plots facilitate visualizing the acceptance set boundary shapes and will often reveal the presence of distinctly separate global optima or near optima. The plots can also give indications of dependencies between the variables which can be used to speed up the search. The second type of plots are the progressive plots of level set value-c versus iterations-I which can be used as an indication of the strength of convergence. Most importantly the $c \sim I$ curve can also suggest the possible existence of fissure like features on the objective function surface. A third type of plot, the plot of number of function evaluations-N_f versus iteration-I, reveals the efficiency of the search as it progresses. Its interpretation can give an indication of the existence of multiple global optima, or the existence of multiple local optima with only one global optimum.

One of the most interesting aspect of the plots discussed is that much of the triggering information for LSP parameter adjustment or problem modification concerns just c, N_f and I. It is also significant from a users point of view that the magnitude of c is related directly to the real world problem being solved while N_f and I are easily understood quantities related to the search. The influence of these quantities on the search strategy is therefore easy for practitioners to understand.

9.5 Civil engineering design problems and LSP

The mathematical equations used in civil engineering design are predominantly nonlinear. In many instances the mathematical formulation of the associated design problems do not meet the ideals, from a nonlinear optimization point of view, of providing a convex objective function and a convex feasible region. It is also not uncommon for the decision variables to refer to discrete choices of pipe diameter, structural member size, etc. so that discrete decision variables must be accommodated. Most civil engineering design optimization problems therefore require the use of nonlinear programming techniques which can accommodate mixed variables for their solution. Furthermore, in engineering design, there are instances where the design which produces the optimum objective function value may not be the best practical option. Alternative solutions which produce slightly inferior values to the mathematical global optimum solution may be preferred because of factors which could not be captured in the formulation. Therefore, a feature of an optimization method which might appeal to civil engineers is one which can identify near optimum solutions. Simplicity of use of an optimization method also translates to economy of use in a commercial engineering office and this alone may be the major factor determining the practical use of optimization.

Currently, there are several NLP packages being offered commercially but most well developed packages use gradient methods and require the differentiability of the objective and constraint functions. These do not fully satisfy the needs of civil engineers for several reasons. Gradient methods basically find local optima and can not guarantee finding a global optimum except under very ideal conditions. Unfortunately, they do not give any indication of the local or the global nature of the solution found. When problems have multiple optima, either local or global, a single gradient search can only identify a single solution. The solution found depends entirely on the starting point of the search. No systematic way of selecting starting points to ensure finding the global optimum is available and the responsibility for the overall global search strategy and its success is left entirely to the user.

Most NLP packages which have the ability to handle practical sizes of problems have, until very recently, been available only on mainframe computers. There are only a few nonlinear optimization packages which can address practical engineering problems using personal computers and these use almost without exception, the same gradient methods which have dominated mainframe optimization for the past 30 years.

The feedback provided by the gradient search methods are in the form of matrices of first and second partial derivatives, matrix condition, Lagrange multipliers, etc.. This kind of information rarely has any direct relevance to an engineer's view of his engineering design problem. Additionally, this information can be particularly difficult to interpret when search failures occur or when the global nature of the solution is in question. If a gradient NLP method fails to confirm even a local optimum, the user is left with no suggestion for his next move.

In summary, in spite of the recognized need for optimization in civil engineering, existing NLP packages do not appear to satisfy the needs of practitioners. It is entirely possible that it is the available methods, and not just their computer implementation, that have insufficient appeal to convince engineers that their time should be invested in that direction.

LSP is a global search method and the chance of converging on the global solution is much higher than with any of the gradient based methods such as GRG and MINOS, even when a number of different starting points are used. One of the important aspects of LSP is its capability to identify multiple optima and near optimal solutions. If multiple local optima exist, LSP either identifies all of them or gives indications of their existence in the process of identifying the global optimum.

LSP overcomes many of the limitations of the existing gradient methods but does not radically reduce the computational effort involved for those problems which could be solved using a gradient method. Perhaps the most significant features of LSP from a practical engineering standpoint are that: the method is conceptually simple, and can therefore be easily understood by engineers who are generally not experts in numerical analysis techniques; virtually all of the computations and numerical results generated during the search are meaningful in the context of the engineering problem; the program can be run on personal computers; and an elementary, and therefore fast, graphical interface can display all of the useful information at any stage of the search. Flexibility, simplicity and meaningful intermediate results during the search lower the engineers' time requirements for familiarization and therefore make LSP economical to implement even when its use is only occasional.

The success of global optimization is highly dependent on the complexity of the objective function surface and the boundaries of the feasible region, in other words their general topology. A better appreciation of the topological nature of the global optimization problem can only help to greater understanding and possibly refinement of global optimization methods. Yet topological interpretations have not been widely used in optimization, the theoretical basis for level set optimization presented in [Chew & Zheng, 1988] being a rare example. Topology is an established and mature subtopic of mathematics with a considerable body of theory, research in this topic may therefore provide new perspectives on level sets and provide the basis for further refinements in LSP type methods.

Subdividing the search domain into smaller regions was found to increase the search efficiency considerably for multiple optima problems. It was also observed that determining the number of clusters and allocating points into different clusters becomes difficult with the number of variables in a problem. Consequently, high dimensionality cluster analysis methods should be further explored with LSP's particular requirements of low precision cluster analysis, as discussed in Chapter 3, section 3.1.3, in mind.

Graphical interpretation of both intermediate and final optimization results opens up a new perspective in the field of optimization. The fact that LSP produces and stores a large set of intermediate results in the form of solution points makes the graphical interface far more attractive than with almost all other optimization schemes. This is especially true with high dimensional problems when the information provided in the dynamic graphs is high. Therefore, alternagraphics and similar display techniques should be further researched to fully exploit the inherent ability of LSP to support graphical outputs.

Bibliography

The following bibliography is arranged in conventional alphabetical order. The numerical references provided are used only in the Appendix.

[1] Alperovits, A. and Shamir, U., "Design of optimal water distribution systems", Water Resources Research, pp 885-900, 1977.

[2] Aluffi-Pentini, F., Parisi, V. and Zirilli, F., "Global optimization and stochastic differential equations.", JOTA Vol 47 NO. 1 pp 1-16, Sept 1985.

[3] Armstrong, M. A.,"Basic topology", New York, Springer-Verlag, 1983.

[4] Archetti, F. and Schoen, F.,"A survey on the global optimization problems: General theory and computational approaches", Annals of operations research No 1, pp 87-110, 1984.

[5] Ballard, D. H, Jelinek, C. O., and Schinzinger, R., "An algorithm for the solution of constrained generalized polynomial programming problems", Computer Journal, Vol 17, pp 261-266, 1974.

[6] Bazaraa, S. Mokhater, "Nonlinear programming, theory and algorithms". John Wiley & sons, 1979.

[7] Becker, A. Richard, Cleveland, S. William and Wilks, R. Allan, "Dynamic graphics for data analysis", Statistical Science, Vol 2, No 4, Nov 1987.

[8] Ben Saad, Sihem and Stefer Jacobsen, E. "A level set algorithm for a class of reverse convex programs", Annals of operations research, No 25, pp 19-42, 1990.

[9] Betrio, B. and L. De Biase, "A recursive spline technique for uniform approximation of sample data", Technical Report, University of Piza (1), 1987.

[10] Betro, B. and Schoen, F., "Sequential stopping rules for the multistart algorithm in global optimization", Mathematical programming 38, pp 271-280, 1987.

[11] Boender, C. G. E. and Kan, A. H. G. Rinnooy, "Bayesian stopping rules for multistart global optimization methods", Mathematical programming 37, pp 59-80, 1987.

[12] Box, M. J., "A new method of constrained optimization and a comparison with other methods", Computer Journal, Vol 8 pp 42-52, 1965.

[13] Chew, Soo Hong and Zheng, Quan, "Integral Global Optimization: Theory, implementation and applications", Lecture notes in economics and mathematical systems, Vol 298, Springer-Verlag, Berlin Heidelberg 1988.

[14] Cole, F., Gochet, W. and Smeers, Y., "A comparison between a primal and dual cutting plane algorithm for posynomical geometric programming problems", JOTA Vol 47 NO. 2 Oct pp 159-180, 1985.

[15] Corana, A., Marchesi, M., Martini, C., and Ridella, S., "Minimizing multimodal functions of continuous variables with the simulated annealing algorithm", ACM Transactions on Mathematical Software, Vol. 13, No. 3, pp 262-280, 1987.

[16] Crowder, H. P., Dembo, R. S. and Mulvey, J. M., "Reporting computational experiments in mathematical programming", Math Programming, No 15, pp 316-329, 1978.

[17] Davis, John C., "Statistics and data analysis in geology", John Wiley & Sons 1973.

[18] Devore, Jay L., "Probability and statics for engineering and the sciences", Brooks/Cole publishing company, 1982.

[19] Dixon, L. C. W. and Szego, G. P., "Towards global optimisation 2", Elsevier North Holland, 1976.

[20] Eason, E. D. and Fenton, R. G., "A comparison of numerical optimization methods for engineering design", Journal of Engineering for Industry, Trans of ASME, pp 196-200, February 1974.

[21] Edgar, T. F. and Himmelblau, D. M., "Optimization of chemical processes", McGraw Hill Inc. 1988.

[22] Everitt, Brian, "Math cluster analysis", Heinemann Educational Books Ltd, 1980.

[23] Falk, James E. and Hoffman, Karla L., "Concave minimization via collapsing polytopes", Operations Research Vol 34 No. 6, Nov-Dec, pp 919-929, 1986.

[24] Fleming, John F., "Structural engineering analysis on personal computers", McGraw Hill Book Company, 1986.

[25] Floudas, C. A. and Ciric, A. R., "Strategies for overcoming uncertainties in heat exchanger network synthesis", Computers and Chemical Engineering, 13(10), pp 1133-1152, 1989.

[26] Floudas, C. A. and Pardalos, P. M., "A collection of test problems for constrained global optimization algorithms", Lecture Notes in Computer Science, NO 455, Springer-Verlag, 1990.

[27] Fujiwara, O., Jenchaimahakoon, B., and Edirisinghe, N. C., "A modified Linear programming gradient method for optimal design of looped water distribution networks", Water Resources Research, Vol 23, No 6, pp 977-982, June 1987.

[28] Ghani, S. N., "An improved 'complex' method of function minimization", Computer aided design, January 1972.

[29] Gill, Philip E., Murray, Walter, Saunders, Michael A. and Wright, Margaret H., "Constrained nonlinear programming" in *Handbooks in operations research and management science, Volume 1, Optimization.* North Holland, 1989.

[30] Goldberg, David E., "Genetic algorithms in search, optimization, and machine learning", Addison-Wesley Publishing Company, Inc., 1989.

[31] Goldberg, David E. and Samtani, Manohar P., "Engineering optimization via genetic algorithm" in Electronic computation, pp 471-482, Kenneth M. Will ed. 1986.

[32] Gottfried, Byron S. and Weisman, Joel, "Introduction to optimization theory", Prentice Hall, 1973.

[33] Hartigan, John A., "Clustering Algorithms", Wiley, 1975.

[34] Himmelblau, David M., "Applied nonlinear programming", McGraw Hill Inc. 1972.

[35] Horst, Reinner and Hoang, Tuy, "Global optimization: deterministic approaches", Springer-Verlag, 1990.

[36] Huang, H. Y. and Aggarwal, A. K., "A class of quadratically convergent algorithms for constrained function minimization", JOTA Vol 16 Nos 5/6, pp 447-485, 1975.

[37] Jenson, David L., "The role of cluster analysis in computer assisted mass appraisal", Lincoln Institute Monograph NO 77, July 1977.

[38] Kan, A. H. G. Rinnooy and Timmer, G. T., "Global optimization", in *Handbooks in operations research and management science, Volume 1, Optimization*, North Holland, 1989.

[39] Kassicich, Suleiman K., "International intra-company transfer pricing", Operations Research Vol 29, No 4, pp 817-828, 1981.

[40] Lasdon, Leon S. and Waren, Allan D., "GRG2 user's guide", May 1982.

[41] Leon, A., "A classified bibliography on optimization", in recent advances in optimization techniques, pp 599-649, 1966.

[42] Lipschutz, Seymour, "General topology", Schaum's outline series in Mathematics, McGraw-Hill book company, 1965.

[43] Loganathan, G. V. and Greene, J. J., "Global approaches for the nonconvex optimization of pipe networks", Water Management in the 90's, 1993.

[44] Lucidi, S. and Piccioni, M., "Random tunnelling by means of acceptance-rejection sampling for global optimization", JOTA Vol 62 NO. 2, pp 255-278, 1989.

[45] Luenberger, David G., "Linear and nonlinear programming", Addison Wesley, 1984.

[46] Luus, Rein and Jaakola, T. H. I., "Optimization by direct search and systematic reduction of the size of search region", AIChE Vol 19, NO. 4, pp 760-766, 1973.

[47] Mays, Larry W. and Taur, Cheng-Kang, "Unit hygrographs via nonlinear programming", Water Resources Research, Vol 18, No 4, pp 744-752, 1982.

[48] Mezzich, Juan. E. and Solomon, Herbert, "Taxonomy and behavioral Science, comparative performance of grouping methods", Academic press, 1980.

[49] Morari, Manfred and Grossmann, Ignacio E., "Chemical engineering optimization models with GAMS", CACHE Process design case studies, Vol 6, 1991.

[50] Narula, Subhash C., "Optimization techniques in linear regression: A review", in Optimization in statistics, TIMS Studies in the management sciences, volume 19, pp 11-29, North-Holland publishing company, 1982.

[51] Orth, Hermann, "Model based design of water distribution and sewage systems", John Wiley & Sons, 1986.

[52] Pike, Ralph W., "Optimization for engineering systems", Van Nostrand Reinhold, 1986.

[53] Ratschek, H. and Rokne, J., "New computer methods for global optimization", Ellis Horwood Limited, 1988.

[54] Reklaitis, G. V., Ravindran, A. and Ragsdell, R. M., "Engineering optimization, methods and application", New York Wiley, 1983.

[55] Sandgren, E and Ragsdell, K. M., "The utility of nonlinear programming algorithms: A comparative study parts 1 and 2", ASME J. Mech Des 102(3), pp 540-551, July 1980.

[56] Sarma, M. S., "On the convergence of Buta and Dora random optimization methods", JOTA vol 66, No 2, Aug 1990.

[57] Schittkowski, Klaus, "Nonlinear programming codes: Information, Tests, Performance," Lecture Notes in Economics and Mathematical Systems, Vol 183, Springer-Verlag, New York, 1980.

[58] Schittkowski, Klaus, "Test examples for nonlinear Programming Codes", Lecture Notes in Economics and Mathematical Systems, Vol 187, Springer-Verlag New York, 1981.

[59] Schittkowski, Klaus, "More test examples for nonlinear programming codes", Lecture Notes in Economics and Mathematical Systems, Vol 282, Springer-Verlag New York, 1987.

[60] Shields, F. Douglas Jr., and Thackston, Edward L., "Designing treatment basin dimensions to reduce cost", ASCE, Journal of Environmental Engineering, Vol 117, No 3, May/June 1991.

[61] Subrahmanyam, M. B., "An extension of the simplex method to constrained optimization", JOTA, vol 62, NO 2, pp 311-319, August 1989.

[62] Templeman, Andrew B., "Discrete optimum structural design", Computer and Structures, Vol 30, No 3, pp 511-518, 1988.

[63] Torn, Amino and Zilinskas, Antanas, "Global optimization", Lecture Notes in Computer Science, NO. 350, Springer-Verlag, 1988.

[64] Turner, D. B., "Workbook of atmospheric dispersion estimates", Environmental protection agency, office of Air programs, Research Triangle Park, N.C., 1873.

[65] Venkayya, V. B., "Design of optimum structures" in Computers & Structures, Vol 1, pp 265-309, 1971.

[66] Visweswaran, V., and Floudas, C. A., "A global optimization algorithm (GOP) for certain classes of nonconvex NLPs-II. Application of theory and test problems", Computers and Chemical Engineering, Vol 14, No 12, pp 1419-1434, 1990.

[67] Wang, Bi-Chong and Luus, Rein, "Reliability of optimization procedures for obtaining global optimum", AIChE Vol 24, No 4, pp 619-26, 1978.

[68] Wasserman, William and Kutner, Michael H., "Applied linear regression models", Richard D. Irwin, Inc., 1983.

[69] Zanakis, S. H. and Evans, J. R., "Heuristic optimization: why, when, and how to use it", Interfaces 11, pp 84-91, 1981.

[70] Zupan, Jure, "Clustering of large data sets", Chemometrics research studies series. Research studies press, 1982.

Appendix A

MATHEMATICAL TEST PROBLEMS

Problem 1-1-0-0. Goldstein function Source: [2]

$$f(x) = x^6 - 15x^4 + 27x^2 + 250$$

Subjected to NO constraint.
Initial bounds:

$$-10 \leq x \leq 10$$

Solution in literature:

$$f(x^*) = 7 \; at \, x^* = \pm 3$$
$$\text{and } f(x) = 250 \; at \, x = 0$$

Solution using LSP:

$$f(x^*) = 7.000 \; at \, x^* = \pm 3$$

LSP Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 592 \qquad N_{keep} = 10$$

Problem 1-2-0-0. Source: [63]

$$f(x) = sin\, x + sin(10x/3) + ln\, x - 0.84x$$

Subjected to NO constraint.
Initial bounds:

$$2.7 \leq x \leq 7.5$$

Solution in literature:

$$f(x^*) = -1.6013075 \; at \, x^* = 5.1997784$$

There is an error, $f(x^*)$ should be -4.6013075 for the given x^*.
Computational effort using six different algorithms = (33, 29, 45, 462, 25, 120)
Solution using LSP:

$$f(x^*) = -4.6013074 \; at \, x^* = 5.1995869$$

LSP Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 110 \qquad N_{keep} = 10$$

Problem 1-3-0-0. Source: [63]

$$f(x) = sin\, x + sin(2x/3)$$

Subjected to NO constraint.
Initial bounds:

$$3.1 \leq x \leq 20.4$$

Solution in literature:

$$f(x^*) = -1.9059611 \; at\, x^* = 17.0391986$$

Computational effort using six different algorithms = (37, 38, 442, 448, 45, 158)
Solution using LSP:

$$f(x^*) = -1.9059612 \; at\, x^* = 17.0390205$$

LSP Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 104 \qquad N_{keep} = 10$$

Problem 1-4-0-0. Source: [63] Solution Improved

$$f(x) = -\sum_{i=1}^{5} sin((i+1)x + i)$$

Subjected to NO constraint.
Initial bounds:

$$-10 \leq x \leq 10$$

Solution in literature:

$$f(x^*) = -12.0312494 \; at\, x^* = -6.7745760, -0.4913908 \, or \, 5.7917947$$

There is an error, $f(x^*)$ should be -3.2649353 for the given points.
Computational effort using six different algorithms = (125, 165, 150, 3817, 161, 816)
Solution using LSP:

$$f(x^*) = -3.3728776 \; at\, x^* = -6.7187829, -0.4363553 \, or \, 5.8455772$$

LSP Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 2920 \qquad N_{keep} = 10$$

Problem 1-5-0-0. Source: [63]

$$f(x) = (x + sin\, x)e^{-x^2}$$

Subjected to NO constraint.
Initial bounds:

$$-10 \leq x \leq 10$$

Solution in literature:

$$f(x^*) = -0.6795797 \; at\, x^* = -0.8242384$$

The optimum point and its value are written interchanged
Computational effort using six different algorithms = (35, 34, 98, 376, 229, 83)
Solution using LSP:

$$f(x^*) = -0.824239319999 \; at\, x^* = -0.6793407$$

LSP Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 92 \qquad N_{keep} = 10$$

Problem 1-6-0-0. Source: [26]

$$f(x) = x^6 - \tfrac{52}{25}x^5 + \tfrac{39}{80}x^4 + 7.1x^3 - \tfrac{79}{20}x^2 - x + .1$$

Subject to:

$$-2 \leq x \leq 11$$

Initial bounds: As given in the constraint

Solution in literature:

$$f(x^*) = -29763.233 \; at\, x^* = 10$$

$$but\; f(x) = 803570.1 \text{ at } x = 10$$

Solution using LSP:

$$f(x^*) = -7.487312 \; at\, x^* = -1.19108$$

LSP Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 132 \qquad N_{keep} = 10$$

Problem 2-1-0-2. Source: [36]

$$f(\mathbf{x}) = sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1$$

Subject to:

$$-1.5 \leq x_1 \leq 4$$
$$-3 \leq x_2 \leq 3$$

Solution in literature:

$f(\mathbf{x}^*) = -1.913223$ at $\mathbf{x}^* = (-.54772, -1.5472)$

and $f(\mathbf{x}) = 1.2283700$ at $\mathbf{x} = (2.5944, 1.5944)$

Solution using LSP:

$$f(\mathbf{x}^*) = -1.9131677$$
$$\mathbf{x}^* = (-.54154, -1.54837)$$

LSP initial bounds:

$$-1.5 \leq x_1 \leq 4$$
$$-3 \leq x_2 \leq 3$$

Convergence criterion:

$$V_M \leq 1E - 5$$

LSP computational effort:

$N_f = 374$ $N_{keep} = 20$

Problem 2-2-1-1. Source: [58]

$$f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2$$

Subject to:

$$-.25x_1^2 - x_2^2 + 1 \geq 0$$
$$x_1 - 2x_2 + 1 = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = 1.39346$$
$$\mathbf{x}^* = (.82288, .9114)$$

Computational effort cited in literature (GRGA, FMIN): 108, 838

Solution using LSP:

$$f(\mathbf{x}^*) = 1.39352$$
$$\mathbf{x}^* = (.82285, .91143)$$

LSP initial bounds:

$$-2 \leq x_1 \leq 2$$
$$-1 \leq x_2 \leq 1$$

Convergence criterion:

$$V_M \leq 1E - 5$$

LSP computational effort:

$N_f = 303$ $N_{keep} = 20$

Problem 2-3-1-0. Source: [58]

$$f(\mathbf{x}) = (1 - x_1)^2$$

Subject to:

$$10(x_2 - x_1^2) = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = 0$$

$$\mathbf{x}^* = (1,1)$$

Computational effort cited in literature (GRGA, FMIN): 33, 546

Solution using LSP:

$$f(\mathbf{x}^*) = 4E - 7$$
$$\mathbf{x}^* = (1.0004, 1.0008)$$

LSP initial bounds:

$$-40 \leq x_1 \leq 40$$
$$0 \leq x_2 \leq 1600$$

Convergence criterion:

$$V_M \leq 1E - 5$$

LSP computational effort:

$$N_f = 188 \qquad N_{keep} = 20$$

Problem 2-4-1-0. Source: [58]

$$f(\mathbf{x}) = ln(1 + x_1^2) - x_2$$

Subject to:

$$(1 + x_1^2)^2 + x_2^2 - 4 = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = -1.73205$$
$$\mathbf{x}^* = (0, 1.73205)$$

Computational effort cited in literature (GRGA, FMIN): 204, 260

Solution using LSP:

$$f(\mathbf{x}^*) = -1.73204$$
$$\mathbf{x}^* = (0.00194, 1.73205)$$

initial bounds:

$$-1 \leq x_1 \leq 1$$
$$-2 \leq x_2 \leq 2$$

Convergence criterion:

$$V_M \leq 1E - 5$$

LSP computational effort:

$$N_f = 156 \qquad N_{keep} = 20$$

Problem 2-5-0-1. Source: [58]

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Subject to:

$$1.5 \leq x_2$$

Solution in literature:

$$f(\mathbf{x}^*) = 0.0504261879$$
$$\mathbf{x}^* = (1.224, 1.5)$$

Computational effort cited in literature (GRGA, FMIN): 115, 444

Solution using LSP:

$$f(\mathbf{x}^*) = 0.05044$$
$$\mathbf{x}^* = (1.22449, 1.50000)$$

LSP initial bounds:

$$0 \leq x_1 \leq 5$$
$$1.5 \leq x_2 \leq 5$$

Convergence criterion:

$$V_M \leq 1E-5$$

LSP computational effort:

$$N_f = 2088 \qquad N_{keep} = 20$$

Problem 2-6-0-3. Source: [58]

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1-x_1)^2$$

Subject to:

$$x_2 x_1 - 1 \geq 0$$
$$x_1 + x_2^2 \geq 0$$
$$x_1 \leq 0.5$$

Solution in literature:

$$f(\mathbf{x}^*) = 306.5$$
$$\mathbf{x}^* = (0.5, 2)$$

Computational effort cited in literature (GRGA, FMIN): 508, 2464

Solution using LSP:

$$f(\mathbf{x}^*) = 306.71$$
$$\mathbf{x}^* = (0.50000, 2.00059)$$

LSP initial bounds:

$$0 \leq x_1 \leq .5$$
$$2 \leq x_2 \leq 5$$

Convergence criterion:

$$V_M \leq 1E-3$$

LSP computational effort:

$$N_f = 1002 \qquad N_{keep} = 20$$

Problem 2-7-0-0. Rosenbrock function Source: [20], [58]

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1-x_1)^2$$

Subject to NO constraint.

Solution in literature:

$$f(\mathbf{x}^*) = 0$$
$$\mathbf{x}^* = (1, 1)$$

Computational effort cited in literature (GRGA, FMIN): 596, 638

Solution using LSP:

$$f(\mathbf{x}^*) = 1E-7$$
$$\mathbf{x}^* = (1.0001, 1.0002)$$

LSP initial bounds:

$$-5 \leq x_1 \leq 5$$
$$-5 \leq x_2 \leq 5$$

Convergence criterion:

$$V_M \leq 1E-12$$

LSP computational effort:

$$N_f = 3206 \qquad N_{keep} = 20$$

Problem 2-8-0-5. Source: [23]

$$f(\mathbf{x}) = -(x_1 - 2)^2 - (x_2 - 2)^2$$

Subject to:

$$x_1 + x_2 \geq 1$$
$$x_1 - 2x_2 \leq 1$$
$$2x_1 - x_2 \leq 5$$
$$3x_1 + 5x_2 \leq 27$$
$$-6x_1 + 10x_2 \leq 30$$

Solution in literature:

$$f(\mathbf{x}^*) = -5$$
$$\mathbf{x}^* = (0,1), (4,3), (0,3) \text{ or } (1,0)$$

Solution using LSP:

$$f(\mathbf{x}^*) = -4.999 \qquad \text{at } \mathbf{x}^* = (0,1)$$
$$f(\mathbf{x}^*) = -4.973 \qquad \text{at } \mathbf{x}^* = (0,2.987)$$
$$\text{and } f(\mathbf{x}^*) = -4.990 \qquad \text{at } \mathbf{x}^* = (0.995, 0.005)$$

A single run could not easily detect all 4 global points unless N_{keep} was raised above 30.

LSP initial bounds:

$$0 \leq x_1 \leq 9$$
$$0 \leq x_2 \leq 6$$

Convergence criterion:

$$V_M \leq 1E-3$$

LSP computational effort:

$$N_f = 24253 \qquad N_{keep} = 30$$

Problem 2-9-0-0. Source: [34]

$$f(\mathbf{x}) = (x_1^2 + 12x_2 - 1)^2 + (49x_1^2 + 49x_2^2 + 84x_1 + 2324x_2 - 681)^2$$

Subject to NO constraint.

Solution in literature:

$$f(\mathbf{x}) = 5.9225 \qquad \text{at } \mathbf{x} = (0.28581, 0.27936)$$
$$\text{and } f(\mathbf{x}^*) = 0.0000 \qquad \text{at } \mathbf{x}^* = (-21.026653, -36.760090)$$

Solution using LSP:

$$f(\mathbf{x}^*) = 2E-7$$
$$\mathbf{x}^* = (-21.0267, -36.7600)$$

LSP initial bounds:

$$-25 \leq x_1 \leq 5$$
$$-40 \leq x_2 \leq 5$$

Convergence criterion:

$$V_M \leq 1E-12$$

LSP computational effort:

$$N_f = 110032 \qquad N_{keep} = 50$$

The global optimum was missed with N_{keep} values below 40.

Problem 2-10-0-0. Source: [34]

$$f(\mathbf{x}) = -[e^{(-x_1^2-x_2^2)} * (2x_1^2 + 3x_2^2)]$$

Subject to NO constraint.

Solution in literature:

$f(\mathbf{x}^*) = -1.1036$
$\mathbf{x}^* = (0,1)$ or $\mathbf{x}^* = (0,-1)$

Solution using LSP:

$f(\mathbf{x}^*) = -1.1036$
$\mathbf{x}^* = (-0.0018, -0.9999)$ or $\mathbf{x}^* = (0.0007, 0.9976)$

LSP initial bounds:

$-3 \leq x_1, x_2 \leq 3$

Convergence criterion:

$V_M \leq 1E-8$

LSP computational effort:

$N_f = 5848 \qquad N_{keep} = 20$

Problem 2-11-0-0. Source: [34], [54]

$$f(\mathbf{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

Subject to NO constraint.

Solution in literature:

$f(\mathbf{x}^*) = 0$
$\mathbf{x}^* = (3.58443, -1.84813)$ or $\mathbf{x}^* = (3,2)$

Only two points are mentioned in [34].

Solution using LSP:

$f(\mathbf{x}^*) = 0.00000$ at $\mathbf{x}^* = (-3.779, -3.283)$
$f(\mathbf{x}^*) = 0.00009$ at $\mathbf{x}^* = (-2.806, 3.133)$
$f(\mathbf{x}^*) = 0.00088$ at $\mathbf{x}^* = (2.999, 1.998)$
$f(\mathbf{x}^*) = 0.00058$ at $\mathbf{x}^* = (3.586, -1.847)$

LSP initial bounds:

$-5 \leq x_1 \leq 5$
$-5 \leq x_2 \leq 5$

Convergence criterion:

$V_M \leq 1E-3$

LSP computational effort:

$N_f = 2853 \qquad N_{keep} = 30$

Problem 2-12-0-4. Source: [61]

$$f(\mathbf{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

Subject to:

$-x_1 + 13 \leq 0$
$-(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$
$(x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$
$-x_2 \leq 0$

Solution in literature:

$f(\mathbf{x}^*) = -6961.8106875$
$\mathbf{x}^* = (14.0950013, 0.8429636)$

Computational effort cited in the literature = 833

Solution using LSP:

$f(\mathbf{x}^*) = -6961.795410$
$\mathbf{x}^* = (14.0950079, 0.8429772)$

LSP initial bounds:

$$13 \leq x_1 \leq 16$$
$$0 \leq x_2 \leq 15$$

Convergence criterion:

$$V_M \leq 1E-2$$

LSP computational effort:

$$N_f = 8469 \qquad N_{keep} = 20$$

This problem was initially found to be a challenge to LSP and is discussed in Chapter 5, section 5.5.

Problem 2-13-0-0. Gear train of minimum inertia, (Fenton and Eason's Function)
Source: [54], [20]

$$f(\mathbf{x}) = [12 + x_1^2 + (1 + x_2^2)/x_1^2 + (x_1^2 x_2^2 + 100)/(x_1 x_2)^4]/10$$

Subject to:

$$1 \leq x_1, x_2 \leq 3$$

Solution in literature:

$$f(\mathbf{x}^*) = 1.74$$
$$\mathbf{x}^* = (1.7435, 2.0297)$$

Solution using LSP:

$$f(\mathbf{x}^*) = 1.7442$$
$$\mathbf{x}^* = (1.7333, 2.0449)$$

LSP initial bounds: As given in the set of constraints.
Convergence criterion:

$$V_M \leq 1E-7$$

LSP computational effort:

$$N_f = 200 \qquad N_{keep} = 20$$

Problem 2-14-0-0. Branin Equation Source: [63]

$$f(\mathbf{x}) = (x_2 - \tfrac{5.1}{4\pi^2} x_1^2 + \tfrac{5}{\pi} x_1 - 6)^2 + 10(1 - \tfrac{1}{8\pi}) cos\, x_1 + 10$$

Subject to:

$$-5 \leq x_1 \leq 10$$
$$0 \leq x_2 \leq 15$$

Solution in literature:

$$f(\mathbf{x}^*) = 0.398$$
$$\mathbf{x}^* = (-3.142, 12.275), (3.142, 2.275),\ or\, (9.425, 2.425)$$

Computational effort cited in literature: 134-11910
Solution using LSP:

$$f(\mathbf{x}^*) = 0.39901, 0.39791\, and\, 0.39852$$
$$\mathbf{x}^* = (3.135, 2.310), (9.427, 2.475),\ and\, (-3.152, 12.290)$$

LSP initial bounds: As given in the set of constraints.
Convergence criterion:

$$V_M \leq 1E-5$$

LSP computational effort:

$$N_f = 11895 \qquad N_{keep} = 30$$

Problem 2-15-0-0. Goldstein-2 Equation Source: [63]

$$f(\mathbf{x}) = (1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+x_2^2)*$$
$$(30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2))$$

Subject to:

$$-2 \leq x_1, x_2 \leq 2$$

Solution in literature:

$$f(\mathbf{x}^*) = 3$$
$$\mathbf{x}^* = (0, -1)$$

Computational effort cited in literature: 139 up to 4042

Solution using LSP:

$$f(\mathbf{x}^*) = 3.0000$$
$$\mathbf{x}^* = (-0.0002, -1.0003)$$

LSP initial bounds: As given in the set of constraints.

Convergence criterion:

$$V_M \leq 1E-5$$

LSP computational effort:

$$N_f = 522 \qquad N_{keep} = 20$$

Problem 2-16-0-0. Rastrigin Function Source: [63]

$$f(\mathbf{x}) = x_1^2 + x_2^2 - cos18x_1 - cos18x_2$$

Subject to:

$$-1 \leq x_1, x_2 \leq 1$$

Solution in literature:

There are about 50 local minima arranged in a lattice configuration, of which the following is the global minimum.

$$f(\mathbf{x}^*) = -2$$
$$\mathbf{x}^* = (0, 0)$$

Solution using LSP:

$$f(\mathbf{x}^*) = -1.9999$$
$$\mathbf{x}^* = (-0.0007, -.0001)$$

LSP initial bounds: As given in the set of constraints.

Convergence criterion:

$$V_M \leq 1E-5$$

LSP computational effort:

$$N_f = 2510 \qquad N_{keep} = 20$$

Problem 2-17-0-0. Source: [44], [63]

$$f(\mathbf{x}) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4$$

Subject to:

$$-5 \leq x_1, x_2 \leq 5$$

Solution in literature:

$$f(\mathbf{x}^*) = -1.0316285$$
$$\mathbf{x}^* = (-0.08983, 0.7126)\ or\ (0.08983, -0.7126)$$

Computational effort cited in literature: 40.8 [44]

Solution using LSP:

$f(\mathbf{x}^*) = -1.031570, -1.031559$
$\mathbf{x}^* = (-0.090677, 0.710088)\, and\, (0.093965, -0.713538)$

LSP initial bounds: As given in the set of constraints.
Convergence criterion:

$V_M \leq 1E-5$

LSP computational effort:

$N_f = 8620 \qquad N_{keep} = 20$

Problem 2-18-0-2. Source: [53]

$$f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2$$

Subject to:

$x_1^2 - x_2 \leq 0$
$x_1 + x_2 - 2 \leq 0$

Solution in literature:

$f(\mathbf{x}^*) = 1$
$\mathbf{x}^* = (1, 1)$

Computational effort cited in literature: (252-390)
Solution using LSP:

$f(\mathbf{x}^*) = 1.0033$
$\mathbf{x}^* = (0.9984, 0.9969)$

LSP initial bounds:

$-1.414 \leq x_1 \leq 1.414$
$0 \leq x_2 \leq 2$

Convergence criterion:

$V_M \leq 1E-5$

LSP computational effort:

$N_f = 560 \qquad N_{keep} = 40$

The global optimum was not reliably found with N_{keep} substantially less than 40.

Problem 2-19-0-1. Source: [20]

$$f(\mathbf{x}) = (.44x_1^3 x_2^{-2} + 10x_1^{-1} + 0.592x_1 x_2^{-3})/10$$

Subject to:

$1 - 8.62x_1^{-1} x_2^3 \geq 0$
$0 \leq x_1, x_2 \leq 5$

Solution in literature:

$f(\mathbf{x}^*) = 1.6206$
$\mathbf{x}^* = (1.2867, 0.53047)$

Solution using LSP:

$f(\mathbf{x}^*) = 1.6189$
$\mathbf{x}^* = (1.2862, 0.5308)$

LSP initial bounds: As given in the set of constraints.
Convergence criterion:

$V_M \leq 1E-8$

LSP computational effort:

$N_f = 3580 \qquad N_{keep} = 20$

Problem 2-20-0-0. Penalized Shubert Function Source:[2], [44] Solution Improved

$$f(\mathbf{x}) = \{\sum_{i=1}^{5} i\,cos[(i+1)x_1 + i]\}\{\sum_{i=1}^{5} i\,cos[i+1)x_2 + i]\}$$

$$+.5[(x_1 + 1.42513)^2 + (x_2 + 0.080032)^2]$$

Subject to:

$$-10 \leq x_1, x_2 \leq 10$$

Solution in literature:

The problem has 760 local minima of which the following is the single global minimum.

$f(\mathbf{x}^*) = -186.73091$

$\mathbf{x}^* = (-1.42513, -0.80032)$

Computational effort = 7711 [44], 8755, 76894 [2]

Solution using LSP:

$f(\mathbf{x}^*) = -186.7075$

$\mathbf{x}^* = (-1.4229, -0.7980)$ or $(-0.7984, -1.4217)$

LSP initial bounds: As given in the set of constraints.

Convergence criterion:

$V_M \leq 1E - 6$

LSP computational effort:

$N_f = 5516 \qquad N_{keep} = 20$

Problem 2-21-0-3. Source: [12], [28]

$$f(\mathbf{x}) = -[9 - (x_1 - 3)^2]\frac{x_2^3}{27\sqrt{3}}$$

Subject to:

$0 \leq x_1$

$0 \leq x_2 \leq \frac{x_1}{\sqrt{3}}$

$0 \leq x_1 + \sqrt{3}(x_2) \leq 6$

The problem has a different type of constraint in which the current limit for one of the variables (x_2) depends on the current value of another variable (x_1).

Solution in literature:

$f(\mathbf{x}^*) = -1$

$\mathbf{x}^* = (3, \sqrt{3})$

Solution using LSP:

$f(\mathbf{x}^*) = -0.99999$

$\mathbf{x}^* = (3.000, 1.732)$

Computational effort:

$N_f = 159[12]$ and $642[28]$

LSP initial bounds:

$0 \leq x_1 \leq 6$

$0 \leq x_2 \leq 4$

Convergence criterion:

$V_M \leq 1E - 6$

LSP computational effort:

$N_f = 481 \qquad N_{keep} = 20$

Problem 2-22-0-0. Source: None

$$f(\mathbf{x}) = 100 - (x_1 + x_2 - 10)^2$$

Subject to:

$0 \leq x_1, x_2 \leq 10$

True solution:

$f(\mathbf{x}^*) = 0$

$\mathbf{x}^* = (0, 0)$ *or* $(10, 10)$

Solution using LSP:

$f(\mathbf{x}^*) = 0.00013$

$\mathbf{x}^* = (0.0000, 0.0000)$ *or* $(10.0000, 10.0000)$

LSP initial bounds: As given in the set of constraints.

Convergence criterion:

$V_M \leq 1E - 7$

LSP computational effort:

$N_f = 6354 \qquad N_{keep} = 40$

Problem 3-1-1-0. Source: [58]

$$f(\mathbf{x}) = (x_1 + x_2)^2 + (x_2 + x_3)^2$$

Subject to:

$$x_1 + 2x_2 + 3x_3 - 1 = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = 0$$
$$\mathbf{x}^* = (0.5, -0.5, 0.5)$$

Computational effort, N_f, cited in literature (GRGA, FMIN): 177, 46

Solution using LSP:

$$f(\mathbf{x}^*) = 7E - 5$$
$$\mathbf{x}^* = (0.49493, -0.48871, 0.49417)$$

LSP initial bounds:

$$-5 \leq x_1, x_2, x_3 \leq 5$$

convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 578 \qquad N_{keep} = 30$$

Problem 3-2-1-0. Source: [58]

$$f(\mathbf{x}) = 0.01(x_1 - 1)^2 + (x_2 - x_1^2)^2$$

Subject to:

$$x_1 + x_3^2 + 1 = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = 0.04$$
$$\mathbf{x}^* = (-1, 1, 0)$$

Computational effort, N_f, cited in literature (GRGA, FMIN): 962, 346

Solution using LSP:

$$f(\mathbf{x}^*) = .040084$$
$$\mathbf{x}^* = (-1.0012, 1.0084, 0.0342)$$

LSP initial bounds:

$$-10 \leq x_1 \leq -1$$
$$-5 \leq x_2, x_3 \leq 5$$

Convergence criterion:

$$V_M \leq 1E - 5$$

LSP computational effort:

$$N_f = 696 \qquad N_{keep} = 30$$

Problem 3-3-0-2. Source: [20], [59]

$$f(\mathbf{x}) = -x_1 x_2 x_3$$

Subject to:

$$x_1 + 2x_2 + 2x_3 \geq 0$$
$$72 - x_1 - 2x_2 - 2x_3 \geq 0$$
$$0 \leq x_1 \leq 20$$
$$0 \leq x_2 \leq 11$$
$$0 \leq x_3 \leq 42$$

Solution in literature:

$$f(\mathbf{x}^*) = -3300$$
$$\mathbf{x}^* = (20, 11, 15)$$

Computational effort, N_f, cited in literature (NLPQL): 25 [59]

Solution using LSP:

$$f(\mathbf{x}^*) = -3299.946$$
$$\mathbf{x}^* = (20.0000, 10.9993, 15.0007)$$

LSP initial bounds: As given in the set of constraint

Convergence criterion:

$$V_M \leq 1E-5$$

LSP computational effort:

$$N_f = 8886 \qquad N_{keep} = 30$$

Problem 3-4-2-0. Source: [52]

$$f(\mathbf{x}) = -1 * (3x_1^2 + 2x_2^2 - x_3)$$

Subject to:

$$x_1^2 + x_2^2 = 25$$
$$9x_1 - x_2^2 + x_3 = 27$$

Solution in literature:

$$f(\mathbf{x}^*) = -66$$
$$\mathbf{x}^* = (4, 3, 0)$$

Solution using LSP:

$$f(\mathbf{x}^*) = -66$$
$$\mathbf{x}^* = (4, 3, 0)$$

LSP initial bounds:

$$0 \leq x_i \leq 5 \qquad \text{for } i = 1, 3$$

Convergence criterion:

$$V_M \leq 1E-3$$

LSP computational effort:

$$N_f = 877 \qquad N_{keep} = 30$$

Problem 3-5-2-0. Source: [59]

$$f(\mathbf{x}) = log(x_3) - x_2$$

Subject to:

$$x_2^2 + x_3^2 - 4 = 0$$
$$x_3 - 1 - x_1^2 = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = -1.73205$$
$$\mathbf{x}^* = (0, 1.732, 1)$$

Computational effort, N_f, cited in literature (NLPQL): 54

Solution using LSP:

$$f(\mathbf{x}^*) = -1.732$$
$$\mathbf{x}^* = (-.0225, 1.73176, 1.00050)$$

LSP initial bounds:

$$-1 \leq x_1 \leq 1$$
$$-2 \leq x_2 \leq 2$$
$$1 \leq x_3 \leq 2$$

Convergence criterion:
$$V_M \leq 1E-3$$
LSP computational effort:
$$N_f = 88 \qquad N_{keep} = 20$$

Problem 3-6-0-1. Source: [59]

$$f(\mathbf{x}) = 0.2/(x_1x_2x_3) + 4/x_1 + 3/x_3$$
Subject to:
$$10 - 2x_1x_3 - x_1x_2 \geq 0$$
Solution in literature:
$$f(\mathbf{x}^*) = 3.36168$$
$$\mathbf{x}^* = (2.380, 0.3162, 1.943)$$
Computational effort, N_f, cited in literature (NLPQL): 41
Solution using LSP:
$$f(\mathbf{x}^*) = 3.3628$$
$$\mathbf{x}^* = (2.3699, 0.3162, 1.9505)$$
LSP initial bounds:
$$0 \leq x_1, x_2, x_3 \leq 10$$
Convergence criterion:
$$V_M \leq 1E-3$$
LSP computational effort:
$$N_f = 2804 \qquad N_{keep} = 20$$

Problem 3-7-0-2. Source: [58]

$$f(\mathbf{x}) = 0.2x_3 - 0.8x_1$$
Subject to:
$$x_2 - exp(x_1) \geq 0$$
$$x_3 - exp(x_2) \geq 0$$
$$0 \leq x_1, x_2 \leq 100$$
$$0 \leq x_3 \leq 10$$
Solution in literature:
$$f(\mathbf{x}^*) = 0.5181632741$$
$$\mathbf{x}^* = (0.1841264879, 1.202167873, 3.327322322)$$
Computational effort, N_f, cited in literature (GRGA, FMIN): 208, 1917
Solution using LSP:
$$f(\mathbf{x}^*) = 0.51858$$
$$\mathbf{x}^* = (0.1777, 1.1945, 3.3038)$$
LSP initial bounds: As given in the set of constraints.
Convergence criterion:
$$V_M \leq 1E-7$$
LSP computational effort:
$$N_f = 65973 \qquad N_{keep} = 40$$

Problem 3-8-1-1. Source: [58]

$$f(\mathbf{x}) = (x_1 + 3x_2 + x_3)^2 + 4(x_1 - x_2)^2$$

Subject to:

$$6x_2 + 4x_3 - x_1^3 - 3 \geq 0$$
$$1 - x_1 - x_2 - x_3 = 0$$
$$0 \leq x_i \text{ for } i = 1, 3$$

Solution in literature:

$$f(\mathbf{x}^*) = 1$$
$$\mathbf{x}^* = (0, 0, 1)$$

Computational effort, N_f, cited in literature (GRGA, FMIN): 138, 3202

Solution using LSP:

$$f(\mathbf{x}^*) = 1.0006$$
$$\mathbf{x}^* = (0.0111, 0.0000, 0.9889)$$

LSP initial bounds:

$$0 \leq x_1, x_2, x_3 \leq 1$$

Convergence criterion:

$$V_M \leq 1E-6$$

LSP computational effort:

$$N_f = 661 \qquad N_{keep} = 30$$

Problem 3-9-0-0. Source: [58]

$$f(\mathbf{x}) = \sum_{i=1}^{99} (y_i(\mathbf{x}))^2$$

$$y_i(\mathbf{x}) = -0.01i + exp(-1/x_1(u_i - x_2)^{x_3})$$
$$u_i = 25 + (-50ln(0.01i))^{2/3}$$

Subject to:

$$0.1 \leq x_1 \leq 100$$
$$0 \leq x_2 \leq 25.6$$
$$0 \leq x_3 \leq 5$$

Solution in literature:

$$f(\mathbf{x}^*) = 0$$
$$\mathbf{x}^* = (50, 25, 1.5)$$

Computational effort, N_f, cited in literature (GRGA, FMIN): 506, 873

Solution using LSP:

$$f(\mathbf{x}^*) = 0.26E-5$$
$$\mathbf{x}^* = (52.993, 24.924, 1.519)$$

The problem is very insensitive to x_1.

LSP initial bounds: As given in the set of constraints.

Convergence criterion:

$$V_M \leq 1E-6$$

LSP computational effort:

$$N_f = 6000 \qquad N_{keep} = 40$$

Problem 3-10-0-0. Hartman Family; Source: [19] Solution Improved

$$f(\mathbf{x}) = -\sum_{i=1}^{m} c_i \; exp[-\sum_{j=1}^{n} a_{ij}(x_j - p_{ij})^2]$$

Values of c_i, a_{ij} and p_{ij} are given in Table A.1. Subject to:

$0 \leq x_j \leq 1$ for $j = 1, n$

i	a_{ij}			c_i	p_{ij}		
1	3.0	10	30	1.0	0.3689	0.1170	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.7470
3	3.0	10	30	3.0	0.1091	0.8732	0.5547
4	0.1	10	35	3.2	0.03815	0.5743	0.8828

Table A.1: Data for m = 4 and n = 3

Solution in literature:

$f(\mathbf{x}^*) = -3.8626$
$\mathbf{x}^* = (0.1310, 0.5556, 0.8526)$

Solution using LSP:

$f(\mathbf{x}^*) = -3.08796$ at $\mathbf{x}^* = (0.113, 0.868, 0.561)$
$f(\mathbf{x}^*) = -3.68868$ at $\mathbf{x}^* = (0.700, 0.555, 0.857)$
$f(\mathbf{x}^*) = -3.86131$ at $\mathbf{x}^* = (0.095, 0.555, 0.849)$
$f(\mathbf{x}^*) = -3.86023$ at $\mathbf{x}^* = (0.108, 0.552, 0.853)$

LSP initial bounds:

$0 \leq x_1, x_2, x_3 \leq 1$

Convergence criterion:

$V_M \leq 0.01$

LSP computational effort:

$N_f = 1496 \qquad N_{keep} = 35$

Problem 3-11-0-2. Source: [14]

$$f(\mathbf{x}) = 314.16x_1x_2 + 408.41x_1^2 + 100x_3$$

Subject to:

$1592.36x_1^{-2}x_2^{-1} + 0.125x_1^{-2}x_2^{-1}x_3 \leq 1$
$x_2^2x_3^{-2} + 4x_1^2x_3^{-2} \leq 1$
$x_i \geq 0$ for $i = 1, 3$

Solution in literature:

$f(\mathbf{x}^*) = 91246.018$
$\mathbf{x}^* = (8.648, 21.339, 27.468)$

Solution using LSP:

$f(\mathbf{x}^*) = 91260.79$
$\mathbf{x}^* = (8.6241, 21.4560, 27.5314)$

LSP initial bounds:

$0 \leq x_1, x_2, x_3 \leq 40$

Convergence criterion:

$V_M \leq 1E-2$

LSP computational effort:

$N_f = 304113 \qquad N_{keep} = 30$

Problem 3-12-0-2. Source: [14]

$$f(\mathbf{x}) = 400x_1^{.4} + 4.183x_1^{-1.2}x_3^{1.2} + 10x_1^{-.5}x_2x_3^{.5}$$

Subject to:

$x_3^{-1} + x_2 \leq 1$
$0.1975x_1^{-.5}x_2^{-.5} + 0.1975x_1^{-.5}x_2^{-.5}x_3^{-1} \leq 1$
$x_i \geq 0$ for $i = 1, 3$

Solution in literature:

$f(\mathbf{x}^*) = 280.623$
$\mathbf{x}^* = (0.238, 0.413, 1.704)$

Solution using LSP:

$f(\mathbf{x}^*) = 280.674$
$\mathbf{x}^* = (0.240, 0.410, 1.696)$

LSP initial bounds:

$0 \leq x_1 \leq 5$
$0 \leq x_2 \leq 1$
$1 \leq x_3 \leq 5$

Convergence criterion:

$V_M \leq 1E-3$

LSP computational effort:

$N_f = 2556 \qquad N_{keep} = 30$

Problem 3-13-2-0. Source: [46]

$$f(\mathbf{x}) = -x_1^2 - x_2^2 - x_3^2$$

Subject to:

$x_1 + 2x_2 + 3x_3 = 1$
$x_1^2 + \frac{x_2^2}{2} + \frac{x_3^2}{4} = 4$

The constraints form ellipses.

Solution in literature:

$f(\mathbf{x}^*) = -9.995$ and other local optima values $-9.051, -4.430, -4.049$

Four optimal values are given though the corresponding points are not specified.

$\mathbf{x}^* = (\textit{not given})$

Solution using LSP:

$f(\mathbf{x}^*) = -9.9946$
$\mathbf{x}^* = (-0.16401, -2.43407, 2.201072)$

LSP initial bounds:

$-2 \leq x_1 \leq 2$
$-2.828 \leq x_2 \leq 2.828$
$-4 \leq x_3 \leq 4$

Convergence criterion:

$V_M \leq 1E-4$

LSP computational effort:

$$N_f = 387287 \qquad N_{keep} = 30$$

Problem 3-14-0-1. Fuel allocation problem. Source: [46], [54]

$$W_1 = 1.4609 + 0.15186x_1 + 0.00145x_1^2$$
$$W_2 = 1.5742 + 1.631x_1 + 0.001358x_1^2$$
$$Y_1 = 0.8008 + 0.2031(50 - x_1) + 0.000916(50 - x_1)^2$$
$$Y_2 = 0.7266 + 0.2256(50 - x_1) + 0.000778(50 - x_1)^2$$
$$f(\mathbf{x}) = x_2 W_1 + x_3 Y_1$$

Subject to:

$$(1 - x_2)W_2 + (1 - x_3)Y_2 \leq 10$$

Solution in literature (53):

$$f(\mathbf{x}^*) = 3.05$$
$$\mathbf{x}^* = (30, 0.00, 0.58)$$

Solution using LSP:

Number of functional evaluation cited in Reklaitis = 8600, AIChE = 2989

$$f(\mathbf{x}^*) = 3.0557$$
$$\mathbf{x}^* = (29.6843, 0.0018, 0.5736)$$

Convergence criterion:

$$V_M \leq 1E - 5$$

LSP computational effort:

$$N_f = 62275 \qquad N_{keep} = 30$$

Problem 3-15-0-2. Flywheel design Source: [20]

$$f(\mathbf{x}) = -0.0201x_1^4 x_2 x_3^2/10^7$$

Subject to:

$$675 - x_1^2 x_2 \geq 0$$
$$0.419 - (x_1 x_3)^2/10^7 \geq 0$$
$$0 \leq x_1 \leq 36$$
$$0 \leq x_2 \leq 5$$
$$0 \leq x_3 \leq 125$$

Solution in literature:

$$f(\mathbf{x}^*) = -5.684802$$
$$\mathbf{x}^* = (\textit{not given})$$

Solution using LSP:

$$f(\mathbf{x}^*) = -5.6830$$
$$\mathbf{x}^* = (18.802, 1.909, 108.871)$$

LSP initial bounds: As given in the set of constraints

Convergence criterion:

$$V_M \leq 1E - 5$$

LSP computational effort:

$$N_f = 114824 \qquad N_{keep} = 30$$

The slow convergence is discussed in Chapter 5, section 5.3.

Problem 3-16-0-2. Source: [67]

$$f(\mathbf{x}) = -[x_1^2 + x_2^2 + x_3^2]$$

Subject to:

$$4(x_1-.5)^2+2(x_2-.2)^2+x_3^2+.1x_1x_2+.2x_2x_3 \le 16$$
$$2x_1^2+x_2^2-2x_3^2 \ge 2$$
$$-2.3 \le x_1, x_2, x_3 \le 2.7$$

Solution in literature:

The problem has four local optima, of which the global value is

$$f(\mathbf{x}^*) = -11.67664$$
$$\mathbf{x}^* = (0.988, 2.674, -1.884)$$

Solution using LSP:

$$f(\mathbf{x}^*) = -11.676620$$
$$\mathbf{x}^* = (0.9853, 2.6752, -1.8839)$$

LSP initial bounds: As given in the set of constraints

Convergence criterion:

$$V_M \le 1E-8$$

LSP computational effort:

$$N_f = 1032281 \qquad N_{keep} = 30$$

Problem 3-17-0-0. Nonlinear regression model fitting. Source: [18]

Problem formulation:

The following formulation is directly taken from the source literature.

An article in Lubrication Engineering ("Accelerated Testing of Solid Film Lubricants", 1972, pp 365-372) reported on an investigation of wear life for solid film lubricant. Three sets of journal bearing tests were run on a Mil-L-8937 type film at each combination of three loads(3000, 6000 and 10000 psi) and three speeds (20, 60 and 100 rpm) and the wear life (hours) was recorded for each run. It is also known that a three parameter nonlinear model can fit the experimental data. The model is written as

$$w = c * s^a * l^b$$

where

w is the wear life (hours)
s is the speed (rpm)
l is the load (psi)
a, b and c are the model parameters.

It is required to determine the parameters for the multiple nonlinear regression model using the experimental data given in Table A.2. The model is expected to produce the minimum sum of squares between the recorded and simulated wear life.

Method used in the literature:

The data was linearized by transforming to logarithmic values so that a multiple linear model could be used. The linear model was then solved using the least squares method.

LSP solution:

The nonlinear model was solved using LSP without any linearization. The solutions cited in the literature and those found with LSP are given in Table A.3. The solutions found with LSP showed about 4% improvement over the solution cited in the literature.

s	l	w	s	l	w
20	3000	300.20	60	6000	65.90
20	3000	310.80	60	10000	10.70
20	3000	333.00	60	10000	34.10
20	6000	99.60	60	10000	39.10
20	6000	136.20	100	3000	26.50
20	6000	142.40	100	3000	22.30
20	10000	20.20	100	3000	34.80
20	10000	28.20	100	6000	32.80
20	10000	102.70	100	6000	25.60
60	3000	67.30	100	6000	32.70
60	3000	77.90	100	10000	2.30
60	3000	93.90	100	10000	4.40
60	6000	43.00	100	10000	5.80
60	6000	44.50			

Table A.2: Experimental data used to fit the nonlinear regression model

It took 19842 function evaluations to meet the convergence criterion of $V_M \leq 1E-5$, and this was achieved in 3 minutes on an 80386 machine.

Parameter	Literature	LSP
a	-1.20603	-1.2319
b	-1.39876	-1.3371
c	$8.3 * 10^8$	$5.6 * 10^8$
$\sum(w-\overline{w})^2$	10748.95	10312.1
Improvement	—	4.06%

Table A.3: Optimal solution for the regression model

Problem 4-1-0-3. Source: [59]

$$f(\mathbf{x}) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

Subject to:

$$-x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 + x_3 + x_4 + 8 \geq 0$$
$$-x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 9 \geq 0$$
$$-2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \geq 0$$

Solution in literature:

$$f(\mathbf{x}^*) = -44$$
$$\mathbf{x}^* = (0, 1, 2, -1)$$

Computational effort, N_f, cited in literature (NLPQL): 95

Solution using LSP:

$$f(\mathbf{x}^*) = -44.097$$
$$\mathbf{x}^* = (0.01337, 0.84628, 1.99776, -1.11141)$$

LSP initial bounds:

$$-5 \leq x_i \leq 5 \qquad \text{for } i = 1, 4$$

Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 16701 \qquad N_{keep} = 40$$

Problem 4-2-0-6. Source: [58]

$$f(\mathbf{x}) = x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4$$

Subject to:

$$8 - x_1 - 2x_2 \geq 0$$
$$12 - 4x_1 - x_2 \geq 0$$
$$12 - 3x_1 - 4x_2 \geq 0$$
$$8 - 2x_3 - x_4 \geq 0$$
$$8 - x_3 - 2x_4 \geq 0$$
$$5 - x_3 - x_4 \geq 0$$
$$0 \leq x_i \qquad \text{for } i = 1, 4$$

Solution in literature:

$$f(\mathbf{x}^*) = -15$$
$$\mathbf{x}^* = (0, 3, 0, 4)$$

Computational effort, N_f, cited in literature (GRGA, FMIN): 203, 8547

Solution using LSP:

$$f(\mathbf{x}^*) = -14.988$$
$$\mathbf{x}^* = (0.0004, 2.9995, 0.004, 3.9976)$$

There is a local optima at $x = (3, 0, 4, 0)$ with $f(\mathbf{x}) = 13$

LSP initial bounds:

$$0 \leq x_1, x_2 \leq 3$$
$$0 \leq x_3, x_4 \leq 4$$

Convergence criterion:

$$V_M \leq 1E - 4$$

LSP computational effort:

$$N_f = 5000 \qquad N_{keep} = 40$$

Problem 4-3-2-0. Source: [52]

$$f(\mathbf{x}) = x_1^2 + 4x_2^2$$

Subject to:

$$x_1 + 2x_2 - x_3 = 1$$
$$-x_1 + x_2 + x_4 = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = 0.5$$
$$\mathbf{x}^* = (0.5, 0.25, 0, 0.25)$$

Solution using LSP:

$$f(\mathbf{x}^*) = 0.5001$$
$$\mathbf{x}^* = (0.5023, 0.2489, 0.0001, 0.2535)$$

LSP initial bounds:

$$0 \leq x_i \leq 1 \qquad \text{for } i = 1, 4$$

Convergence criterion:

$$V_M \leq 1E - 6$$

LSP computational effort:

$$N_f = 4696 \qquad N_{keep} = 40$$

Problem 4-4-1-1. Source: [58]

$$f(\mathbf{x}) = x_1 x_4 (x_1 + x_2 + x_3) + x_3$$

Subject to:

$$x_1 x_2 x_3 x_4 - 25 \geq 0$$
$$x_1^2 + x_2^2 + x_3^2 + x_4^2 - 40 = 0$$
$$1 \leq x_i \leq 5 \qquad \text{for } i = 1, 4$$

Solution in literature:

$$f(\mathbf{x}^*) = 17.0140173$$
$$\mathbf{x}^* = (1, 4.7429994, 3.8211503, 1.3794082)$$

Computational effort, N_f, cited in literature (GRGA, FMIN): 411, 6175

Solution using LSP:

$$f(\mathbf{x}^*) = 17.015$$
$$\mathbf{x}^* = (1.0002, 4.7565, 3.8034, 1.3817)$$

LSP initial bounds:

$$1 \leq x_i \leq 5 \qquad \text{for } i = 1, 4$$

Convergence criterion:

$$V_M \leq 1E - 5$$

LSP computational effort:

$$N_f = 15186 \qquad N_{keep} = 40$$

Problem 4-5-0-0. Wood's function Source: [20], [34], [54]

$$f(\mathbf{x}) = 100(x_2 - x_1^2) + (1 - x_1)^2 + 90(x_4 - x_3^2) + (1 - x_3)^2$$
$$+10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$$

Subject to:

$$-10 \leq x_i \leq 10 \qquad \text{for } i = 1, 4$$

Solution in literature:

$$f(\mathbf{x}^*) = 0$$

$\mathbf{x}^* = (1,1,1,1)$

Solution using LSP:

$f(\mathbf{x}^*) = 0.00062$

$\mathbf{x}^* = (1.011, 1.023, 0.988, 0.978)$

LSP initial bounds:

$-10 \leq x_i \leq 10 \quad \text{for } i = 1,4$

Convergence criterion:

$V_M \leq 1E-3$

LSP computational effort:

$N_f = 50824 \qquad N_{keep} = 40$

Problem 4-6-1-2. Source: [58]

$f(\mathbf{x}) = 24.55x_1 + 26.75x_2 + 39x_3 + 40.50x_4$

Subject to:

$x_1 + x_2 + x_3 + x_4 - 1 = 0$

$2.3x_1 + 5x_2 + 11.1x_3 + 1.3x_4 - 5 \geq 0$

$12x_1 + 11.9x_2 + 41.8x_3 + 52.1x_4 - 21$

$-1.645(0.28x_1^2 + 0.19x_2^2 + 20.5x_3^2 + 0.62x_4^2)^{1/2} \geq 0$

$0 \leq x_i \quad \text{for } i = 1,4$

Solution in literature:

$f(\mathbf{x}^*) = 29.894378$

$\mathbf{x}^* = (0.355216, -.12E-11, 0.3127019, 0.05177655)$

Constraints are violated.

Computational effort, N_f, cited in literature (GRGA, FMIN): 394, 4620

Solution using LSP:

$f(\mathbf{x}^*) = 29.8947$

$\mathbf{x}^* = (0.6351, 0.0004, 0.3126, 0.0518)$

LSP initial bounds:

$0 \leq x_i \leq 1 \quad \text{for } i = 1,4$

Convergence criterion:

$V_M \leq 1E-3$

LSP computational effort:

$N_f = 13924 \qquad N_{keep} = 40$

Problem 4-7-2-0. Source: [59]

$$f(\mathbf{x}) = \sum_{i=1}^{2}[1 - exp(-10x_i exp(-x_{i+2}))]$$

Subject to:

$x_1 + x_2 - 1 = 0$

$x_3 + x_4 - 1 = 0$

$0 \leq x_i \leq 1 \quad \text{for } i = 1,4$

Solution in literature:

$f(\mathbf{x}^*) = 0.974747$

$\mathbf{x}^* = (1,0,1,0)$

Computational effort, N_f, cited in literature (NLPQL): 12
Solution using LSP:

$$f(\mathbf{x}^*) = 0.9749$$
$$\mathbf{x}^* = (1.0000, 0.0000, 0.9995, 0.0005)$$

LSP initial bounds:

$$0 \leq x_i \leq 1 \qquad \text{for } i = 1,4$$

Convergence criterion:

$$V_M \leq 1E-3$$

LSP computational effort:

$$N_f = 800 \qquad N_{keep} = 40$$

Problem 4-8-0-3. Source: [14] Solution Improved

$$f(\mathbf{x}) = x_1 x_3^{-1} x_4 + 2x_1 + x_1 x_4 + x_2^{-1}$$

Subject to:

$$x_1^{-1} x_2 + x_1^{-1} \leq 1$$
$$x_1 x_4^{-1} \leq 1$$
$$x_1 x_3 \leq 1$$
$$x_i \geq 0 \qquad \text{for } i = 1,4$$

Solution in literature:

$$f(\mathbf{x}^*) = 11.946$$
$$\mathbf{x}^* = (1.267, 0.267, 0.789, 1.267)$$

$f(\mathbf{x}*) = 9.919$ *when the given* $\mathbf{x}$ *value is used.*

Solution using LSP:

$$f(\mathbf{x}^*) = 9.8089$$
$$\mathbf{x}^* = (1.3216, 0.3216, 0.7566, 1.3219)$$

LSP initial bounds:

$$0 \leq x_i \leq 2 \qquad \text{for } i = 1,4$$

Convergence criterion:

$$V_M \leq 1E-5$$

LSP computational effort:

$$N_f = 178145 \qquad N_{keep} = 40$$

Problem 4-9-0-0. Shekel's family. Source: [19]

$$f(\mathbf{x}) = -\sum_{i=1}^{m} [(\sum_{j=1}^{n} (x_j - a_{ij})^2) + c_i]^{-1}$$

Values of a_{ij} and c_i are given in Table A.4.
Subject to:

$$0 \leq x_j \leq 10 \text{ for } j = 1, n$$

Solution in literature:

$$f(\mathbf{x}^*) = -10.1532$$
$$\mathbf{x}^* = (4, 4, 4, 4)$$

Solution using LSP:

i	a_{ij}				c_i
1	4	4	4	4	0.1
2	1	1	1	1	0.2
3	8	8	8	8	0.2
4	6	6	6	6	0.4
5	3	7	3	7	0.4

Table A.4: Data for m = 5 and n = 4

$$f(\mathbf{x}^*) = -10.1528$$
$$\mathbf{x}^* = (4.002, 4.000, 3.999, 4.001)$$

LSP initial bounds:
$$0 \leq x_i \leq 10 \qquad \text{for } i = 1,4$$

Convergence criterion:
$$V_M \leq 1E-3$$

LSP computational effort:
$$N_f = 14008 \qquad N_{keep} = 80$$

Problem 4-10-1-2. Source: [26] Solution Improved

$$f(\mathbf{x}) = x_1^{.6} + 2x_2^{.6} + 2x_3 - 2x_2 - x_4$$

Subject to:
$$x_2 - 3x_1 - 3 = 0$$
$$x_1 + 2x_3 \leq 4$$
$$x_2 + x_4 \leq 4$$
$$x_1 \leq 3$$
$$x_4 \leq 2$$
$$x_i \geq 0 \qquad \text{for } i = 1,4$$

Solution in literature:
$$f(\mathbf{x}^*) = -2.07$$
$$\mathbf{x}^* = (4/3, 4, 0, 0)$$

Solution using LSP:
$$f(\mathbf{x}^*) = -3.1325$$
$$\mathbf{x}^* = (0.0000, 3.0000, 0.0001, 0.9995)$$

LSP initial bounds:
$$0 \leq x_1 \leq 3$$
$$0 \leq x_2 \leq 4$$
$$0 \leq x_3 \leq 2$$
$$0 \leq x_4 \leq 2$$

Convergence criterion:
$$V_M \leq 1E-7$$

LSP computational effort:
$$N_f = 5035 \qquad N_{keep} = 40$$

Problem 4-11-0-0. Source: [68]

Problem formulation:

The following formulation is directly taken from the source literature.

An electronic products manufacturer undertook the production of a new product in two locations (location A: coded x = 1, location B: coded x = 0). Location B has more modern facilities and hence was expected to be more efficient than location A, even after the initial learning period. An industrial engineer calculated the expected unit production cost for a modern facility after learning has occurred. Weekly unit production costs for each location were then expressed as a fraction of this expected cost. The reciprocal of this fraction is a measure of relative efficiency, and this measure was utilized as the efficiency measure in this study.

The model used was

$$Y_i = \gamma_0 + \gamma_1 x_i + \gamma_3 exp(\gamma_2 w_i) + \epsilon_i$$

Here γ_0 is the upper asymptote for location B as w gets large, and $\gamma_0 + \gamma_1$ is the upper asymptote for location A. The parameters γ_2 and γ_3 reflect the speed of learning, which was expected to be the same in the two locations.

The data on location, time and relative efficiency are presented in Table A.5. Note that the relative efficiency in location B toward the end of the 90 week period even exceeded 1.0, i.e. the actual unit costs then were lower than the industrial engineer's expected unit cost.

Location	Week	Rel. eff.	Location	Week	Rel. eff.
x_i	w_i	Y_i	x_i	w_i	Y_i
1	1	0.483	0	1	0.517
1	2	0.539	0	2	0.598
1	3	0.618	0	3	0.635
1	5	0.707	0	5	0.75
1	7	0.762	0	7	0.811
1	10	0.815	0	10	0.848
1	15	0.881	0	15	0.943
1	20	0.919	0	20	0.971
1	30	0.964	0	30	1.012
1	40	0.959	0	40	1.015
1	50	0.968	0	50	1.007
1	60	0.971	0	60	1.022
1	70	0.96	0	70	1.028
1	80	0.967	0	80	1.017
1	90	0.975	0	90	1.023

Table A.5: Experimental data used to fit the nonlinear regression model

Method used in the literature:

A direct search computer program was used to estimate the four parameters in the model. A reasonable starting point was used as input to run the computer package. The resulting least squares regression function was

$$\hat{Y} = 1.0156 - 0.04727x - (0.5524)exp(-0.1348w)$$

and the error sum of squares was $SSE = 0.00329$.

LSP solution:

Output from LSP confirmed the results given in the source reference. It took 2128 function evaluations to meet a termination criterion of $V_M \leq 1E-5$.

Problem 4-12-1-5. Source: [57]

$$f(\mathbf{x}) = s_0(\mathbf{x}) + \tfrac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + q^T\mathbf{x} + \alpha$$

H is $(n \times n)$ matrix, $q \in \mathbf{R}^n$, $\alpha \in \mathbf{R}$ and **T** designates the transpose of a matrix. Subject to:

$$\begin{aligned}
&s_1(\mathbf{x}) + 40.99 = 0\\
&s_2(\mathbf{x}) + 3.98 \geq 0\\
&s_3(\mathbf{x}) + 4.61 \geq 0\\
&s_4(\mathbf{x}) + 32.06 \geq 0\\
&s_5(\mathbf{x}) - 2.71 \geq 0\\
&s_6(\mathbf{x}) + 2.27 \geq 0\\
&1 \leq x_i \leq 4 \qquad \text{for } i = 1,2,3,4
\end{aligned}$$

where

$$\begin{aligned}
s_0(\mathbf{x}) &= .32x_1^{2.82}x_2^{2.16} - .88x_3^{2.86} + .81x_2^{1.07}x_3^{1.48}x_4^{1.4} + .66x_1^{1.56} - .83x_1^{2.3}x_2^{2.40}x_3^{2.27}\\
s_1(\mathbf{x}) &= .96 - 1.12x_3x_4^2 + 3.13x_1^{-1}x_2^{-1}x_3^{-1}x_4^2 + 3.3x_1^2 - 4.84x_1^{-1}x_3\\
s_2(\mathbf{x}) &= 5.39 + 1.98x_2x_3^2x_4^{-1} - 4.14x_2 - 1.88x_1x_4\\
s_3(\mathbf{x}) &= .99 - 4.64x_3x_4^{-1} - .57x_4\\
s_4(\mathbf{x}) &= -3.23 - 1.89x_2^2 - 1.18x_3^2 - 2.01x_2^2x_4^{-1} - 1.99x_1\\
s_5(\mathbf{x}) &= 3.45 - 2.98x_1x_2 - .96x_1^2x_3^2 - 2.90x_2^{-1}x_4^{-1} + 3.39x_3^2\\
s_6(\mathbf{x}) &= -.51 - 2.89x_2^2x_3 - 6.24x_2^2 + 2.15x_3^2x_4
\end{aligned}$$

$$\mathbf{H} = \begin{pmatrix} 171.49 & 200.07 & 156.82 & -3.12\\ 200.07 & 78.48 & 86.63 & -12.85\\ 156.82 & 86.63 & 51.75 & -20.46\\ -3.12 & -12.85 & -20.46 & -5.07 \end{pmatrix}$$

$$q = (-908.19, -546.53, -407.35, 64.17)^T$$

$$\alpha = 1499.97$$

Solution in the literature:

$$\mathbf{x}^* = (1.35, 2.17, 2.80, 3.71)$$

Objective function calculated at optimal point = -0.2298

Solution using LSP:

$$f(\mathbf{x}^*) = -0.2716$$

$$\mathbf{x}^* = (1.341152, 2.190871, 2.787508, 3.715618)$$

LSP initial bounds: As given in the set of constraints.

Convergence criterion:

$$V_M \leq 1E - 4$$

LSP computational effort:

$$N_f = 602,075 \qquad N_{keep} = 40$$

Problem 5-1-3-0. Source: [36], [58]

$$f(\mathbf{x}) = (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^4 + (x_4 - x_5)^4$$

Subject to:

$$x_1 + x_2^2 + x_3^3 - 3 = 0$$
$$x_2 - x_3^2 + x_4 - 1 = 0$$
$$x_1 x_5 - 1 = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = 0$$
$$\mathbf{x}^* = (1, 1, 1, 1, 1)$$

Computational effort, N_f, cited in literature (GRGA, FMIN): = 623, 472 [58]

Solution using LSP:

$$f(\mathbf{x}^*) = 8.8 * 10^{-6}$$
$$\mathbf{x}^* = (1.0006, 1.0008, 0.9978, 0.9949, 0.9994)$$

LSP initial bounds:

$$0 \leq x_i \leq 13 \qquad \text{for } i = 1, 5$$

convergence criterion:

$$\sigma \leq 1E - 6$$

LSP computational effort:

$$N_f = 1008 \qquad N_{keep} = 50$$

Problem 5-2-0-2. Source: [14]

$$f(\mathbf{x}) = 10x_1^{-.25} x_2 x_4^{-1} x_5^{-.5} + 20x_2^{-1} x_3 x_4^2 x_5^{-.5} + 15x_1^{.5} x_2^{-1} x_3^{-1} x_5^{1.5}$$

Subject to:

$$.01x_1 x_2 + .02x_2^{-2} x_4 x_5 \leq 1$$
$$.04x_1 x_3^{-3} + .02x_2^{-2} x_3 + .02x_4 x_5^{-3} + .06x_1^{-1} x_2 x_4^{-.5} x_5^{.5} \leq 1$$
$$x_i \geq 0 \qquad \text{for } i = 1, 5$$

Solution in literature:

$$f(\mathbf{x}^*) = 37.216$$
$$\mathbf{x}^* = (11.425, 8.747, 0.787, 2.401, 1.179)$$

Solution using LSP:

$$f(\mathbf{x}^*) = 37.226$$

The problem has many global optimal points, of which one is

$$\mathbf{x}^* = (4.6472, 16.3588, 9.3017, 1.7066, 12.5463)$$

LSP initial bounds:

$$0 \leq x_i \leq 20 \qquad \text{for } i = 1, 5$$

Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 40887 \qquad N_{keep} = 50$$

Problem 5-3-0-6. Source: [34], [28]

$$f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1 x_5 + 37.293239x_1 - 40792.141$$

Subject to:

$$0 \leq 85.334407 + 0.0056858x_2 x_5 + 0.0006262x_1 x_4 - 0.0022053x_3 x_5 \leq 92$$
$$90 \leq 80.51249 + 0.0071317x_2 x_5 + 0.0029955x_1 x_2 + 0.0021813x_3^2 \leq 110$$

$$20 \leq 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \leq 25$$
$$78 \leq x_1 \leq 102$$
$$33 \leq x_2 \leq 45$$
$$27 \leq x_3, x_4, x_5 \leq 45$$

Solution in literature:

$$f(\mathbf{x}^*) = -30665.5$$
$$\mathbf{x}^* = (78, 33, 29.995, 45, 36.776)$$

Solution using LSP:

$$f(\mathbf{x}^*) = -30633.80$$
$$\mathbf{x}^* = (78.00016, 33.04861, 30.1344, 44.99984, , 36.57489)$$

LSP initial bounds: As given in the set of constraints

Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 5876 \qquad N_{keep} = 50$$

Problem 5-4-3-0. Source: [58]

$$f(\mathbf{x}) = (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2$$

Subject to:

$$x_1 + 3x_2 = 0$$
$$x_3 + x_4 - 2x_5 = 0$$
$$x_2 - x_5 = 0$$
$$-10 \leq x_i \leq 10 \qquad \text{for } i = 1, 5$$

Computational effort, N_f, cited in literature (GRGA, FMIN): = 82, 2172

Solution in literature:

$$f(\mathbf{x}^*) = 176/43$$
$$\mathbf{x}^* = (-33, 11, 27, -5, 11)/43$$

Solution using LSP:

$$f(\mathbf{x}^*) = 4.0973$$
$$\mathbf{x}^* = (-0.7911, 0.2637, 0.5934, -0.0660, 0.2637)$$

LSP initial bounds:

$$-10 \leq x_i \leq 10 \qquad \text{for } i = 1, 5$$

Convergence criterion:

$$V_M \leq 1E - 3$$

LSP computational effort:

$$N_f = 619 \qquad N_{keep} = 50$$

Problem 6-1-0-2. Transformer Design Source: [5], [56], [58]

$$f(\mathbf{x}) = 0.0204x_1x_4(x_1+x_2+x_3)+0.0187x_2x_3(x_1+1.57x_2+x_4)$$
$$+0.0607x_1x_4x_5^2(x_1+x_2+x_3)+0.0437x_2x_3x_6^2(x_1+1.57x_2+x_4)$$

Subject to:

$$0.001x_1x_2x_3x_4x_5x_6-2.07 \geq 0$$
$$1-0.00062x_1x_4x_5^2(x_1+x_2+x_3)-0.00058x_2x_3x_6^2(x_1+1.57x_2+x_4) \geq 0$$
$$0 \leq x_i \quad \text{for } i=1,6$$
$$5.27 \leq x_1 \leq 5.81$$
$$4.29 \leq x_2 \leq 4.71$$
$$10.14 \leq x_3 \leq 10.56$$
$$11.89 \leq x_4 \leq 12.31$$
$$0.73 \leq x_5 \leq 1.27$$

Bounds are taken from [56].

Solution in literature:

$$f(\mathbf{x}^*) = 135.075961$$
$$\mathbf{x}^* = (5.332666, 4.651744, 10.43299, 12.08230, 0.7526074, 0.87865084)$$

Computational effort, N_f, cited in literature:

(GRGA, FMIN): = 2577, 3419 [58] and 4360355 [56]

Solution using LSP:

The problem has different near optimal points, of which one is

$$f(\mathbf{x}^*) = 135.08276$$
$$\mathbf{x}^* = (5.28766, 4.68762, 10.40192, 12.15321, 0.75406, 0.87688)$$

LSP initial bounds: As given in the set of constraints.

Convergence criterion:

$$V_M \leq 1E-4$$

LSP computational effort:

$$N_f = 272225 \qquad N_{keep} = 60$$

Problem 6-2-0-6. Hesse's Function Source: [66]

$$f(\mathbf{x}) = -25(x_1-2)^2-(x_2-2)^2-(x_3-1)^2-(x_4-4)^2$$
$$-(x_5-1)^2-(x_6-4)^2$$

Subject to:

$$x_1+x_2 \geq 2$$
$$x_1+x_2 \leq 6$$
$$-x_1+x_2 \leq 2$$
$$x_1-3x_2 \leq 2$$
$$(x_3-3)^2+x_4 \geq 4$$
$$(x_5-3)^2+x_6 \geq 4$$
$$x_1, x_2 \geq 0$$
$$1 \leq x_3 \leq 5$$
$$0 \leq x_4 \leq 6$$
$$1 \leq x_5 \leq 5$$
$$0 \leq x_6 \leq 10$$

Solution in literature:

$$f(\mathbf{x}) = -310$$

The problem has 18 local optima one of which is global.

$\mathbf{x}^* = (5, 1, 5, 0, 5, 10)$

Solution using LSP:

$f(\mathbf{x}^*) = -309.997$

$\mathbf{x}^* = (5.00000, 1.00000, 4.99999, 0.00030, 4.99996, 9.99997)$

LSP initial bounds: As given in the set of constraints.

Convergence criterion:

$V_M \leq 1E-5$

LSP computational effort:

$N_f = 14064 \qquad N_{keep} = 60$

Problem 6-3-6-2. Source: [58]

$$f(\mathbf{x}) = x_1 + 2x_2 + 4x_5 + exp(x_1 x_4)$$

Subject to:

$$x_1 + 2x_2 + 5x_5 - 6 = 0$$
$$x_1 + x_2 + x_3 - 3 = 0$$
$$x_4 + x_5 + x_6 - 2 = 0$$
$$x_1 + x_4 - 1 = 0$$
$$x_2 + x_5 - 2 = 0$$
$$x_3 + x_6 - 2 = 0$$
$$0 \leq x_i \qquad \text{for } i = 1, 6$$
$$x_1, x_4 \leq 1$$

Solution in literature:

$f(\mathbf{x}^*) = 19/3$

$\mathbf{x}^* = (0, 4/3, 5/3, 1, 2/3, 1/3)$

Computational effort, N_f, cited in literature (FMIN): = 4767

Solution using LSP:

$f(\mathbf{x}^*) = 6.335$

$\mathbf{x}^* = (.001, 1.334, 1.665, 0.999, 0.666, 0.335)$

LSP initial bounds:

$0 \leq x_1, x_4 \leq 1$

$0 \leq x_2, x_3, x_5, x_6 \leq 2$

Convergence criterion:

$V_M \leq 1E-3$

LSP computational effort:

$N_f = 4966 \qquad N_{keep} = 60$

Problem 6-4-4-0. Source: [34], [58]

$$f(\mathbf{x}) = f_1(x_1) + f_2(x_2))$$

$$f_1(x_1) = \begin{cases} 30x_1 & 0 \leq x_1 < 300 \\ 31x_1 & 300 \leq x_1 < 400 \end{cases}$$

$$f_2(x_2) = \begin{cases} 28x_2 & 0 \leq x_2 < 100 \\ 29x_2 & 100 \leq x_2 < 200 \\ 30x_2 & 200 \leq x_2 < 1000 \end{cases}$$

Subject to:

$$x_1 = 300 - \tfrac{x_3x_4}{131.078}cos(1.48577 - x_6) + \tfrac{0.90798x_4^2}{131.078}cos(1.47588)$$
$$x_2 = -\tfrac{x_3x_4}{131.078}cos(1.48577 + x_6) + \tfrac{0.90798x_4^2}{131.078}cos(1.47588)$$
$$x_5 = -\tfrac{x_3x_4}{131.078}sin(1.48577 + x_6) + \tfrac{0.90798x_4^2}{131.078}sin(1.47588)$$
$$200 - \tfrac{x_3x_4}{131.078}sin(1.48577 - x_6) + \tfrac{0.90798x_4^2}{131.078}sin(1.47588) = 0$$

$0 \leq x_1 \leq 400$
$0 \leq x_2 \leq 1000$
$340 \leq x_3 \leq 420$
$340 \leq x_4 \leq 420$
$-1000 \leq x_5 \leq 1000$
$0 \leq x_6 \leq 0.5236$

Solution in literature:

Results differ in the precision of the $\mathbf{x}$ *vector as a discontinuity in the constraint derivatives forces jump changes in* $f(\mathbf{x})$ *and* $\mathbf{x}$.

High precision: $f(\mathbf{x}^*) = 8927.5888$ [34], [58]
$\mathbf{x}^* = (107.81, 196.32, 373.83, 420.00, 21.31, 0.153)$

Moderate precision: $f(\mathbf{x}^*) = 8853.44$ or 8953.40 [34]
$\mathbf{x}^* = (201.78, 100.00, 383.07, 420.00, -10.907, 0.07314)$

Computational effort, N_f, cited in literature (, GRGA, FMIN): = 1428, 4630 [58]

Solution using LSP:

$f(\mathbf{x}^*) = 8889.881$
$\mathbf{x}^* = (202.996, 100.000, 383.071, 419.999, -10.914, 0.073)$

LSP initial bounds:

As given in the set of constraints

Convergence criterion:

$V_M \leq 1E - 4$

LSP computational effort:

$N_f = 3924 \qquad N_{keep} = 60$

Problem 6-5-3-12. Source: [26] Solution Improved

$f(\mathbf{x}) = x_1^{.6} + x_2^{.6} + x_3^{.4} + 2x_4 + 5x_5 - 4x_3 - x_6$ Subject to:
$x_2 - 3x_1 - 3x_4 = 0$
$x_3 - 2x_2 - 2x_5 = 0$
$4x_4 - x_6 = 0$
$x_1 + 2x_4 \leq 4$
$x_2 + x_5 \leq 4$

$$x_3 + x_6 \leq 6$$
$$x_1 \leq 3$$
$$x_3 \leq 4$$
$$x_5 \leq 2$$
$$x_i \geq 0 \qquad \text{for } i = 1, 6$$

Solution in literature:

$$f(\mathbf{x}^*) = -11.96$$
$$\mathbf{x}^* = (0.67, 2, 4, 0, 0, 0)$$

Solution using LSP:

$$f(\mathbf{x}^*) = -13.402$$
$$\mathbf{x}^* = (0.1667, 2.0000, 4.0000, 0.5000, 0.0000, 2.0000)$$

LSP initial bounds:

$$0 \leq x_1 \leq 3$$
$$0 \leq x_2 \leq 4$$
$$0 \leq x_3 \leq 4$$
$$0 \leq x_4 \leq 2$$
$$0 \leq x_5 \leq 2$$
$$0 \leq x_6 \leq 6$$

Convergence criterion:

$$V_M \leq 1E - 7$$

LSP computational effort: $N_f = 25927$ $N_{keep} = 60$

Problem 6-6-0-4. Source: [8]

$$f(\mathbf{x}) = 10.5x_1 - 3.95x_2 + 3x_3 + 5x_4 + 1.5x_5 - 1.5x_6$$
$$-1.5x_1^2 - x_2^2 - x_3^2 - 2x_4^2 - x_5^2 - 2.5x_6^2$$

Subject to:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 500$$
$$x_1 + 3x_2 + 6x_3 + 2x_4 \geq 50$$
$$3x_5 + 4x_6 \geq 50$$
$$x_3 + 2x_4 + 3x_5 + x_6 \leq 350$$
$$0 \leq x_i \leq 99 \qquad \text{for } i = 1, 2, ..., 6$$

Solution in literature:

$$f(\mathbf{x}^*) = -70262.05$$
$$\mathbf{x}^* = (99, 99, 53, 99, 0, 99)$$

Solution using MINOS 5.1, as reported in the literature

$$f(\mathbf{x}^*) = -69181.04$$
$$\mathbf{x}^* = (99, 99, 99, 76, 0, 99)$$

Solution using LSP:

$$f(\mathbf{x}^*) = -70261.953$$
$$\mathbf{x}^* = (98.99994, 98.99965, 52.99992, 98.99999, 0.00002, 99.00000)$$

LSP initial bounds: As given in the set of constraints.

Convergence criterion:

$$V_M \leq 1E - 2$$

LSP computational effort:

$N_f = 46986$ $N_{keep} = 60$

Problem 7-1-0-24. Source: [13]

$$f(\mathbf{x}) = .7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.5080x_1(x_6^2 + x_7^2)$$
$$+7.4770(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

Subject to:

$$x_1x_2^2x_3 \geq 27$$
$$x_1x_2^2x_3^2 \geq 397.5$$
$$x_2x_3x_6^4/x_4^3 \geq 1.93$$
$$a_1 = ((745 * x_4/x_2/x_3)^2 + 16.91 * 10^6)^{.5}$$
$$a_2 = ((745 * x_5/x_2/x_3)^2 + 157.5 * 10^6)^{.5}$$
$$b_1 = 0.1 * x_6^3$$
$$b_2 = 0.1 * x_7^3$$
$$a_1/b_1 \leq 1100$$
$$a_2/b_2 \leq 850$$
$$x_2x_3 \leq 40$$
$$x_1/x_2 \geq 5$$
$$x_1/x_2^2 \leq 12$$
$$1.5x_6 + 1.9 \leq x_4$$
$$1.1x_7 + 1.9 \leq x_5$$
$$3 \leq x_1 \leq 4$$
$$.6 \leq x_2 \leq .8$$
$$17 \leq x_3 \leq 20$$
$$6.6 \leq x_4 \leq 8.3$$
$$7.3 \leq x_5 \leq 8.3$$
$$2.9 \leq x_6 \leq 3.9$$
$$5 \leq x_7 \leq 5.5$$

Solution in literature:

$$f(\mathbf{x}^*) = 2994.47$$
$$\mathbf{x}^* = (3.5, 0.7, 17, 7.3, 7.71, 3.35, 5.287)$$

Solution using LSP:

$$f(\mathbf{x}^*) = 2994.5$$
$$\mathbf{x}^* = (3.50, 0.70, 17.00, 7.30, 7.716, 3.351, 5.287)$$

LSP initial bounds: As given in the set of constraints.
convergence criterion:

$$V_M \leq 1E - 3$$

LSP Number of functional evaluation:

$$N_f = 82826 \qquad N_{keep} = 70$$

Problem 7-2-2-3. Source: [59]

$$f(\mathbf{x}) = -5x_1 - 5x_2 - 4x_3 - x_1x_3 - 6x_4 - (5x_5)/(1 + x_5)$$
$$-(8x_6)/(1 + x_6) - 10(1 - 2exp(-x_7) + exp(-2x_7))$$

Subject to:

$$10 - \sum_{i=1}^{7} x_i \geq 0$$
$$5 - \sum_{i=1}^{4} x_i \geq 0$$
$$5 - x_1 - x_3 - x_5 - x_6^2 - x_7^2 \geq 0$$
$$2x_4 + x_5 + 0.8x_6 + x_7 - 5 = 0$$

$$x_2^2 + x_3^2 + x_5^2 + x_6^2 - 5 = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = -37.4130$$

$$\mathbf{x}^* = (1.47, 1.98, 0.35, 1.2, 0.57, 0.78, 1.41)$$

Computational effort, N_f, cited in literature (NLPQL): = 437

Solution using LSP:

$$f(\mathbf{x}^*) = -37.352$$

$$\mathbf{x}^* = (1.271, 1.972, 0.536, 1.220, 0.498, 0.758, 1.455)$$

LSP initial bounds:

$$0 \leq x_2, x_3, x_5, x_6, x_7 \leq 2.3$$

$$0 \leq x_1, x_4 \leq 5$$

Convergence criterion:

$$V_M \leq 1E - 3$$

LSP Number of functional evaluation:

$$N_f = 6973724 \qquad N_{keep} = 50$$

Problem 7-3-0-4. Source: [61]

$$f(\mathbf{x}) = (x_1-10)^2+5(x_2-12)^2+(x_3^4+3(x_4-11)^2+10*x_5^6+7x_6^2+x_7^4-4x_6x_7-10x_6-8x_7)$$

Subject to:

$$2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \leq 0$$

$$7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \leq 0$$

$$23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \leq 0$$

$$4x_1^2 + x_2 - 3x_1x_2 + 2x_3^2 5x_6 - 11x_7 \leq 0$$

Solution in literature:

$$f(\mathbf{x}^*) = 680.63005743$$

$$\mathbf{x}^* = (2.330517, 1.9513717, -0.47759196, 4.365728, -0.6245119, 1.0381598, 1.5942702)$$

Computational effort, N_f, cited in literature: = 4368

Solution using LSP:

$$f(\mathbf{x}^*) = 680.6488$$

$$\mathbf{x}^* = (2.33976, 1.94295, -0.44145, 4.38498, -0.63151, 1.04464, 1.60214)$$

LSP initial bounds:

$$-5 \leq x_i \leq 5 \qquad \text{for } i = 1, 7$$

Convergence criterion:

$$V_M \leq 1E - 3$$

LSP Number of functional evaluation:

$$N_f = 68027 \qquad N_{keep} = 70$$

Problem 8-1-0-10. Source: [39]

$$\begin{aligned} f(\mathbf{x}) = -1 * [&0.034x_1x_5 + 0.1225x_3x_7 - 0.280125x_2x_6 \\ &-0.191625x_4x_8 + 50x_7 + 145.6875x_6 + 145.6875x_8 \\ &+50x_5 - 47.52x_5 - 47.52x_6 - 44.3625x_7 - 44.3625x_8] \end{aligned}$$

Subjected to:

$$\begin{aligned} &x_1x_5 + x_2x_6 \geq 2000 + 80x_5 + 80x_6 \\ &x_3x_7 + x_4x_8 \geq 7000 + 65x_7 + 65x_8 \\ &x_1x_5 + x_3x_7 \leq 10000 + 89.286(x_5 + x_7) \\ &x_2x_6 + x_4x_8 \leq 3000 + 166.66(x_7 + x_8) \\ &4x_5 + 4x_6 \leq 1000 \\ &5x_7 + 5x_8 \leq 700 \\ &x_5 + x_6 \leq 233 \\ &x_7 + x_8 \leq 150 \\ &x_5 + x_7 \geq 200 \\ &x_6 + x_8 \geq 100 \\ &48 \leq x_1 \leq 120 \\ &48 \leq x_2 \leq 200 \\ &42 \leq x_3 \leq 120 \\ &42 \leq x_4 \leq 200 \end{aligned}$$

Solution in literature:

$$f(\mathbf{x}^*) = -16592.55$$
$$\mathbf{x}^* = (120, 53.51, 120, 108.87, 122.89, 110.1, 77.1, 62.89)$$

The first two constraints are slightly violated

Solution using LSP:

$$f(\mathbf{x}^*) = -16563.059$$
$$\mathbf{x}^* = (120, 57.41, 119.98, 110.28, 116.06, 116.95, 83.95, 56.06)$$

LSP Initial bounds:

$$\begin{aligned} &48 \leq x_1 \leq 120 \\ &48 \leq x_2 \leq 200 \\ &42 \leq x_3 \leq 120 \\ &42 \leq x_4 \leq 200 \\ &0 \leq x_5 \leq 233 \\ &0 \leq x_6 \leq 233 \\ &0 \leq x_7 \leq 140 \\ &0 \leq x_8 \leq 140 \end{aligned}$$

convergence criterion:

$$V_M \leq 1E - 3$$

Computational effort:

$$N_f = 1426144 \qquad N_{keep} = 80$$

Problem 8-2-0-6. Source: [58]

$$f(\mathbf{x}) = .4x_1^{.67}x_7^{-.67} + .4x_2^{.67}x_8^{-.67} + 10 - x_1 - x_2$$

Subject to:

$$\begin{aligned} &1 - .0588x_5x_7 - .1x_1 \geq 0 \\ &1 - 0.0588x_6x_8 - .1x_1 - .1x_2 \geq 0 \\ &1 - 4x_3x_5^{-1} - 2x_3^{-.71}x_5^{-1} - 0.0588x_3^{-1.3}x_7 - 1 \geq 0 \\ &1 - 4x_4x_6^{-1} - 2x_4^{-.71}x_6^{-1} - 0.0588x_4^{-1.3}x_8 - 1 \geq 0 \end{aligned}$$

$$1 \leq f(\mathbf{x}) \leq 4.2$$
$$0.1 \leq x_i \leq 10 \qquad \text{for } i = 1, 8$$

Solution in literature:

$$f(\mathbf{x}^*) = 3.9511634396$$
$$\mathbf{x}^* = (6.465114, 2.232709, 0.6673975, 0.5957564$$
$$5.932676, 5.527235, 1.013322, 0.4006682)$$

Computational effort, N_f, cited in literature (GRGA, FMIN): = 5929, 9544

Solution using LSP:

$$f(\mathbf{x}^*) = 3.960746$$
$$\mathbf{x}^* = (6.092, 2.550, 0.706, 0.611, 6.016, 5.546, 1.105, 0.416)$$

LSP Initial bounds:

$$0.1 \leq x_i \leq 10 \qquad \text{for } i = 1, 8$$

convergence criterion:

$$V_M \leq 1E - 6$$

LSP Computational effort:

$$N_f = 192453 \qquad N_{keep} = 80$$

Problem 8-3-0-7. Source: [14]

$$f(\mathbf{x}) = 19.4x_1^{-1.47} + 16.8x_2^{-1.66} + 91.5x_3^{-.3} + 19.4x_4^{-1.47}$$
$$+86x_5^{-.38} + 152x_6^{-.27} + 16.8x_7^{-1.66} + 27.4x_8^{-.63}$$

Subjected to:

$$0.537x_1x_2x_3 + 4.47x_4x_5x_6 + 0.386x_7x_8 \leq 1$$
$$.65x_1^{-1} \leq 1$$
$$.15x_1^{-1}x_2^{-1}x_3^{-1} \leq 1$$
$$.65x_4^{-1} \leq 1$$
$$.15x_4^{-1}x_5^{-1} \leq 1$$
$$.60x_7^{-1} \leq 1$$
$$.20x_7^{-1}x_8^{-1} \leq 1$$
$$x_i \geq 0 \qquad \text{for } i = 1, 8$$

Solution in literature:

$$f(\mathbf{x}^*) = 550.642$$
$$\mathbf{x}^* = (0.937, 0.901, 0.676, 0.671, 0.331, 0.467, 1.096, 0.554)$$

Solution using LSP:

$$f(\mathbf{x}^*) = 551.1217$$
$$\mathbf{x}^* = (0.940, 0.954, 0.656, 0.693, 0.326, 0.487, 1.090, 0.581)$$

LSP Initial bounds:

$$0 \leq x_i \leq 3 \qquad \text{for } i = 1, 8$$

LSP convergence criterion:

$$V_M \leq 1E - 3$$

LSP Computational effort:

$$N_f = 5485 \qquad N_{keep} = 80$$

Problem 9-1-0-13. Source: [58]

$$f(\mathbf{x}) = -.5 * (x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$$

Subject to:

$$1 - x_3^2 - x_4^2 \geq 0$$
$$1 - x_5^2 - x_6^2 \geq 0$$
$$1 - (x_1 - x_5)^2 - (x_2 - x_6)^2 \geq 0$$
$$1 - (x_1 - x_7)^2 - (x_2 - x_8)^2 \geq 0$$
$$1 - (x_3 - x_5)^2 - (x_4 - x_8)^2 \geq 0$$
$$1 - (x_3 - x_7)^2 - (x_4 - x_8)^2 \geq 0$$
$$1 - x_7^2 - (x_8 - x_9)^2 \geq 0$$
$$x_3x_9 \geq 0$$
$$x_5x_8 - x_6x_7 \geq 0$$
$$1 - x_9^2 \geq 0$$
$$1 - x_1^2 - (x_2 - x_9)^2 \geq 0$$
$$x_1x_4 - x_2x_3 \geq 0$$
$$x_5x_9 \geq 0$$

Solution in literature:

$$f(\mathbf{x}^*) = -0.8660254038$$

$\mathbf{x}^* = (.884129, .4672425, .03742076, .9992996, .884129, .467242, .03742076, .99929996, 0)$

Computational effort, N_f, cited in literature (GRGA, FMIN):= 16857, 16724

GRGA did not find the global solution

Solution using LSP:

$$f(\mathbf{x}^*) = -0.8657007$$

$\mathbf{x}^* = (.96099, .29952, .23727, .917137, .95973, .28040, .24091, .99339, .02330)$

LSP initial bounds:

$0 \leq x_i \leq 1$ for i= 1 to 9

convergence criterion:

$$V_M \leq 1E-7$$

LSP computational effort:

$N_f = 60500$ $N_{keep} = 80$

Problem 9-2-6-0. Source: [59]

$$f(\mathbf{x}) = \sum_{i=4}^{9} x_i^2$$

Subject to:

$$x_1 + x_2 * e^{(-5x_3)} + x_4 - 127 = 0$$
$$x_1 + x_2 * e^{(-3x_3)} + x_5 - 151 = 0$$
$$x_1 + x_2 * e^{(-x_3)} + x_6 - 379 = 0$$
$$x_1 + x_2 * e^{(x_3)} + x_7 - 421 = 0$$
$$x_1 + x_2 * e^{(3x_3)} + x_8 - 460 = 0$$
$$x_1 + x_2 * e^{(5x_3)} + x_9 - 426 = 0$$

Solution in literature:

$$f(\mathbf{x}^*) = 13390.1$$

$$\mathbf{x}^* = (523.31, -156.95, -0.2, 29.61, -86.62, 47.33, 26.47, 22.92, -39.47)$$

Computational effort, N_f, cited in literature (NLPQL):= 2443

The constraints are slightly violated, possibly due to round off.

Solution using LSP:

$f(\mathbf{x}^*) = 13390.102$

$\mathbf{x}^* = (523.45996094, -157.13957214, -0.19949156, 29.60516357, -86.56939697,$
$47.37318420, 26.26046944, 22.91170883, -39.50439453)$

LSP initial bounds:

$$400 \leq x_1 \leq 600$$
$$-200 \leq x_2 \leq -100$$
$$-10 \leq x_3 \leq 0$$
$$0 \leq x_4 \leq 600000$$
$$-10000 \leq x_5 \leq 10000$$
$$0 \leq x_6 \leq 800$$
$$-10 \leq x_7 \leq 400$$
$$-10 \leq x_8 \leq 400$$
$$-600 \leq x_9 \leq 10$$

Convergence criterion:

$$V_M \leq 1E-3$$

LSP computational effort (number of functional evaluation):

$$N_f = 16601 \qquad N_{keep} = 90$$

Problem 15-1-0-29. Source: [58]

$$f(\mathbf{x}) = \sum_{k=0}^{4}(2.3x_{3k+1} + 0.0001x_{3k+1}^2 + 1.7x_{3k+2} + 0.0001x_{3k+2}^2 + 2.2x_{3k+3} + 0.00015x_{3k+3}^2)$$

Subject to:

$$0 \le x_{3j+1} - x_{3j-2} + 7 \le 13$$
$$0 \le x_{3j+2} - x_{3j-1} + 7 \le 14$$
$$0 \le x_{3j+3} - x_{3j} + 7 \le 13$$
$$\text{for } j = 1,4$$
$$x_1 + x_2 + x_3 - 60 \ge 0$$
$$x_4 + x_5 + x_6 - 50 \ge 0$$
$$x_7 + x_8 + x_9 - 70 \ge 0$$
$$x_{10} + x_{11} + x_{12} - 85 \ge 0$$
$$x_{13} + x_{14} + x15 - 100 \ge 0$$
$$8 < x_1 \le 21$$
$$43 \le x_2 \le 57$$
$$3 \le x_3 \le 16$$
$$0 \le x_{3k+1} \le 90$$
$$0 \le x_{3k+2} \le 120$$
$$0 \le x_{3k+3} \le 60$$
$$\text{for } k = 1,4$$

Solution in literature:

$$f(\mathbf{x}^*) = 664.8204500$$
$$\mathbf{x}^* = (8,49,3,1,56,0,1,63,6,3,70,12,5,77,18)$$

Computational effort, N_f, cited in literature (GRGA): = 3857

Solution using LSP:

$$f(\mathbf{x}^*) = 668.921$$
$$\mathbf{x}^* = (8.113, 48.861, 3.031, 2.081, 48,934, 0.070, 8.067, 55.933, 6.002, 13.793, 62.804, 8.405, 16.043, 69.665, 14.305)$$

LSP initial bounds: As given in the set of constraints.

convergence criterion:

$$V_M \le 1E-3$$

LSP computational effort:

$$N_f = 3805238 \qquad N_{keep} = 120$$

Problem 20-1-1-0. Source: [59] Solution improved

$$f(\mathbf{x}) = \sum_{i=1}^{20} i(x_i^2 + x_i^4)$$

Subject to:

$$\sum_{i=1}^{20} x_i^2 = 1$$

Solution in literature:
$f(\mathbf{x}^*) = 1.91667$
$\mathbf{x}^* = (0.91287, 0.408286, -0.00017, -0.0000054, 0.0000, -0.0000089,)$
$0.0000082, -0.000014, 0.000022, -0.0, 0.0000135, -0.000004,$
$-0.000011, -0.000013, 0.0, 0.00002, 0.00000546, -0.000009, -0.00001, 0.0)$
Computational effort, N_f, cited in literature (NLPQL): = 198
Solution using LSP:
$f(\mathbf{x}^*) = 1.91836$
$\mathbf{x}^* = (-0.9112, -0.4116, -0.0101, -0.0087, 0.0077, 0.0021, -0.0050,$
$-0.0016, 0.0051, 0.0008, -0.0013, 0.0047, -0.0034, 0.0010,$
$0.0013, 0.0004, -0.0041, -0.0024, -0.0036, 0.0005)$
LSP initial bounds:
$-1 \leq x_i \leq 1$ for $i = 1, 20$
Convergence criterion:
$V_M \leq 1E-6$
LSP computational effort:
$N_f = 767986$ $N_{keep} = 120$

The LSP solution is inferior by 0.088% and different from the one given in the literature, but multiple local optima were observed using LSP. All ordinates with negative and positive sign give optimal solution. There are 2^{20} possible solution points if the initial bounds are $-1 \leq x_i \leq 1$ for $i = 1, 20$.

Appendix B

DIRECT SEARCH METHODS

To give a wider perspective of direct search methods some of the most common methods which appear in the literature are described below. The descriptions are presented verbatim from the primary sources cited.

Blind search [Leon, 1966]: The most elementary type of sampling procedure is one in which trial points are generated randomly within the feasible region. This search method simply selects a starting feasible point $\mathbf{x}^0$, evaluates $f(\mathbf{x})$ at $\mathbf{x}^0$ and then randomly selects another feasible point $\mathbf{x}^1$, and evaluates $f(\mathbf{x})$ at $\mathbf{x}^1$. In effect, both the search direction and the step length are chosen simultaneously. The current point has to provide a better value of the objective function to be retained, otherwise it is discarded. Such a process continues until a specified number of points have been tested or a specified computational effort expended.

A slightly more efficient approach is to divide the sampling range into a set of subareas known as blocks. Initially the search is carried out in each block separately. The block providing the best values is used to initiate the next phase of the search which is executed within progressively smaller blocks.

Grid search : A series of points are evaluated about a reference point selected according to some type of design. Then move to that point which improves the objective function the most, and repeat. If each length of the search domain is divided into b

sections, there will be b^n grid points for an n dimensional problem. Therefore, this method function evaluations at $b^n - 1$ points in addition to the reference point.

For $n = 10$, we must examine $3^{10} - 1 = 59,048$ values of $f(\mathbf{x})$ if a three-level factorial design is to be used, which is a prohibitive number of function evaluations [Edgar & Himmelblau, 1988] for an equivalent $N_{keep} = 3$ and a single iteration.

Univariate search [Edgar & Himmelblau, 1988]: Select n fixed search directions (usually the coordinate axes) for an objective function of n variables. Then $f(\mathbf{x})$ is minimized in each direction sequentially using a one dimensional search. While this method is effective for a quadratic function of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} c_i x_i^2$$

because the search direction lines up with the principal axes, it does not perform satisfactorily for the more general quadratic objective function of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j$$

where c_i and d_{ij} are coefficients.

Conjugate direction methods [Edgar & Himmelblau, 1988]: Conjugate direction methods are presented as methods for optimizing strictly convex quadratic functions. Experience has shown that conjugate directions are more effective as search directions than arbitrarily chosen search directions or even orthogonal search directions. Two directions S^i and S^j are said to be conjugate (or conjugated) with respect to each other if

$$(S^i)^T Q (S^j) = 0$$

where, the matrix Q is the Hessian matrix of the objective function, H.

In general, at a set of n linearly independent directions of search $S^0, S^1, ...s^{n-1}$ are said to be conjugate with respect to a positive definite square matrix Q if

$$(S^i)^T Q(S^j) = 0; 0 \leq i \neq j \leq n-1$$

The conjugate directions exist only for a quadratic approximation of the function at a single stage k. Once the objective function is modeled by a new approximation at stage $(k+1)$, the directions on stage k are unlikely to be conjugate to any of the directions selected on stage $(k+1)$.

Powell's conjugate direction methods [Reklaitis et al., 1983]: This algorithm uses the history of the iterations to build up directions for acceleration and at the same time avoids degenerating to a sequence of coordinate searches. It is based upon the model of a quadratic function.

The motivation for the algorithm stems from the observation that if a quadratic function in n variables can be transformed so that it is just the sum of perfect squares, then the optimum can be found after exactly n single variable searches, one with each of the transformed variables.

Simplex method [Edgar & Himmelblau, 1988]: It uses a rectangular geometric figure (a simplex) to select points at the vertices of the simplex at which to evaluate $f(\mathbf{x})$.

The simplex search method is based upon the observation that the first-order experimental design requiring the smallest number of points is the regular simplex. In n dimensions, a regular simplex is a polyhedron composed of $n+1$ equidistant points, which form its vertices.

The method begins by setting up a regular simplex in the space of the independent variables as

$$x_i = b_i^L + \xi_i(b_i^U - b_i^L)$$

where ξ_i is a random deviate uniformly distributed over the interval $[0, 1]$, b_i^L and b_i^U are the lower and upper bounds respectively for each variable i.

The objective function is evaluated at each vertex. The vertex with highest function value (for a minimization problem) is located and labelled as the "worst" vertex. Before throwing away the "worst" vertex, it is reflected through the centroid to generate a new trial point, which is used to complete the next simplex. As long as the objective function decreases smoothly, the iterations move along crabwise until either the minimum is straddled or the iterations begin to cycle between two or more simplexes. Then, the simplex sides are reduced to advance the search until a prescribed side length is maintained or until the standard deviation of the function values at the vertices gets smaller than a prescribed value.

Complex method [Box, 1965]: The constrained simplex (Complex) method searches for the minimum value of a function $f(\mathbf{x})$ subject to m constraints.

The simplex direct search is based on the generation and maintenance of a pattern of search points and the use of projections of undesirable points through the centroid of the remaining points as the means of finding new trial points. In the presence of inequality constraints, it is evident that if the new point is infeasible, then it is a simple matter to retract it toward the centroid until it becomes feasible.

It is assumed that an initial point $\mathbf{x}^0$, which satisfies all constraints is available. It is assumed that the feasible region is convex.

For an n variable problem, $k \geq n+1$ points are used, of which one is the given point. The further $(k-1)$ points required to set up the initial configuration are obtained one at a time by the use of random numbers and ranges for each of the independent variables.

The function is evaluated at each vertex, and the vertex of highest function value (for a minimization problem) is replaced by a point $\alpha \geq 1$ times as far from the centroid of the remaining points as the reflection of the worst point in the centroid, the new point being collinear with the rejected point and the centroid of the retained vertices. If the trail point is also the worst, it is moved halfway towards the centroid of the remaining points to give a new trial point.

Typical values of k and α are $2 * n$ and 1.3 respectively.

If a trial vertex does not satisfy some constraint on some independent variable x_i, that variable is reset to a value 0.000001 inside the appropriate limit.

Different runs are required to identify the global solution. Therefore, this method too is a local search.

The Hooke-Jeeves pattern search [Reklaitis et al., 1983]: This method is a combination of "exploratory" moves of the one variable at a time kind with "pattern" or acceleration moves regulated by some heuristic rules [Reklaitis et al., 1983]. The exploratory moves examine the local behaviour of the function and seek to locate the direction of any sloping valleys that might be present. The pattern moves utilize the information generated in the exploration to step rapidly along the valleys. A pattern move consists of a single step from the present base point along the line from the previous to the current base point. Thus, a new pattern point, $\mathbf{x}_p$, is calculated as

$$\mathbf{x}_p^{(k+1)} = \mathbf{x}^{(k)} + (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$$

where

$\mathbf{x}^{(k)}$ is the current base point

$\mathbf{x}^{(k-1)}$ is the previous base point

$\mathbf{x}_p^{(k+1)}$ is the pattern move point

$\mathbf{x}^{(k+1)}$ is the next (new) base point

Recognizing that this move may not result in an improvement, the point $\mathbf{x}_p^{(k+1)}$ is accepted only temporarily. It becomes the temporary base point for a new exploratory move. If the result of this exploratory move is a better point than the previous base point $\mathbf{x}^{(k)}$, then this point is accepted as the new base point $\mathbf{x}^{(k+1)}$. On the other hand, if the exploratory move does not produce improvement, then the pattern move is discarded and the search returns to $\mathbf{x}^{(k)}$, where an exploratory search is undertaken to find a new pattern.

Because of its reliance on coordinate steps, the algorithm can, however, terminate prematurely, and in the presence of severe non-linearities will generate to a sequence of exploratory moves without benefit of pattern acceleration.

Simulated annealing [Corana et al., 1987]: Simulated annealing means simulating the annealing process by a Monte Carlo method (random changes in the state of the system,), where the global minimum of the objective function represents the low energy configuration [Torn & Zilinskas, 1988]. The method discriminates between "gross behaviour" of the objective function and finer "wrinkles", where the gross behaviour directs the search to the region where the global optimum should be present. Then a finer local search is performed within the neighbourhood of the presumed global optimum. In any case this method does not guarantee finding the global optimum. For this reason, it is referred to as a suitably perturbed local search procedure [Kan et al., 1989].

The method proceeds iteratively starting from a given point and generating new candidates around the current point applying random moves along each coordinate direction, one after the other. A candidate point is accepted if it produces an objective function value less than that of the previous point, or if greater accepted with probability P. This probability is given by $P = exp(-\Delta f/T)$ and is a function of the ratio of the increase in objective function $\Delta f = f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)$ and the temperature T. The temperature T starts with a large value and reduces progressively until no more useful objective function improvement is expected. An important characteristic of this random procedure is that the next point accepted may have higher objective value than the previous one [Torn & Zilinskas, 1988].

Experimental results on test problems showed that a large number of function evaluations is required to obtain solutions with the simulated annealing method [Aluffi-Pentini et al., 1985]. The method has also been found very inefficient when compared with the simplex method. On the average simulated annealing takes 500-1000 times more function evaluations than the simplex method.