LINEARLY CONSTRAINED NONLINEAR PROGRAMMING: A CONJUGATE DIRECTIONS APPROACH

A THESIS

Presented to

The Faculty of the Division of Graduate Studies

by

Abdelaziz Bouzaher

In Partial Fulfillment

of the Requirement for the Degree

Doctor of Philosophy

in the School of Industrial and Systems Engineering

Georgia Institute of Technology

October, 1983

LINEARLY CONSTRAINED NONLINEAR PROGRAMMING:

A CONJUGATE DIRECTIONS APPROACH

Approved:

Dr. Mokhtar S. Bazaraa, Chairman

Dr. David E. Fyffe

Dr. Jamie J. Goode

Dr. Ronald L. Rardin

Dr. C. M. Shetty

Date approved by Chairman 10/21/83

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Mokhtar S. Bazaraa for the guidance and help he has generously provided during my years as a graduate student, sometimes under difficult conditions.

I am also grateful to Dr. David E. Fyffe, Dr. Jamie J. Goode, Dr. Ronald R. Rardin, and Dr. C. M. Shetty for serving on my dissertation committee. A special thanks to Dr. Jamie J. Goode for his availability and numerous insightful comments.

I am also grateful to the Algerian Ministry of High Education for providing me with the opportunity and support to attend graduate school. In addition, I am grateful to the School of Industrial and Systems Engineering, and especially Dr. Michael E. Thomas, the Director, for providing financial assistance.

Finally, I would like to thank Miss Nancy Houghton for her patience and excellent typing job. Most of all, I would like to thank my parents for their encouragement and support over the years.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS iv

LIST OF TABLES v

LIST OF ILLUSTRATIONS vi

SUMMARY vii

CHAPTER

I. INTRODUCTION 1
   1. Introduction
   2. Problem Statement
   3. Importance of the Problem
   4. Scope of Research

II. LITERATURE REVIEW 24
   1. Introduction
   2. Projection Methods
   3. Methods of Feasible Directions
   4. Methods of Conjugate Directions
   5. Curvilinear Methods
   6. State of the Existing Software

III. A CONJUGATE DIRECTIONS ALGORITHM 89
   1. Introduction
   2. The Proposed Conjugate Directions Algorithm
   3. The Line Search Scheme
   4. The Direction Finding Problem

IV. GLOBAL CONVERGENCE ANALYSIS 128
   1. Introduction
   2. Global Convergence Analysis
      2.1 Properties of the Direction Vector
      2.2 Uniform Boundedness of the Step-Size
      2.3 Convergence of the Subsequence of Restarting Points
      2.4 Convergence of the Sequence Generated by the Algorithm

V. LOCAL CONVERGENCE ANALYSIS 157
   1. Introduction
   2. Stabilization of the Set of Binding Constraints
   3. Properties of the Initial Step-Size Approximation
   4. Verification of the Approximate Conjugacy of Directions
   5. Superlinear Convergence

Page

VI. COMPUTATIONAL RESULTS 207
1. Introduction
2. Coding and Implementation
3. Numerical Results

VII. CONCLUSIONS AND EXTENSIONS 239

APPENDIX A: TEST PROBLEMS 241

BIBLIOGRAPHY 266

VITA 277

LIST OF TABLES

Table		Page
2-1	A Summary of Existing Software	88
3-1	Lemke's Arrays and Corresponding Direction Vectors	127
6-1	A Summary of Test Problems	208
6-2	Solution Summary for 2-Variable Problems	218
6-3	Solution Summary for 3-Variable Problems	219
6-4	Solution Summary for 4-Variable Problems	220
6-5	Solution Summary for 5-Variable Problems	221
6-6	Solution Summary for Problem 21	222
6-7	Solution Summary for Problem 22	222
6-8	Solution Summary for 10-Variable Problems	223
6-9	Solution Summary for Problem 25	224
6-10	Solution Summary for Problem 26	225
6-11	Solution Summary for Problem 27	226
6-12	Solution Summary for Problem 28	227
6-13	Solution Summary for Problem 29	228
6-14	Solution Summary for Problem 30	229
6-15	Average Results Per Number of Variables	230
6-16	Results Showing Sensitivity to α_k and Type of Restarting for Problem 17	232
6-17	Comparative Results: Function and Gradient Evaluations	236
6-18	Comparative Results: Number of Iterations	237
6-19	Comparative Results: Execution Times	238

LIST OF ILLUSTRATIONS

Figure		Page
1-1	Example of a Water Distribution Network	13
3-1	Graphical Illustration of the Proposed Method	103
3-2	A Flowchart of the Proposed Method	107
3-3	Step-Size Determination	112

SUMMARY

The primary purpose of this study is the development of a method to solve the general linearly constrained nonlinear program using conjugate directions. The theoretical results show that the proposed algorithm converges globally and at the same time exhibits a superlinear rate of convergence.

A general outline of the algorithm is as follows. At each iteration a projection problem, which is a strictly convex quadratic program, is solved. This problem is designed to project an unconstrained descent step onto the feasible region in such a way as to produce a feasible direction of descent which is conjugate to previously constructed conjugate directions. This conjugacy property is aimed at producing a fast local rate of convergence of the algorithm. An inexact line search based on the properties of the projection problem is undertaken. This line search produces a step-size with a finite number of functional evaluations automatically. In addition, an initial approximation of the step-size is used close to a solution point. This initial step-size is based on the local quadratic approximation of the objective function. It also has the property of closely approximating the exact step-size along the conjugate directions. This results in the initial approximation being used without further trials. The method will either produce a Kuhn-Tucker point of the original problem or reinitializes a projection operator containing conjugate directional information.

The theoretical research involves the study of global and local

convergence properties of the algorithm. Global convergence is established through the use of the inexact line search. Both the subsequence of restarting points and the sequence of all points generated by the algorithm are studied. The local converges analysis establishes a superlinear rate of convergence of the algorithm by showing that eventually the set of binding constraints will not change, the initial step-size approximates the exact one, and the directions of move constructed satisfy the approximate conjugacy property.

Furthermore, a comprehensive discussion of the literature is provided.

Finally, a set of 30 problems was used to test an implementation of the algorithm on a CDC Cyber 70 Model 74-28 CDC 6400 machine in time-sharing mode.

CHAPTER I

INTRODUCTION

1. Introduction

Mathematical programming in general, and nonlinear programming in particular, have attracted the attention of scientists from a variety of fields for many years. The advances in computing capabilities has made it increasingly possible to solve larger and more complex problems. This, in turn, stimulated a lot more research, not only in developing new and more sophisticated methods, but also in enlarging the scope of applications to such areas as, optimal control, nonlinear networks, economic planning, water resources, and chemical processing, to mention only a few.

A linearly constrained nonlinear program is a mathematical model for optimizing an objective function in the presence of inequality and/or equality constraints. Since the constraints are linear, the only nonlinearities arise in the objective function. This makes this class of problems a subset of the general nonlinear programs where nonlinearities may be present in the objective function and the constraints.

2. Problem Statement

A general linearly constained nonlinear program is of the form:

$$\text{Minimize} \quad f(x)$$

$$\text{Subject to} \quad x \in S$$

where $f\colon E^n \to E^1$ is a continuously differentiable function, $x \in E^n$ is a decision vector, and $S \subset E^n$ is a set of linear restrictions comprising equalities and/or inequalities and possibly bound restrictions. This leads to the most general form:

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{Subject to} & \left.\begin{array}{l} Bx \leq b \\ Dx = d \\ x \geq \ell \\ x \leq u \end{array}\right\} \Longleftrightarrow x \in S \end{array}$$

where B is an $(m_1 \times n)$ dimensional array, $b \in E^{m_1}$, D is an $(m_2 \times n)$ dimensional array, $d \in E^{m_2}$, $\ell \in E^n$, and $u \in E^n$.

In this study we will work with the general inequality constrained problem of the form:

$$\begin{array}{ll} \text{P:} \quad \text{Minimize} & f(x) \\ \quad\quad \text{Subject to} & Ax \leq b \end{array}$$

where A is an $(m \times n)$ dimensional array, $b \in E^m$ and $x \in E^n$. Bounds will be considered as regular constraints, and equality constraints will be handled though some minor changes in implementation. The feasible region is represented by the set:

$$S = \{x \mid Ax \leq b\}$$

S is essentially defined by the intersection of m half-spaces defined by the m hyperplanes: $a_i^t x = b_i$; i=1,...,m. A vector $x \in E^n$ is called a feasible point if $x \in S$. That is, if x satisfies all the restrictions defined by the feasible region. Otherwise, if $x \notin S$ it is an infeasible point.

A feasible point is called a global solution if the objective function value at that point is lowest among the set of all feasible points. In other words, $\bar{x}$ is a global solution if:

$$f(\bar{x}) \leq f(x), \ \forall \ x \in S$$

On the other hand, a feasible point is called a local solution if its objective function value is lowest in a neighborhood around it, that is $\bar{x}$ is a local solution if:

$$f(\bar{x}) \leq f(x), \ \forall \ x \in S \cap \{y \mid \|y-\bar{x}\| < \varepsilon, \ \varepsilon > 0\}$$

We note that a global solution is a local one. Under certain assumptions, a solution to problem (P) satisfies some optimality conditions such as discussed by Gill and Murray (1974a) and McCormick (1970b).

We say that $\bar{x}$ is a Kuhn-Tucker Point (KTP) if $\bar{x}$ satisfies the well-known Kuhn-Tucker (1951) conditions which are necessary for $\bar{x}$ to be a local minimum to problem (P). These conditions, also known as the First Order Necessary Conditions are as follows: There exists a vector $\bar{u} \in E^m$, known as the vector of Kuhn-Tucker Multipliers such that:

$$\nabla f(\bar{x}) + \sum_{i=1}^{m} a_i^t \bar{u}_i = 0$$

$$a_i^t \bar{x} \leq b_i, \ i=1,\ldots,m$$

$$\bar{u}_i(a_i^t \bar{x} - b_i) = 0; \ i=1,\ldots,m$$

$$\bar{u}_i \geq 0; \ i=1,\ldots,m.$$

A point $\bar{x}$ satisfying these conditions will be called a First Order Kuhn-Tucker point.

The above conditions involve only first derivatives of the objective function $\nabla f(x)$. Additional information is obtained when second derivative information is used. Details of such conditions for general problems can be found in Fiacco and McCormick (1968). These conditions are summarized here for problem (P):

i) <u>Second Order Necessary Conditions</u>

If $f(x)$ is twice continuously differentiable, $\bar{x}$ is a local minimum and, the linear independence of the gradients of the binding constraints at $\bar{x}$ is satisfied then:

a) $(\bar{x},\bar{u})$ satisfies the first order Kuhn-Tucker conditions

b) $y^t G(\bar{x})\, y \geq 0$, for all $y \in E^n$ satisfying:

$$a_i^t y = 0; \ i \in I(\bar{x}) = \{i \mid a_i^t \bar{x} = b_i\}$$

where $G(\bar{x})$ is the matrix of second derivatives of $f(x)$ evaluated at $\bar{x}$.

ii) Second-Order Sufficiency Conditions

If f(x) is twice continuously differentiable and, associated with a point $\bar{x}$ is a vector $\bar{u}$ such that:

a) $(\bar{x},\bar{u})$ is a Kuhn-Tucker pair

b) For that $\bar{u}$, $y^t G(\bar{x}) y > 0$

For all $y \neq 0$ such that $a_i^t y = 0$, $i \in I(\bar{x}) \cap I(\bar{u})$

where $I(\bar{u}) = \{i \mid \bar{u}_i > 0\}$

Then $\bar{x}$ is an isolated local minimum for problem (P).

A linearly constrained algorithm aims at iteratively finding a local or global solution to problem (P). This research is concerned with developing such an algorithm.

3. Importance of the Problem

In this section we will highlight the main reasons for which the linearly constrained nonlinear program has received considerable attention over the past decade.

In recent years several methods have been proposed in which a general nonlinear programming problem is solved as a sequence of linearly constrained problems. This will be the subject of the first part of this section. Perhaps also as significant is the fact that linearly constrained problems form a class of their own in that many real world problems are formulated as such; we will discuss this application aspect in the second part of the section.

3.1 The Linearly Constrained Problem as a Subproblem to a More General Procedure

For the general nonlinear programming problem:

$$\text{(NLP):}\quad \text{Minimize} \quad f(x)$$
$$\text{Subject to} \quad g_i(x) \leq 0;\ i=1,\ldots,m$$
$$h_i(x) = 0;\ i=1,\ldots,\ell$$

Several methods of solution have been proposed which are based on the linearization of the constraint set. A representative selection of the important work in this area is: Robinson (1972), Rosen and Kreuser (1972), Gruver and Engersback (1976), Rosen (1977), and Van Der Hoek (1980).

The linearized problem is defined in general form as:

$$\text{(LNLP):}\quad \text{Minimize} \quad f(x) + \phi(x_k, x)$$
$$\text{Subject to} \quad g_i(x_k) + \nabla g_i(x_k)^t (x - x_k) \leq 0;\ i=1,\ldots,m$$
$$h_i(x_k) + \nabla h_i(x_k)^t (x - x_k) = 0;\ i=1,\ldots,\ell$$

where, following Van Der Hoek (1980), $\phi(x_k, x)$ is a correction term designed to offset, by means of a corrected objective function, any possible poor behavior of the algorithm caused by the local linearization. $\phi(x_k, x)$ will generally depend on $g_i(x)$ and $h_i(x)$ and/or their linearized forms. Different versions of linearly constrained programs arise from different choices of $\phi(x_k, x)$. For instance, Rosen and Kreuser (1972) propose a choice leading to the following linearly constrained problem at x_k:

$$\text{Minimize} \quad f(x) + \sum_{i=1}^{m} u_i(x_k)\, g_i(x) + \sum_{i=1}^{\ell} v_i(x_k)\, h_i(x)$$

$$\text{Subject to} \quad g_i(x_k) + \nabla g_i(x_k)^t (x - x_k) \leq 0; \; i=1,\ldots,m$$

$$h_i(x_k) + \nabla h_i(x_k)^t (x - x_k) = 0; \; i=1,\ldots,\ell$$

where $u_i(x_k)$ and $v_i(x_k)$ are the current Lagrange Multiplier estimates for $i=1,\ldots,m$. Clearly, the objective function is the Lagrangian function with fixed multipliers.

Robinson (1972) uses another form of $\phi(x_k,x)$, leading to the linearly constrained problem:

$$\text{Minimize} \quad f(x) + \sum_{i=1}^{m} u_i(x_k)[g_i(x) - g_i(x_k) - \nabla g_i(x_k)^t (x - x_k)]$$

$$+ \sum_{i=1}^{\ell} v_i(x_k)[h_i(x) - h_i(x_k) - \nabla h_i(x_k)^t (x - x_k)]$$

$$\text{Subject to} \quad g_i(x_k) + \nabla g_i(x_k)^t (x - x_k) \leq 0; \; i=1,\ldots,m$$

$$h_i(x_k) + \nabla h_i(x_k)^t (x - x_k) = 0; \; i=1,\ldots,\ell$$

It is seen here that the linear approximations to the original contraints are subtracted from the Lagrangian function, as compared to the previous approach.

If the original problem (NLP) includes both linear and nonlinear constraints, Van DerHoek (1980), uses yet another form for $\phi(x_k,x)$ where only the currently active constraints are linearized, which leads to the following linearly constrained problem:

$$\text{Minimize} \quad f(x) + \sum_{i \in I(x_k)} u_i(x_k)[g_i(x) - g_i(x_k) - \nabla g_i(x_k)^t(x - x_k)]$$

$$+ \sum_{i=1}^{\ell} v_i(x_k)[h_i(x) - h_i(x_k) - \nabla h_i(x_k)^t(x - x_k)]$$

$$\text{Subject to:} \quad g_i(x_k) + \nabla g_i(x_k)^t(x - x_k) \leq 0; \quad i=1,\ldots,m$$

$$h_i(x_k) + \nabla h_i(x_k)^t(x - x_k) = 0; \quad i=1,\ldots,\ell$$

$$a_i^t x \leq b_i; \qquad i=1,\ldots,p$$

where $I(x_k)$ is the set of currently active contraints.

A final note about $\phi(x_k,x)$ is that, by looking at the Kuhn-Tucker Conditions for optimality of $\bar{x} = x_k$ for both problems (NLP) and (LNLP) defined above, certain properties which $\phi(x_k,x)$ needs to have will emerge. And these are essentially:

$$\phi(x_k,x_k) = 0 \quad \text{and} \quad \nabla_x \phi(x_k - x)\ (x_k) = 0$$

Once the linearized version has been determined, a procedure for solving the general problem will reduce to solving a sequence of linearly constrained subproblems as follows:

Initialization: Initialize variables, let $k = 0$.

Main Step: Given x_k, find a first order Kuhn-Tucker point of the linearly constrained problem (LNLP)

Stopping Step: If convergence tests are satisfied, Stop; Otherwise, set $k = k+1$, and return to the main step.

Finally, we note that the linearly constrained program also arises in another area of optimization: geometric programming. For illustration purposes, we give here the formulation of both the primal and dual "classical geometric program" (Duffin, Peterson, and Zener (1967)).

Primal:

$$\text{Minimize} \quad g_0(x)$$

$$\text{Subject to} \quad g_j(x) \leq 1; \; j=1,\ldots,p$$

$$x_j > 0; \; j=1,\ldots,m$$

where:

$$g_k(x) = \sum_{i \in J(k)} c_i x_1^{a_{i1}} x_2^{a_{i2}} \cdots x_m^{a_{im}}; \; k=0,1,\ldots,p$$

$$J(k) = \{m_k, m_{k+1}, \ldots, n_k\}; \; k=0,1,2,\ldots,p$$

$$m_0 = 1, \; m_1 = m_{0+1}, \ldots, \; m_p = n_{p-1}, \; n_p = n$$

The exponents a_{ij} are arbitrary; c_i are positive. The $g_k(x)$ functions are called posynomials.

Dual (Linearly Constrained):

$$\text{Maximize} \left\{ \prod_{i=1}^{n} \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \right\} \prod_{k=1}^{p} \lambda_k(\delta)^{\lambda_k(\delta)}$$

$$\text{Subject to} \sum_{i \epsilon J(0)} \delta_i = 1$$

$$\sum_{i=1}^{n} a_{ij} \delta_i = 0; \; j=1,2,\ldots,m$$

$$\delta_i > 0; \; i=1,\ldots,n$$

where $\lambda_k(\delta) = \sum_{i \epsilon J(k)} \delta_i; \; k=1,\ldots,p$

and $J(k)$, c_i are as in the primal problem.

It emerges from the above considerations that each of the subproblems should be solved as efficiently as possible. In addition, the method should be guaranteed to converge, to insure that the outer iterations of the general procedure will converge. The method we propose in this study to solve the linearly constrained problem will have these properties.

3.2 Applications

Nonlinearities in modeling real world situations arise in many contexts as illustrated by the wide scope of applications in a recent publication edited by Geoffin and Rousseau (1982). In engineering, nonlinearities arise in a natural way as systems configurations and cost functions depend nonlinearly on design variables. See for example Avriel and Dembo (1979). But in this section we will limit ourselves to a brief description of the types of applications that give rise to linearly constrained nonlinear programming problems, grouping them in different categories like: nonlinear networks, production and distribution, economic planning, and oil and chemical industry.

Our aim is not to be exhaustive, but rather to illustrate the kinds of applications relating to this research. A more comprehensive survey of application in the general area of nonlinear programming can be found in Lasdon and Warren (1980), in addition to the previously cited studies.

Nonlinear Networks: In this area a vast amount of work has been done for a variety of specific problems. We choose to discuss four important classes of applications: water distribution systems, electrical networks, long term power generation and expansion, and multicommodity network flow problems.

Water distribution systems are designed to deliver water from sources to consumers through pipeline networks equipped with valves, pumps, reservoirs and other components. The underlying mathematical structure is represented by a network. Construction of optimization models can be quite complex if the major aspects of planning, design and operation of the system are considered together. Then, decisions have to be made about such things as, pipe layout and diameter, pump locations and characteristics, valve locations, reservoir locations and sizes, pumps and valves to be operated under different loading conditions; for operations over time: times at which the controlled components are switched on and off, control of reservoir levels, etc. The system's constraints include such aspects as, satisfaction of physical laws of flow in the network, satisfaction of demand from different types of consumers (domestic, irrigation, industry), bounds on pressure in the pipes.

A subclass of these problems, reducing the decision variables to optimal pipe diameters only, have been formulated as linearly constrained problems where the objective function is separable. They have been applied to real world systems, such as New York's primary distribution system (De Neufville (1971)). The reader can find more details about this class of problems in Avriel and Dembo (1979).

Another subclass, called "pipe network" problems, is concerned with finding a set of flows and pressures in a water distribution network when supply and demand in the system are known. The nonlinearities arise from a set of stationary point conditions governing the flows and pressures in the network. That is, the pressure decrease from one end of the pipe to the other is a nonlinear function of the flow per unit time. Until recently, this problem was solved mainly by special techniques for solving systems of equations representing the stationary point conditions. The Newton-Raphson method of Donachie (1973) is one such approach. For the interested readers, a comprehensive study of these methods is the one by Jeppson (1975). But mathematical programmers have started to address this problem and other related ones. An important study in this regard was done by Collins, Helgason, Kennington and Leblanc (1978), who formulated the pipe network problem as a general convex linearly constrained problem. We give here a summarized version of this model. The distribution problem is formulated over a directed network (N,E) where N is the set of nodes and E the set of arcs. Letting x_{ij} denote the flow from node i to node j for all $(i,j) \in E$, for each $n \in N$, y_n denotes the pressure as measured by "hydraulic head" at node n. If the node is a reservoir, the head at the node is fixed (see figure below). For all $n \in N$, a_n denotes the flow requirements at node n: $a_i > 0$ for a supply node and $a_i < 0$ for a demand node. A typical such network is illustrated in Figure 1-1.

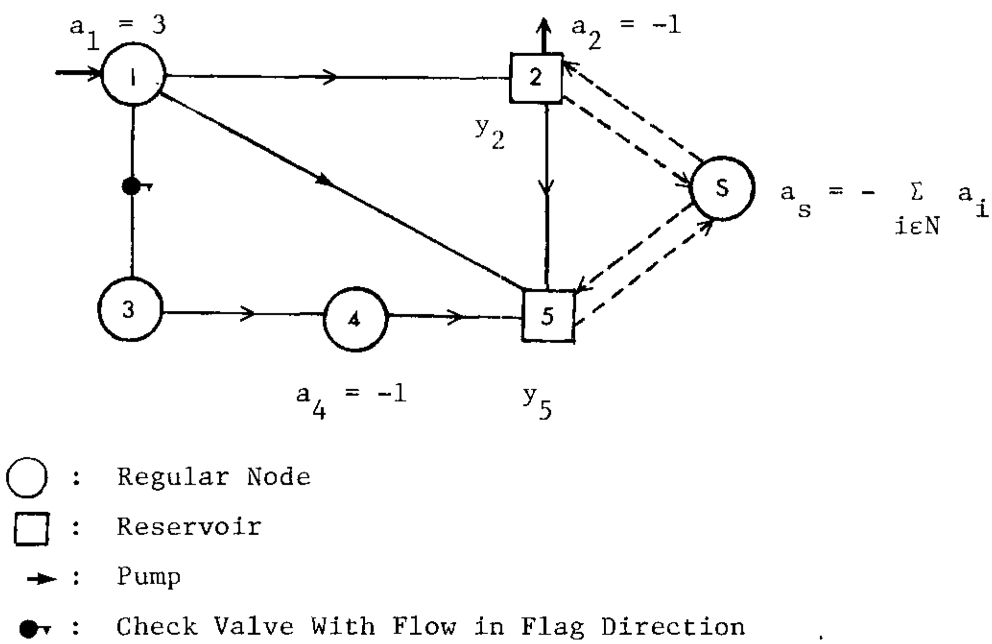

Figure 1-1. Example of a Water Distribution Network

S is a ground node connected in both directions to each reservoir to account for flow in and out of the system.

Let: $R \subset N = \{n: \text{ node n is a reservoir}\}$

$E_1 \subset E = \{(n,s), (s,n): n \in R\}$

The stationary point conditions to be satisfied are as follows:

(i) Flow Conservation:

$$\sum_{(n,j) \in E \cup E1} x_{nj} - \sum_{(i,n) \in E \cup E1} x_{in} = a_n, \quad n \in N \cup \{S\}$$

(ii) Head Discharge Behavior:

$$y_i - y_j = f(x_{ij}), \text{ for all } (i,j) \in E$$

These conditions are actually a representation of the following physical fact: the pressure at the end of the pipe (i,j) is equal to the pressure at the entrance of the pipe minus an amount of energy loss due to friction, and which depends nonlinearly on the flow x_{ij} in the pipe.

In this model, $f(x_{ij})$ is a continuous and monotone increasing function over every pipe, so that it is differentiable and convex.

(iii) Reservoir Head Conditions:

$$y_n = y_n^*, \text{ for all } n \in R.$$

The following nonlinear network is formualted to minimize what the authors call "system content" and which is in fact the total loss of "head" due to friction in the system, while satisfying flow conservation constraints:

$$\text{Minimize} \quad \sum_{(i,j)\in E} \{\int_0^{x_{ij}} f(x_{ij})\,(t)\,dt\} - \sum_{(s,n)\in E_1} \{\int_0^{x_{s,n}} y_n^*\,dt\}$$

$$+ \sum_{(n,s)\in E_1} \{\int_0^{x_{n,s}} y_n^*\,dt\}$$

Subject to Flow Conservation Equations

and $x_{ij} \geq 0,\ (i,j) \in E \cup E_1$.

This model has reportedly been applied to a trimmed version of Dallas, Texas network system with 452 nodes, 21 of which are reservoir, and 530 elements, 14 of which are pumps.

Finally, we note that this pipe network problem has applications in many other areas such as finding steady state currents and voltages in a nonlinear resistive electrical network, transportation problem with random demand traffic assignment, and dynamic production scheduling with nonlinear costs. The interested reader is referred to Lasdon and Warren (1980) for more references.

In the area of electrical networks, most models generally involve nonlinear constraints. A few specific models, however, turn out to be linearly contrainted. The optimal dispatch problem in which the total cost of allocation of power demand between the generating units in use is minimized under various restrictions (Happ (1977)). The medium-range planning in hydroelectric power systems where hydraulically and electrically interconnected reservoirs and generating plants are operated. For the operation planning overtime of the system, model have been formulated as minimization of nonlinear cost functions of meeting weekly loads (Hanscom (1976)), or as maximization of the total energy remaining in the system at the end of the planning horizon (Baxter (1975)). In the latter case, nonlinearities arise from relations describing power generation as a function of various water levels and flows. The linear constraints in all these models represent restrictions on flow conservation, and bounds on reservoir levels and outflows, as well as irrigation, navigation and conservation requirements.

In the area of power expansion, planning to meet long term forecasted demands has received considerable attention from operation researchers in the last few years. The requirements are that the total costs of investment and operation must be kept as low as possible while the risk, as measured by an average number of hours of failure during a year, must not exceed a given level.

The main features of the system units are their capital and operation costs and their "inertia": delays of several years between ordering and commissioning (2-8 years); and long life span (25-50 years). Long term planning allows the ordering of a new unit to be made in time, so that this unit is operational when required by the load growth. Clearly, late or early commissioning can cost a lot of money.

Juseret (1978) formulated the problem as a linearly constrained convex problem:

$$\text{Minimize} \quad \sum_{t=1}^{T} \left[\sum_{i=1}^{N} I_{it}\, x_{it} + G_t \left(\sum_{j=1}^{t} x_{ij} \right) \right]$$

$$\text{Subject to} \quad x_{it}^{min} \leq x_{it} \leq x_{it}^{max} \qquad i=1,\ldots,N$$

$$y_{it}^{min} \leq \sum_{j=1}^{t} x_{ij} \leq y_{it}^{max} \qquad t=1,\ldots,T$$

$$\sum_{i=1}^{N} p_{it}\, y_{it} \geq \bar{p}_t \qquad t=1,\ldots,T$$

Where:

- x_{it} are continuous variables representing the capacity in units of type i commissioned at year t.

- I_{it} is the present value of the investment costs per MW of type i commissioned at year t.

- $G_t(\sum_{j=1}^{t} x_{ij})$ is the present value of the operation costs for the whole system at year t.

- $y_{it} = \sum_{j=1}^{t} x_{ij}$ is the total capacity in units of type i on line at year t.

- p_{it} is the guaranteed capacity of unit i at time t.

The functions G_t are shown to be convex and are derived from the "load function" which is essentially a relationship between demand and time. The system clearly involves T time periods and N different types of electrical generating units. $\bar{P}_t$ denotes the sum of the guaranteed capacities required in order that the risk, as measured by the probability that the system is unable to meet the load, is equal to an acceptable level. It is seen that the objective is to minimize an index of total discounted investments and operating costs under capacity constraints on both the whole system and each unit separately, and security constraints for each time period.

Finally, multicommodity network flow problems arise in a variety of contexts, most prominently in delay optimization in data communication networks, and equilibrium studies of transportation networks (see Gallager (1974)). We discuss briefly a single commodity version of this type of problems as formulated by Bertsekas (1978).

If we consider a network (N,E) which is directed and connected, and let x_{ij} be the flow in arc $(i,j) \in E$, $E_1(i) = \{\text{node } \ell: (i,\ell) \in E\}$, $E_2(i) = \{\text{node } \ell: (\ell,i) \in E\}$, we have the following formulations:

$$\text{Minimize} \quad \sum_{(i,\ell)\varepsilon E} f_{i\ell}\,(x_{i\ell})$$

$$\text{Subject to} \quad \sum_{\ell\varepsilon E_1(i)} x_{i\ell} - \sum_{m\varepsilon E_2(i)} x_{mi} = a_i, \; i=1,\ldots,n-1$$

$$x_{i\ell} \geq 0, \; (i,\ell) \; \varepsilon \; E, \; i=1,\ldots,n-1$$

a_i is a known "traffic input" at node i; all flow is assumed to have as destination the single node n ε N. The functions $f_{i\ell}(x_{i\ell})$ are defined on (0, $c_{i\ell}$) and are convex functions; $c_{i\ell}$ refers to the capacity of the link (i,ℓ) ε E.

We also note that multicommodity network formulations have recently been applied to the assignment of circuits in case of failure in a communication network (Ishiyama (1978)). In this case, the important issues are the most economical arrangement of stand by facilities and the most effective use of these stand by facilities when they are limited.

Oil, Natural Gas, and Chemical Production: In this area, mostly linear and separable programming models have an established record of successful use for many years. Applications range from product blending and distribution, to minimum cost development of oil and gas fields. Here again we shall only give an overview of two major areas of applications: reservoir modeling, and production planning and operations. Mathematically, an oil and gas reservoir is described by a set of partial differential equations governing flow through porous media. In simulating different field operating modes, the reservoir characteristics need to be known; they can be determined by comparing actually observed reservoir behavior with the computed one. This operation is called "matching" or "parameter identification." A good survey of optimization techniques in this area is given by Durren and Slater (1977). But most recently, models involving linearly constrained programs for solving this problem have been used, such as, quadratic programming models by Yeh (1974) and least-squares models by Boberg et al (1974). In the area of oil and gas production planning and operations, the problems formulated are mostly nonlinear problems with a mix of linear and nonlinear constraints. The nonlinearities arise from modeling the relationship between reservoir productivity in a time period and the total production up to that period, as well as the total water and gas injection in the reservoir. The linear constraints, in each period, restrict the production from all reservoirs, its quality, and different capacity limitations. Cheifetz (1974) reports about such real world applications at Gulf Oil.

In the chemical industry, nonlinearities arise from product blending, as the various outputs from the chemical process are highly nonlinear functions of the process variables. Here also linear restrictions are imposed on material balance, equipment capacity bounds, external demands, and distribution of products. Another problem in the chemical and oil industries is called "pooling". It arises from the mixing of more intermediate products before they are blended. This usually happens because of such reasons as storage, transportation and pipeline availability.

Economic Production and Planning Models: In this area, models have been built both at the unit of production level and at the national level for economic policy decisions.

At the unit of production level, dynamic production scheduling models are most popular. A typical such model is the one developed by Ratliff (1978). It is restricted to convex costs and batch processing. It considers the production of n products on m identical facilities where each product i is produced in batches of size b_i over a horizon of T equal time periods. Facilities can produce only one batch of any product during a time period. The objective function represents total cost to be minimized:

$$\sum_{i=1}^{n} \sum_{t=1}^{T} a_{it} (x_{it}) + \sum_{i=1}^{n} \sum_{t=1}^{T} q_{it} \left(\sum_{j=1}^{t-1} x_{ij} \right)$$

where $a_{it}(x_{it})$ is a convex function of x_{it}, the number of batches of product i produced in period t, and $q_{it}(\sum_{j=1}^{t-1} x_{ij})$ is a convex function of $\sum_{j=1}^{t-1} x_{jt}$, the total number of batches produced before period t. Ratliff (1978) shows that this problem can be modeled as a minimum cost network flow problem with convex arc costs and integer capacities.

For national economic planning, at lot of models have been formulated as general nonlinear programming problems. Some of them have been used in practice with some success; see for example PROLOG, YULGOK and CHENERY models discussed by Lasdon and Warren (1980). An important class of planning models, however, have been formulated as quadratic programs. The quadratic objective functions arise from the following considerations:

As demand and supply vary according to prices, in some economic models some measure of income is maximized which is a function of the supply and demand functions. Normally, these functions are assumed to be linear and integrable. This makes the objective function a quadratic maximization of the sum of the producers' and consumers' surpluses. If p is a vector of prices, d and s vectors of demand and supply, then since p = f(d) and p = g(s), the return function is given by:

$$r(d,s) = \int_0^d f(v)\, dv - \int_0^s g(t)\, dt$$
$$= F(d) - G(s)$$

The quadratic problem resulting is then:

Maximize $F(d) - G(s) - c(d,s)$

Subject to Demand and supply restrictions, and bounds of various types.

where c(d,s) is a cost function of the quantities produced which depends on d and s.

The introduction of risk considerations in these models also gives rise to quadratic objectives. Applications of this type of linearly constrained model include: coal production and distribution (Dux (1977)), more general energy models (Glassey (1978), Manne (1979), and agricultural planning (Heady 1975), Bouzaher (1978)).

Finally, we note that nonlinearities in economic production models arise in a natural way when the production functions involved are nonlinear in the inputs used, such as labor ℓ, and capital k, giving: Production = $f(\ell,k)$. This is in contrast with the extensively used linear programming models which assume constant return to scales.

To close this applications section, we briefly discuss input-output models which are used at the national level for economic policy making. A typical model was studied by Glassey (1978): The problem is to maximize an index called the Net Social Payoff, a welfare index, subject to equilibrium conditions between production, consumption, imports, and exports. Mathematically, we have:

$$\text{Maximize} \quad \sum_{i=1}^{n} c_i \,|\alpha_{1i} - \beta_{1i}c_i| + z_i \,|\alpha_{2i} - \beta_{2i}\, z_i| + m_i \,|\alpha_{3i} - \beta_{3i}\, m_i| - v_i x_i$$

$$\text{Subject to} \quad (I - A)x + (I - B)m - z - c = g$$

$$x \leq x_0$$

$$x,m,z,\ c \geq 0.$$

There are n sectors in the economy, x_i is the output of sector i, m_i is the amount of product i imported, z_i is the amount of product i exported, c_i is the personal consumption of product i, A and B are matrices of technological coefficients for production and imports, and x_0 is a vector of production capacities. v_i represents the value added from sector i which includes wages, salaries and other items. The quadratic terms in the objective function represent prices times quantities, since prices are assumed to be linear of the form:

$$p_i = \alpha_{\ell i} - \beta_{\ell i}\, y_i$$

with the scalar $\alpha_{\ell i}$ and $\beta_{\ell i}$, $\ell=1,2,3$, $i=1,\ldots,n$ being computed from estimates of quantities known as prices elasticities.

4. Scope of Research

This research has as an objective the development of a general purpose algorithm for the linearly constrained nonlinear programming problem. This algorithm is based on an extension of the class of algorithms known as "projected gradient" algorithms. This extension is in two main areas. First, direction finding. In this case linearized conjugacy constraints are added recursively to the direction finding subproblem to accelerate the known linear rate of convergence of the projected gradient methods. Second, line search procedure. In this case an inexact rule is used which is based on the properties of the projection problem and which insures a monotone decrease of the objective function. The theoretical core of this research will be the formulation of the direction finding problem and the study of the global and local convergence properties of the algorithms. The performance of the algorithms will be illustrated by the numerical solution of problems from the nonlinear programming literature.

CHAPTER II

LITERATURE REVIEW

1. Introduction

All the existing methods to be reviewed in this chapter are descent methods. That is, they generate a sequence of points $\{x_k\}$ such that:

$$x_{k+1} = x_k + \lambda_k d_k \quad \text{and} \quad f(x_{k+1}) < f(x_k)$$

where d_k is the direction of search and λ_k is the steplength.

Throughout this chapter we will work with the linearly constrained problem defined in Chapter I, referring to it as problem (P):

$$\begin{aligned} P: \quad & \text{Minimize} \quad f(x) \\ & \text{Subject to} \quad x \in S = \{x \in E^n | Ax \leq b\} \end{aligned}$$

We have classified the existing methods to solve problem (P) into three main categories based on the general approach used. Also, in each category any method could be interpreted as a special case. However, the different categories are not mutually exclusive.

2. Projection Methods

In this section we discuss all the methods which solve problem (P) by projecting an unconstrained direction of search onto the feasible region. This will include all Newton-type methods as special projection procedures. In all these methods the inequality constrained problem (P) is solved as a sequence of equality constrained subproblems. For this reason, we will first consider the equality problem and its properties. We will then discuss the extension to the general inequality problem.

2.1 Definitions

Consider the following problem (P):

$$\text{Minimize} \quad f(x)$$
$$\text{Subject to} \quad x \in S = \{x \in E^n | Ax \leq b\}$$

where A is (mxn) and $b \in E^m$. If at a point x, a number of contraints are satisfied as equalities, we write $\bar{A}x = \bar{b}$, where $\bar{A}$ is (rxn), $\bar{b} \in E^r$, and $r \leq m$. We will assume the rank of $\bar{A}$ to be r and $\bar{A}$ will be referred to as the matrix of active constraints.

2.1.1 At any point x, the objective function f(x) may be approximated by a quadratic function as follows:

$$f(x) = q(d) + o(\|d\|^2), \text{ where} \qquad (2.1)$$

$$o(\|d\|^2)/\|d\|^2 \to 0 \text{ as } \|d\|^2 \to 0,$$

$$x = x_k + d, \text{ and} \qquad (2.2)$$

$$q(d) = f(x_k) + f(x_k)^t d + 1/2 d^t G_k d \qquad (2.3)$$

2.1.2 The (n-r) dimensional subspace $M_o \varepsilon E^n$ is defined as:

$$M_o = \{d_1 \varepsilon E^n | Ad = 0\} \tag{2.4}$$

The linear manifold associated with M_o is denoted as:

$$M = \{x \varepsilon E^n | Ax = b\} \tag{2.5}$$

The r-dimensional subspace $\bar{M} \varepsilon E^n$ is defined by:

$$\text{For all } d_2 \varepsilon \bar{M},\ d_2 = H_k A^t \lambda \tag{2.6}$$

where $\lambda \varepsilon E^r$ and $H_k = G_k^{-1}$

From this definition, we see that:

$$\forall d_1 \varepsilon M_o \text{ and } d_2 \varepsilon \bar{M},\ d_1^t G_k d_2 = 0 \tag{2.7}$$

and if $G_k = I$, then $d_1^t d_2 = 0$, that is d_1 and d_2 are orthogonal.

Also, for $d \varepsilon E^n$, there exists a representation:

$$d = d_1 + d_2 \tag{2.8}$$

since $E^n = M_o \oplus \bar{M}$ (Luenberger (1973), Rustem (1981)).

2.1.3 Newton method for problem (P) is based on the generation of descent directions which are the solution of the following problem:

$$\text{Minimize} \quad q(d)$$
$$\text{Subject to} \quad d \in M_o \tag{2.9}$$

which leads to a solution $\bar{x}$ of the linear equality constrained problem:

$$\text{Minimize} \quad f(x)$$
$$\text{Subject to} \quad x \in M \tag{2.10}$$

Associated with (2.10) are the Kuhn-Tucker necessary conditions for $\bar{x}$ to be a solution:

$$\nabla f(\bar{x}) - \bar{A}^t \lambda = 0 \tag{2.11}$$

$$\bar{A}\bar{x} - \bar{b} = 0 \tag{2.12}$$

Equation (2.11) form a system of n equations in r unknowns which have the unique solution (Powell, 1974-b, theorem 1.5):

$$\bar{\lambda} = (\bar{A}\bar{A}^t)^{-1} \bar{A} \nabla f(\bar{x}) \tag{2.13}$$

At points other then $\bar{x}$, there is no vector $\bar{\lambda}$ which satisfies (2-11). In this case estimates of $\bar{\lambda}$ may be obtained from the solution of problem (2-9).

2.2 The Projection Problem

The following result is important in that it shows that the solution to problem (2-9) above is merely the projection of the unconstrained minimum of q(d) (given by (2-3)) onto the subspace M:

Theorem 1. (Rustem, (1981)) If $\bar{x}$ is a solution of the minimum norm problem:

$$\text{Minimize} \quad \{\|x - x_u\|_G^2 \mid x \in S\} \tag{2.14}$$

where $\|x - x_u\|_G^2 = (x - x_u)^t G(x - x_u)$,

then it is also a solution of the quadratic optimization problem:

$$\text{Minimize} \quad \{q(x) \mid x \in S\} \tag{2.15}$$

$$\text{where} \quad q(x) = a^t x + 1/2 x^t Gx$$

$$\text{and} \quad x_u = -G^{-1} a = -Ha \ (H = G^{-1})$$

Based on this result, we see that the projection of the vector $d_1 \in E^n$ onto M_o, call it $\bar{d}_1$, is given by the solution of the following problem:

$$\text{Minimize} \quad \{\|-H_k \nabla f(x_k) - d_1\|_{G_k}^2 \mid d \in M_o\} \tag{2.16}$$

which is also a solution to problem (2-9).

Now we turn to the different ways of solving problem (2-9). One of the following two approaches can be taken:

2.2.1 Solving a Reduced Problem

In this method any vector $d_1 \in M_o$ may be expressed as a linear combination of the (n-r) basis vectors z_j ($j = 1,2,\ldots,n-r$) of the subspace M_o. Therefore, letting the $nx(n-r)$ matrix Z be defined by:

$$Z = [z_1, \ldots, z_{n-r}]$$

any $d_1 \varepsilon M_o$ may be expressed as:

$$d_1 = Zv \tag{2.17}$$

where v is an (n-r) vector.

This will lead to expressing problem (2.16) above as an (n-r) dimensional unconstrained minimization problem:

$$\underset{v}{\text{Minimize}} \quad \{\| -H_k \nabla f(x_k) - Zv\|^2_{G_k}\} \tag{2.18}$$

whose solution is obtained from:

$$2Z^t \; [-\nabla f(x_k) - G_k Zv] = 0$$

that is:

$$(Z^t G_k Z) \; v = -Z^t \; \nabla f(x_k)$$

which in conjunction with (2.17) gives:

$$\bar{d}_1 = -Z(Z^t G_k Z)^{-1} Z^t \nabla f(x_k) \tag{2.19}$$

We note that in this method $(Z^t G_k Z)$ and not G_k is required to be positive definite to insure a direction of descent. We will see later the importance of this fact for certain methods. Also $(Z^t G_k Z)$ is of dimension (n-r) which makes it computationally more attractive (Gill and Murray, 1974-a).

2.2.2 Solving from the Optimality Conditions

From (2-8) $d \in E^n$, $d = \bar{d}_1 + \bar{d}_2$, $\bar{d}_1 \in M_o$, $\bar{d}_2 \in \bar{M}$ and hence:

$$\bar{d}_1 = d - \bar{d}_2 \tag{2.20}$$

Here, $\bar{d}_2$ is first explicitly found. From (2.20) above, $\bar{d}_2$ is the projection of d onto the subspace $\bar{M}$, with $d_2 = H_k A^t \lambda$, $\lambda \in E^r$; thus $\bar{d}_2$ is given by the solution of:

$$\text{Minimize} \quad \{\| d - d_2 \|^2_{G_k} \mid d \in \bar{M}\} \tag{2.21}$$

or, equivalently, in terms of $\bar{\lambda}$ which solves:

$$\underset{\lambda}{\text{Minimize}} \quad \{\| d - H_k \bar{A}^t \lambda \|^2_{G_k}\} \tag{2.22}$$

$$\text{giving} \qquad \bar{d}_2 = H_k \bar{A}^t \bar{\lambda}$$

If d is given by: $d = -H_k \nabla f(x_k)$, then the unconstrained minimum of (2.22) with respect to λ yields:

$$\bar{\lambda} = -(\bar{A}H_k\bar{A}^t)^{-1} \bar{A}H_k\nabla f(x_k) \qquad (2.23)$$

which in turn from (2.20) gives:

$$\bar{d}_1 = d - H_k\bar{A}^t\bar{\lambda}$$

$$= -H_k\nabla f(x_k) + H_k\bar{A}^t (\bar{A}H_k\bar{A}^t)^{-1} \bar{A}H_k\nabla f(x_k)$$

$$= -[I - H_k\bar{A}^t (\bar{A}H_k\bar{A}^t)^{-1} \bar{A}] H_k\nabla f(x_k) \qquad (2.24)$$

Note that if $(\bar{A}H_k\bar{A}^t)$ and (Z^tG_kZ) are positive definite, the vectors $\bar{d}_1$ computed from (2.19) and (2.24) are identical.

A condition for $\bar{x}$ to be a solution of the minimization problem with linear equality constraints is that $\bar{d}_1$ in (2.24) be zero at $\bar{x}$. Alternatively, since

$$\left.\begin{aligned} \nabla f(\bar{x}) &= \bar{A}^t\bar{\lambda} \\ \overline{Ax} &= \bar{b} \\ Z^t\bar{A}^t &= 0 \end{aligned}\right\} \iff Z^t\nabla f(\bar{x}) = 0 \qquad (2.25)$$

(2.25) means that the projected gradient is zero (Gill and Murray, 1974-a).

2.2.3 Lagrangian Multipliers Estimates

Since at $x_k \neq \bar{x}$, there is no λ_k such that: $\bar{A}^t\lambda_k = \nabla f(x_k)$, a simple estimate of the Lagrangian multipliers at x_k is given by the vector λ_k which solves:

$$\text{Minimize} \quad \{\|\nabla f(x_k) - \bar{A}^t\lambda\|^2\}$$

that is:

$$\lambda_k = (\bar{A}\bar{A}^t)^{-1}\,\bar{A}\,\nabla f(x_k) \tag{2.26}$$

Also, an estimate of the Lagrangian multipliers at $(x_k + \bar{d}_1)$ can be obtained from the solution of problem (2.9). That is, at the solution of (2.9), there exists λ_q such that:

$$\bar{A}^t\lambda_q = \nabla q(d)$$

from which we have:

$$\lambda_q = (\bar{A}\bar{A}^t)^{-1}\,\bar{A}\,[G_k\bar{d}_1 + \nabla f(x_k)] \tag{2.27}$$

λ_k given by (2.26) is a "first order approximation" since it can be shown that:

$$\|\bar{\lambda} - \lambda_k\| = O(\|\bar{d}\|) \tag{2.28}$$

and λ_k given by (2.27) is a "second order approximation" since it can also be shown that:

$$\|\bar{\lambda} - \lambda_q\| = O(\|\bar{d}\|^2) \tag{2.29}$$

where $\bar{\lambda}$ is the vector of Lagrangian multipliers at the optimal solution $\bar{x}$.

2.3 Projection Methods for Problem (P)

Projection algorithms for problem (P) generate their descent directions either by solving a variant of the problem (2.22),

$$\underset{\lambda}{\text{Minimize}} \quad \{\|d - H_k \bar{A}^t \lambda\|^2_{G_k}\} \tag{2.30}$$

or a variant of the problem (2.18),

$$\underset{v}{\text{Minimize}} \quad \{\|-H_k \nabla f(x_k) - Zv\|^2_{G_k}\} \tag{2.31}$$

However, most of them do not use the actual current Hessian matrix G_k, but a positive definite approximation to it. This makes them belong to the class of Quasi-Newton methods.

2.3.1 Methods Based on Problem (2.30)

Most methods here, as we will see, use approximations to G_k, or operators involving H_k, where $H_k = G_k^{-1}$.

2.3.1.1. The first algorithm that applied projections to linearly constrained optimization was Rosen's gradient projection (Rosen, 1960). The descent directions and the Lagrangian multipliers generated by Rosen's algorithm may be obtained by setting $G_k = I$ in (2.30), thus yielding:

$$\bar{\lambda} = -(\bar{A}\bar{A}^t)^{-1}\, \bar{A}\, \nabla f(x_k)$$

$$\bar{d}_1 = -[I - \bar{A}^t\, (\bar{A}\bar{A}^t)^{-1}\, \bar{A}]\; f(x_k) \qquad (2.32)$$

The directions generated by Rosen's algorithm are basically the steepest descent directions projected into the intersection of the currently active constraints.

2.3.1.2. A method which updates H_k is due to Murtagh and Sargent (1969). In this method $\bar{\lambda}$ and $\bar{d}_1$ are given by (2.23) and (2.24) respectively, where a rank-one updating formula is used to update an approximation to H_k, and then $\bar{\lambda}$ and $\bar{d}_1$ are formed.

2.3.1.3. Another method uses the Davidon-Fletcher-Powell Quasi-Newton formula to update an approximation to the operator:

$$P[H_k] = \{[I - H_k\bar{A}^t(\bar{A}H_k\bar{A}^t)^{-1}\bar{A}]H_k\} \qquad (2.33)$$

directly. This is the well known Goldfarb's (1969) method.

The above three methods all use some sort of active set strategy to solve the linear inequality problems as a sequence of linear equality subproblems. A detailed discussion of most used active set strategies is left to a subsequent section. But we will briefly mention here that the basic strategy is to minimize the objective function over the face of the constraint polytype formed by the currently active constraints. This face is changed only when the search for a minimum along the projected search direction (2.24) encounters another contraint or when the minimum value of the objective function may be decreased further only by moving off this face.

We now discuss in more detail some important aspects of the above methods and their relationships.

Davidon (1959) extended his algorithm for unconstrained minimization to solve the linear equality problem by pointing out that the initial approximation of the inverse hessian should be chosen to be in the null space of the matrix $\bar{A}$. One such matrix is the orthogonal projection operator:

$$P_o = P[I] = [I - \bar{A}^t(\bar{A}\bar{A}^t)^{-1}\,\bar{A}]$$

which projects all vectors in E^n onto the subspace M_o with respect to the euclidean norm (i.e., with $G_k = I$). Thus clearly:

$$\bar{A}P_o = 0$$

which ensures that the direction of search computed using the approximation P_o:

$$\bar{d}_1 = -P_o \nabla f(x_o)$$

satisfies $\bar{A}\bar{d}_1 = 0$. Hence $\bar{d}_1$ is feasible ($\bar{d}_1 \varepsilon M_o$). The inverse Hessian approximation is updated with incoming curvature information obtained along $\bar{d}_1$. As above, feasibility of subsequent search directions is maintained because the approximations always remain in the null space of the matrix $\bar{A}$. The updating is done using the Davidon-Fletcher-Powell (DFP) formula. However the rank-one or the BFGS may also be used as was shown by Powell (1974-a) and Fischer (1981).

Goldfarb (1969) extended Davidon's method to inequality constraints using the techniques developed by Rosen (1960). For the equality constraints case with a quadratic objective function, Goldfarb proved convergence in (n-r) iterations and the approximation to P_k updated by the algorithm becomes:

$$P_{n-r} = P[H] = \{H - H\bar{A}^t(\bar{A}H\bar{A}^t)^{-1} \bar{A}H\}$$

where H is the true inverse Hessian of the quadratic objective function. Only the first order estimates of the Lagrangian multipliers are computed in Goldfarb's method since the approximation to P_k always remains in the null space of $\bar{A}$.

A disadvantage of Goldfarb's method, however, is that if a constraint is dropped from the active set such that the rank of the approximation to P_k is increased by one, no information about the curvature of the objective function is available in the direction of the normal of this constraint. Murtagh and Sargent (1969) have attempted to overcome this by updating directly the approximation to the inverse Hessian rather than (2.33). The projection operator $P[\hat{H}_k]$ is then constructed given $\hat{H}_k$, the approximation to H. They use the rank-one formula for updating $\hat{H}_k$. An important feature of this formula is that it does not require exact line searches, which makes it attractive compared to others. However, when a constraint becomes active and is added to the active set, $\hat{H}_k$ is updated by the rank-one formula Gill and Murray (1974-b, pp. 72-74) argue that updating $\hat{\hat{H}}_k$ in E^n does not provide curvature information in the direction of the normals of the active constraints. Since the descent directions, $\bar{d}_l$, have to be orthogonal to these normals to be feasible, the updates $\hat{H}_k$ contain curvature information in these directions only. They point out that, only in the case where movement off a constraint occurs near the point of its addition to the active set, that such curvature information becomes significant. On the other hand, if a constraint is dropped long after it has become active, the curvature information in the direction of its normal, obtained prior to its becoming active, has no longer any significance. Also, since $\bar{A}\bar{d}_l = 0$, if the BFGS formula is used to update $\hat{H}_k$, for descent steps taken in the intersection of the same constraints, then, by inspection of the BFGS formula, it follows that:

$$(\bar{A}\hat{H}_{k+1}\ \bar{A}^t) = (\bar{A}\hat{H}_k\bar{A}^t)$$

thus, the Lagrange multipliers computed according to:

$$\lambda_k = (\bar{A}\hat{H}_k\bar{A}^t)^{-1} \bar{A}\hat{H}_k \nabla f(x_k)$$

may no longer be regarded as second order estimates since if we start with the approximation $\hat{H}_o = I$, then:

$$(\bar{A}\hat{H}_k\bar{A}^t) = (\bar{A}\hat{H}_o\bar{A}^t) = (\bar{A}\bar{A}^t).$$

Similarly, the descent direction:

$$\bar{d}_1 = -P[\hat{H}_k]\nabla f(x_k)$$

is no longer equivalent to (2.24).

Powell (1974-a) established the equivalence, under certain assumptions, of updating approximations to P_k and updating $\hat{H}_k$ then computing $P[\hat{H}_k]$. Powell also proved the convergence of Goldfarb's method for a wider class of quasi-Newton updates and for inequality constraints. The rate of convergence for quadratic functions and using exact line searches is established to be $(n+\ell)$ where ℓ is the number of faces of the constraint polytope over which the search for results to inexact line searches and showed superlinear convergence of a modification of Goldfarb's method for general objective functions.

3.1.4. Other methods that could be seen as extensions of problem (2.30) are methods that solve efficiently a series of linear or quadratic programming problems where the restriction $d_1 \in M_o$ is relaxed to $d_1 \in S$. The method of hypercubes due to Fletcher (1972a) generates quadratic approximations to the objective function f at the point x_{k+1} as:

$$q(x_{k+1}) = f(x_k) + \nabla f(x_k)^t (x_{k+1} - x_k) + 1/2(x_{k+1} - x_k)^t G_k(x_{k+1} - x_k) \qquad (2.34)$$

To make the approximation as close as possible, the stepsize is restricted. Thus the method involves successive solutions to the problem:

$$\begin{array}{ll} \text{Minimize} & q(x) \\ \text{Subject to} & Ax \leq b \\ & \|x - x_k\|_\infty \leq \delta_\infty \end{array} \qquad (2.35)$$

Assuming (2.34) is a positive definite approximation, its unconstrained optimum is given by:

$$\nabla\hat{q}(x_k) = \nabla f(x_k) + \hat{G}_k(x_k - \bar{x}) = 0$$

$$\text{giving} \quad \bar{x} = x_k - \hat{G}_k^{-1} \nabla f(x_k) \qquad (2.36)$$

By theorem 1 we can see that the solution of (2.35) may be expressed as the projection problem:

$$\text{Minimize} \quad \{1/2\|x - \bar{x}\|^2_{\hat{G}_k}\}$$

$$\text{Subject to} \quad Ax \leq b$$

$$\text{And} \quad \|x - \bar{x} - H_k \nabla f(x_k)\|_\infty \leq \delta_\infty$$

The matrix $\hat{G}_k$ is updated using a rank 2 formula.

An earlier variant of problem (2.35), without the constraint on the step size, was developed by Wilson (1963) and described in Beale (1967). In this approach G_k is evaluated at every iteration.

2.3.2 Methods Based on Problem (2.31)

This section deals with descent methods based on the solution of the unconstrained optimization problem defined by (2.31). Algorithms in this class perform an unconstrained minimization of the function projected onto the (n-r) dimensional subspace defined by the active constraints. Because of that the determination of a basis of the subspace M_o becomes important.

Given the matrix $\bar{A}$ of active constraints assumed to have rank r, the following (nxn) matrix T is defined as:

$$T = \begin{bmatrix} \bar{A} \\ \bar{V} \end{bmatrix} \quad : \quad \begin{array}{ll} \bar{A} & (r\text{x}n) \\ V & \text{is an } (n-r)\text{x}n \text{ matrix such that } T \text{ is nonsingular.} \end{array} \qquad (2.37)$$

The following result is very useful in determining a basis for M_o (Gill and Murray, 1974-a, theorem 2.5):

<u>Theorem 2</u>. The last (n-r) columns of the matrix T^{-1} span the subspace M_o.

Based on this result, the matrix Z in the definition of d_1 (i.e., $d_1 = Zv$) may be set equal to the vectors t_j, j=r+1, r+2,...,n, where t_j is the jth column of T^{-1}.

Also, using T and Z, Lagrangian multiplier estimates can be obtained using the relation:

$$\bar{A}^t \lambda_q = \nabla q(\bar{d}_1)$$

that is:

$$\bar{A}^t \lambda_q = [\bar{A}^t | V^t] \begin{bmatrix} \lambda_q \\ \hline 0 \end{bmatrix} = T^t \begin{bmatrix} \lambda_q \\ \hline 0 \end{bmatrix}$$

$$= \nabla q(\bar{d}_1) = G_k \bar{d}_1 + \nabla f(x_k)$$

which implies that:

$$\begin{bmatrix} \lambda_q \\ \hline 0 \end{bmatrix} = \begin{bmatrix} (\bar{A}\bar{A}^t)^{-1}\bar{A} \\ \hline Z \end{bmatrix} [G_k d_1 + \nabla f(x_k)] \qquad (2.38)$$

(2.38) gives second order Lagrange multipliers estimates. We note that first order estimates could be obtained from the same expression simply by ignoring the second order term in (2.38).

We recall here that the expression for $\bar{d}_1$ is:

$$\bar{d}_1 = -Z(Z^t G_k Z)^{-1} Z^t \nabla f(x_k) \qquad (2.39)$$

It is clear at this point that different choices of the matrix V in the formation of T will lead to different algorithms.

2.3.2.1 The Reduced Gradient Method of Wolfe (1967)

This method generates Z using T^{-1} by selecting the columns of V from the normals of inactive constraints. This approach gives the descent direction:

$$\bar{d}_1 = -ZZ^t \nabla f(x_k)$$

Gill and Murray (1974-a) show that effectively this descent direction is not the same as the direction (2.39) above, even when $G_k = I$.

2.3.2.2. The Variable Reduction Method of McCormick (1970-a)

This method chooses the columns of V from the columns of the identity matrix I, which gives T as:

$$T = \left[\begin{array}{c|c} (\bar{A}^1) & (\bar{A}^2) \\ \hline 0 & I \end{array}\right] \Rightarrow Z = \left[\begin{array}{c} -(\bar{A}^1)^{-1}\bar{A}^2 \\ \hline I_{rxr} \end{array}\right] \qquad (2.40)$$

where $\bar{A}^1$ and $\bar{A}^2$ are respectively (rxr) and (n-r)xr submatrices of $\bar{A}$ such that:

$$\bar{A} = [\bar{A}^1 | \bar{A}^2] \qquad (2.41)$$

The matrix A^{-1} corresponds to the first r elements of the vector $x \in E^n$ and $\bar{A}^2$ to the remaining (n-r) elements.

The DFP updating formula with resetting is used for approximating H_k in M_o. A further extension of this algorithm has been made by McCormick (1970-b) by allowing the computation of G_k in E^n and then computing $(Z^t G_k Z)$ used in (2.39) above.

In the variable reduction method the linear equality constrained problem (2.10) may be reduced to an unconstrained optimization problem as follows: The constraints $\bar{A}x = \bar{b}$ may be written using (2.41):

$$\bar{A}^1 x_1 + \bar{A}^2 x_2 = \bar{b}$$

$$\Rightarrow x_1 = -[(\bar{A}^1)^{-1}]\bar{A}^2 x_2 + (\bar{A}^1)^{-1}\bar{b}$$

with

$x_1 \in E^r$, the vector of dependent variables,

$x_2 \in E^{n-r}$, the vector of independent variables, and

$f(x) = f(x_1, x_2) = f[(\bar{A}^1)^{-1}\bar{A}^2 x_2, x_2] = f(Zx_2)$.

Another method which minimizes $f(x_2)$ is due to Ganzhella (1970) who uses a variable metric method to minimize the objective function in the linear manifold M.

2.3.2.3. Factorization Methods

Gill and Murray (1974-a) pointed out that a poor choice of V in (2.37) may lead to an ill-conditioned matrix T and thus to an ill-conditioned matrix Z affecting the expressions:

$$[Z^t \nabla f(x_k)] \quad \text{and} \quad [(Z^t G_k Z)]$$

This, in turn, could result in poor estimates of the Lagrange multipliers in (2.38) and the computation of the direction $\bar{d}_1$ in (2.39).

The condition of any matrix A can be measured by the magnitude k(A), the condition number of A, which is defined as

$$k(A) = \|A\| \, \|A^+\|$$

where A^+ is the pseudo-inverse of A.

Gill and Murray (1974-a) recommend a choice of Z which makes the condition number of (Z^tG_kZ), $K(Z^tG_kZ)$, dependent only on the conditioning of G_k, since

$$K(Z^tG_kZ) \leq K(G_k)[K(Z)]^2$$

They propose to that effect an LQ factorization of $\bar{A}$ such that L is an (rxr) lower triangular matrix and Q is an orthogonal matrix, that is $Q^tQ = Q^tQ = I$, giving:

$$\bar{A} = [L|0]Q$$

Furthermore, Q is partitioned as follows:

$$Q = \begin{bmatrix} Q_1 \\ \hline Q_2 \end{bmatrix} \begin{matrix} (rxn) \\ (n-r)xn \end{matrix} \qquad \bar{A} = [L|0]\begin{bmatrix} Q_1 \\ \hline Q_2 \end{bmatrix} = LQ_1$$

Now, since:

$$\bar{A}\left[\frac{Q_1}{Q_2}\right] = [L|0]QQ^t = [L|0],$$

then clearly, $\bar{A}Q_2^t = 0$.

Therefore, if Q_2 is used as the required matrix V, we will have:

$$T = \left[\frac{\bar{A}}{V}\right] = \left[\frac{\bar{A}}{Q_2}\right] \text{ and } (T^t)^{-1} = \left[\frac{(\bar{A}\bar{A}^t)^{-1}\bar{A}}{Q_2}\right]$$

that is Z is chosen to be the matrix Q_2^t, and such a choice results in a minimum condition number. Also, for a positive definite G_k,

$$K(Z^tG_kZ) < K(G_k)$$

which means that the conditioning of the algorithm now depends on that of the problem only.

This choice of Z is used by Gill and Murray to compute $\bar{d}_1$ as in (2.39) both in a Newton and a Quasi-Newton setting. It is widely acknowledged in the literature that this is one of the most numerically stable procedure in this class of methods.

A variant of Goldfarb's algorithm that uses the transformation T with the columns of V chosen from the normals of the inactive constraints has been suggested by Buckley (1975). Finally, the efficient implementation of the above algorithm using the Cholesky factors of G_k, (Z^tG_kZ) or H_k, the orthogonal factorizations of $\bar{A}$ and the modification of these factorizations at each iteration, have been extensively discussed by: Bartels, Golub and Saunders (1970); Gill, Golub, Murray and Saunders (1974); Goldfarb (1976).

A combination of some of the best features of the above methods is found in the work of Murtagh and Saunders (1978), who combine stable Quasi-Newton methods with the generalized reduced gradient where the variables are decomposed into nonbasic variables, which are fixed at their bounds, basic variables which are used to maintain feasibility, and superbasic variables which are allowed to vary. We note, however, that Murtagh and Saunders' algorithm is mostly suited for large scale problems with few nonlinearities, therefore allowing extensive use of the revised simplex and sparce matrix techniques.

2.3.3 Generalization of the Godstein-Levitin-Polyak (GLP) Projection Algorithm

The (GLP) algorithm, contrary to other feasible direction algorithms, proceeds along arcs on the constraint surface using the steepest descent direction [Goldstein (1964), Levitin and Polyak (1966)]. A typical iteration is of the form:

$$x_{k+1} = P_S[x_k - \lambda_k \nabla f(x_k)]; \; k = 0,1,\ldots, \qquad (2.42)$$

where $P_S(y)$ denotes the unique projection of a vector y on the feasible region S; $\lambda_k > 0$, denotes the step size.

If $S = \{x \in E^n | Ax < b\}$, the following quadratic program is solved:

$$\text{Minimize} \quad \{1/2\|[x_k - \lambda_k \nabla f(x_k)] - x\|_2^2\} \tag{2.43}$$

$$\text{Subject to} \quad Ax \leq b$$

We note that this procedure differs from Rosen's gradient projection method in that the points x_k and x_{k+1} need not lie on one face of the constraint polyhedron.

If $\bar{S} = \{x \in E^n | Ax = b\}$, then (2.43) is written in explicit form as follows:

$$\text{Minimize} \quad \{1/2\|[x_k - \lambda_k \nabla f(x_k)] - x\|_2^2\}$$

$$\text{Subject to} \quad Ax = b$$

whose Kuhn-Tucker conditions are:

$$x - [x_k - \lambda_k \nabla f(x_k)] = A^t u$$

$$Ax = b$$

from which:

$$x_{k+1} = [I - A^t(AA^t)^{-1}A]\ (x_k - \lambda_k \nabla f(x_k))$$

$$= P_{\bar{S}}\ [x_k - \lambda_k \nabla f(x_k)]$$

while in Rosen's method:

$$x_{k+1} = x_k - \lambda_k P_{\bar{S}} \nabla f(x_k)$$

An implementable version of (2.43) was developed by Bazaraa and Goode (1981) for problem (P) where the projection of the negative gradient is into the set of equality and the binding and near binding constraints.

A generalization of the GLP algorithm as briefly described by Bertsekas (1976) was developed by Rustem (1981) who uses more general projections and Newton or Quasi-Newton descent directions. This algorithm uses approximate line searches and is shown to be globally convergent with a superlinear or quadratic speed. The iterations of the algorithm are determined by solution of the positive definite quadratic programming problem:

$$\text{Minimize} \quad 1/2 \| x - x_k + G_k^{-1} \nabla f(x_k) \|_{G_k}^2$$

$$\text{Subject to} \quad x \in S$$

Where G_k is either the Hessian of the objective function or denotes a symetric positive definite approximation to it.

We note that the generalization here comes from the fact that the projection of the unconstrained Newton or Quasi-Newton step is sought, and a metric, rather than a Euclidean norm is used.

2.4 Active Set Strategies

We have mentioned at the outset that most methods we have discussed solve the inequality constrained problem by solving a sequence of equality constrained subproblems. So in this section we briefly discuss methods designed to perform the extension from the equality problem to the inequality constrained one. There are basically two approaches to handling the inequality constraints at each iteration.

2.4.1. Adding Slack Variables

This strategy is a simple extension of the idea of slack variables used in standard linear programming. This is discussed by Sargent and Murtagh (1973) in relation to nonlinear constraints.

We note that, to our knowledge, this strategy is not used to solve problem (P) except by those methods based on the simplex procedure. Clearly, the adding of slack variables increases the dimension of the problem, an unattractive feature.

2.4.2. Using the Binding Constraints at Each Iteration

This second type of strategy aims at including only a subset of the inequality constraints in the active set during each iteration. But there are variations in including and dropping constraints from the active set. Such strategies are mostly suited for problem (P).

At a current point x, the constraints satisfied as equalities (together with any equality constraints in the problem) are included in the active set. A constraint is added to the set when the search direction from x hits one which is not already in the active set; this is usually accomplished through a minimum ratio test to determine the nearest inactive constraint. Dropping a constraint from the active set is accomplished in one of two ways:

i) Retain all the currently active constraints until a minimum is found in their intersection. Inspection of the Lagrange multipliers at that minimum point should indicate if we might obtain a lower point by deleting the constraint with the most negative Lagrange multiplier from the current active set. Under the assumption that the number of stationary points is finite, and the constraints active are linearly independent, it is shown that this scheme will terminate at a strong local minimum after only a finite number of changes in the active set (Gill and Murray, 1974-a). An obvious disadvantage of this scheme is that we may be using a lot of computational effort in order to find the minimum on a subspace which is far from the solution.

ii) Compute estimates of the Lagrange multipliers at each iteration and move off the constraints for which the value of the multipliers is negative. Since the Lagrange multiplier of a constraint gives the rate of change of the objective function in relation to this constraint (see for example Sargent (1974)), a negative multiplier implies that if the corresponding constraint is dropped then a reduction may be obtained in the objective function value. Thus if a single constraint is to be dropped, the one with the most negative multiplier is chosen. However, given that the relation:

$$\bar{A}^t u = g(\bar{x}) = \nabla f(\bar{x})$$

is not satisfied exactly because we are not at the solution to (P) yet, in general, the estimates of the multipliers u tend to be very inaccurate. This may lead to a constraint being repeatedly droped and then included again in the active set and therefore progress to the solution can be very slow. So, in general, it is not possible that this strategy will terminate after a finite number of changes in the active set. This pheonomeon is called "zigzagging" as illustrated by the example of Wolfe (1972) which will be found in section three to follow this one.

In practice, most people use a combination of i) and ii), such as illustrated in the computational study of Lenard (1979). Zoutendijk (1960) proposed the following strategy: a constraint that has previously been dropped and returned to is kept in the active set until a minimum solution is obtained on a subspace. This will ensure finite termination.

McCormick (1969) avoids zigzagging by projecting the direction of search onto successive active sets formed by incorporating each new constraint as it is encountered. This is called "bending" of the current direction of search, which will continue until either an unconstrained minimum is found with respect to the step size, or the matrix of constraints defines a subspace containing a local minimum. Only at this point is the gradient information recomputed and a new direction of search defined. McCormick (1970) provides a proof that zigzagging cannot occur in this way.

Finally, we note that criteria based on the value of the Lagrange multipliers and the amount of reduction in the objective function have been formulated in terms of the solution to a complementary pivoting algorithm (Lemke, 1968).

2.5 Quasi-Newton Updates

There are major difficulties associated with using the exact Hessian of the objective function in any optimization procedure.

First, the computation of second derivatives is generally very expensive. Second, when the starting point, x_o, is far from the solution, Newton methods, which use exact Hessian information, may fail to converge even in the unconstrained case. Second order rate of convergence of Newton methods has been proven for functions with positive definite Hessians which are Lipschitz continuous at the optimum, provided x_o is sufficiently close to the solution. Such results are found in Orega and Rheinboldt (1970), and Dennis and More (1977). Third, when x_o is far from the solution, restrictions on the step size, such as the following by Wolfe (1969), have been suggested to help convergence:

$$\nabla f(x_k + \alpha_k d_k)^t d_k > c_1 \nabla f(x_k)^t d_k$$

$$f(x_k + \alpha_k d_k) - f(x_k) < c_2 \alpha_k \nabla f(x_k)^t d_k$$

where c_1 and c_2 are scalars such that: $0 < c_2 < c_2 < 1$, and $0 < \alpha_k < \alpha_{max}$.

But even with such precautions, a well defined direction may be computed which is orthogonal to $\nabla f(x_k)$. This results in a zero steplength, stopping further progress toward the solution. This point is illustrated by the following example due to Powell (1966).

$$\text{Minimize} \quad f(x) = x_1^4 + x_1 x_2 + (1 + x_2)^2$$

If $x_o = (0,0)^t$ is chosen as the starting point, then

$$g_o = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad G_o = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}, \quad d_o = -G_o^{-1} g_o = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

Since a line search along $\pm\, d_o$ changes only the x_1 component of the vector x, it is clear that $x_1 = 0$ will minimize f(x) and we fail to make any progress. The problem is due to the fact that $g_o^t d_o = 0$, and the directions $\pm\, d_o$ are not descent directions.

These difficulties may be overcome by restricting the steplength to the region in which the quadratic approximation of the objective function is valid (see for example Fletcher (1972)) or by forcing the Hessian matrix to be positive definite when it is not. In this case, unless $\nabla f(x_k) = 0$, $d_k = -H_k \nabla f(x_k)$ is a descent direction.

In constrained optimization, forcing the approximation $\hat{G}_k$, to the hessian matrix G_k, to be positive definite may be a serious problem. An objective function with a singular Hessian matrix need not have an unconstrained minimum. Its minimum may, however, exist on some constraints if the Hessian of the Lagrangian has a positive definite projection on the intersection of these constraints. We note that for problem (P) the Lagrangian is defined as:

$$L(x,u) = f(x) - u^t(Ax - b) \text{ with } \nabla L_{xx} = \nabla_{xx} f$$

In the unconstrained case, the Hessian matrix is positive definite in the neighborhood of a strong local minimum. The same need not be true at the constrained minimum and the convergence rate may be affected by forcing G_k to be positive definite when it is not.

For the linearly constrained case, Gill and Murray (1974-b) suggest a method restricting only the projection of the Hessian matrix onto the set of active constraints, $(Z^t G_k Z)$, to be positive definite [see (2.38)]. Consequently, in the linear equality constrained subproblem (2.16), when $(Z^t G_k Z)$ is positive definite and $Z^t \nabla f(x_k) = 0$, x_k is a strong loal minimum.

The algorithms described in Section 3.3 that project the unconstrained step onto the inequality constraints, rather than taking steps along a descent direction projected onto the active constraints, may generate successive points on different faces of the constraint polyhedron. This may imply different active sets for successive points. When G_k is updated at these points it collects curvature information in all directions. So it seems that these methods should provide more curvature information than active set strategy methods.

We now briefly discuss, after a motivation of the need for Quasi-Newton approximations, the important Quasi-Newton updates for the inverse Hessian approximation $\hat{H}_k$. These updates are based on the general formula introduced by Broyden (1967) of which one representation is:

$$\hat{H}_{k+1} = \hat{H}_k - \frac{\hat{H}_k \gamma\gamma^t \hat{H}_k}{\gamma^t \hat{H}_k \gamma} + \frac{\delta\delta^t}{\delta^t \gamma} + \beta_k w w^t \qquad (2.44)$$

where:

$$\gamma = \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$\delta = x_{k+1} - x_k$$

$$w = \hat{H}_k \gamma - \frac{\delta}{\gamma^t \delta} \gamma^t \hat{H}_k \gamma$$

and the scalar β_k is a free parameter whose values define different updating procedures:

i) The Rank-One formula (Broyden, 1967) is obtained for:

$$\beta_k = \frac{\gamma^t \gamma}{(\delta^t \gamma)(\gamma^t \hat{H}_k \gamma) - (\gamma^t \hat{H}_k \gamma)^2}$$

ii) The Davidon-Fletcher-Powell (DFP) due to Davidon (1959), Fletcher and Powell (1963) is obtained for:

$$\beta_k = 0$$

iii) The Broyden-Fletcher-Goldfarb-Shanno (BFGS) due to Broyden (1970), Fletcher (1970), Goldfarb (1970), and Shanno (1970) is obtained for:

$$\beta_k = \frac{1}{\gamma^t \hat{H}_k \gamma}$$

The (BFGS) is generally acknowledged to be the best choice among the members of the family (2.44). A desirable property of the members of this family is that they satisfy the, so-called, Quasi-Newton equation:

$$\hat{H}_{k+1}\gamma = \delta$$

which is an attractive feature because if f(x) were quadratic with the inverse of its Hessian given by H, then $H\delta = \delta$. For the quadratic case, Huang (1970) showed that for an unconstrained optimization procedure taking steps along $d_k = -\hat{H}_k \nabla f(x_k)$, starting with x_o, and choosing α_k to minimize:

$$f[x_k - \alpha_k \hat{H}_k \nabla f(x_k)]$$

the sequence of points generated, x_k (k = 0,1,...,) is independent of the scalar β_k. An even more important result due to Dixon (1972) is that Huang's conclusion holds when f(x) is a general function.

3. Methods of Feasible Directions

Methods of feasible directions, called MFD's in the sequel, are credited to Zoutendijk (1960, 1970, 1974, and 1976) who developed much of their early theory.

3.1 Definitions

3.1.1. For problem (P):

$$\text{Minimize} \quad f(x)$$

$$\text{Subject to} \quad x \in S = \{x \in E^n \mid Ax < b\}$$

for all $x \in S$, a cone of feasible directions, $D(x)$, is defined by:

$$d \in D(x) \Leftrightarrow \lambda > 0, \quad \mu: \quad 0 < \mu < \lambda, \ (x + \mu d) \in S$$

3.1.2. A direction $d \in D(x)$ will be usable if a λ exists such that, for all μ with $0 < \mu < \lambda$, $f(x + \mu d) < f(x)$ will hold. As f is assumed differentiable, this is satisfied if $\nabla f(x)^t d < 0$.

3.1.3. The general approach to solve (P) using MFD's is as follows: Given $x_o \in S$ which is a feasible starting point, a sequence of points $x_k \in S$ $(k = 0,1,2,\ldots,)$ will be determined by:
First, finding a suitable feasible direction:

$$d_k \in \{D(x_k) \text{ and } [d \mid \nabla f(x_k)^t d < 0]\}$$

Next, the steplength λ_k will be determined in such a way that:

$$(x_k + \lambda_k d_k) \in S \quad \text{and} \quad f(x_k + \lambda_k d_k) < f(x_k)$$

Finally, setting: $x_{k+1} = x_k + \lambda_k d_k$

If $\bar{x}$ is an accumulation point of the sequence $\{x_k\}$ so that:

$$\lim_{k\to\infty} f(x_k) = f(\bar{x}),$$

then, under certain conditions, $\bar{x}$ is a local stationary point, that is a point satisfyig the Kuhn-Tucker first order necessary conditions for optimality.

Methods of feasible directions differ largely in the way the directions are chosen. In all methods, a direction finding problem is explicitly or implicitly solved in which a direction vector is found such that:

i) $d_k \in D(x_k)$

ii) $\nabla f(x_k)^t d < 0$

iii) And some additional requirements to guarantee and/or speed-up convergence.

For $\quad S = \{x \in E^n \mid Ax \leq b\}$

Letting $\quad I(x_k) = \{i \mid a_i^t x_k = b_i\}$

Then $\quad D(x_k) = \{d \mid a_i^t d \leq 0, \ i \in I(x_k)\} \qquad (3.1)$

So that the requirement that $d \in D(x_k)$ results in a set of homogeneous linear relations.

The steplength is formulated as a one dimensional minimization problem:

$$\text{Minimize } \{f(x_k + \lambda d_k) \mid x = x_k + \lambda d_k \in S\}$$

which can be solved either exactly or approximated.

3.2 Direction Finding Subproblem

There are three classes of direction-finding problems:

3.2.1. Direct Methods

Here a direction is implicitly determined through the original variables x_k. This is accomplished through a linear approximation of the objective function such as suggested by Frank and Wolfe (1956). At each step a linear subproblem is solved of the form:

$$\text{Minimize } \{\nabla f(x_k)^t x \mid Ax < b\} \qquad (3.2)$$

Given that $\bar{x}$ is the solution to (3.2), if:

$$\nabla f(\bar{x})^t (\bar{x} - x_k) = 0$$

the minimum for (P) has been achieved. If $\nabla f(x_k)^t (\bar{x} - x_k) < 0$, then $d_k = (x - x_k)$ will be usable feasible direction. A steplength is determined and a new point is defined.

A variant of this method in which, instead of solving (3.2), the simplex method is used to determine a vertex $\bar{x}$ such that $\nabla f(x_k)^t (\bar{x} - x_k) < 0$, is the convex simplex method of Zangwill (1969).

3.2.2. Optimization Methods

Here the direction finding problem is of the form:

$$\text{Minimize} \quad \{\nabla f(x_k)^t d \mid d \in D(x_k),\ d \in T,\ d \in N\} \tag{3.3}$$

where $D(x_k)$ is defined by (3.1) above, T is a set of linear relations to speed-up or guarantee convergence, and N is a set of relations to prevent infinite solutions.

N is usually taken to be the set: $\{d \mid \|d\| < 1\}$, giving the problem in the form:

$$\text{Minimize} \quad \{\nabla f(x_k)^t d \mid Ad < 0,\ Yd = 0,\ \|d\| < 1\}$$

with:

i) The L1 norm giving: $\sum_{i=1}^{n} |d_i| < 1$

ii) The L2 norm giving: $d^t d \leq 1$

iii) The L_∞ norm giving: $-1 < d_j < 1;\ j = 1,\ldots,n$

iv) The metric norm with P a symetric positive definite matrix giving: $d^t Pd < 1$.

Clearly, this leaves the choice for a variety of different methods. i) and iii) will result in linear programs; ii) and iv) will result in quadratic programs.

3.2.3. Feasibility Methods

Contrary to the methods in Sections 3.2.1. and 3.2.2. where the directions found had to be the locally best ones, there a direction is found which only has to satisfy the necessary requirements. That is d is such that:

i) $d \in D(x_k)$, $d \in T$

ii) $\nabla f(x_k)^t d < 0$ [for example: $\nabla f(x_k)^t d = -1$]

So the different methods in this class reduce to ways of solving systems of linear inequalities. The major methods suggested for that are:

i) Using the simplex method to determine the extreme directions leading to unboundedness and defining d_k appropriately.

ii) Using successive projections. That is, letting:

$$d_k = P_A \nabla f(x_k)$$

with $$P_A = [I - A^t(AA^t)^{-1}A]$$

This is done recursively as follows:

$$Q_o = I$$

$$Q_k = Q_{k-1} - \frac{Q_{k-1}a_k a_k^t Q_{k-1}}{a_k^t Q_{k-1} a_k} \; ; \; k = 1,\ldots,m$$

$P_A = Q_m$ (assuming A has m rows).

3.3 Discussion of Convergence of MFD's

Global convergence of methods of feasible directions has been studied extensively [see for example Zoutendijk (1960), Topkis and Veinott (1967), Polak (1971), Klessig (1974)]. It is shown that MFD's converge provided some antizigzagging provision is used to avoid jamming (nonfinite convergence or convergence to a non-stationary point) as illustrated by the now famous example of Wolfe (1972):

$$\text{Maximize} \quad \{-4/3\,(x_1^2 - x_1x_2 + x_2^2) - x_3\}$$

$$\text{Subject to} \quad x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0$$

The maximum clearly occurs at $\bar{x} = (0, 0, 0)^t$.

By choosing $\bar{x}_o = (a, 0, c)^t$ with

$$a < \sqrt{2}/4 \quad \text{and} \quad c > (1 + \sqrt{2}/2)\sqrt{a}$$

Wolfe shows that even with the projected gradient as direction, the sequence of points generated is of the form:

$$\lim_{k\to\infty} x_k = \begin{bmatrix} 0 \\ 0 \\ c - \sqrt{a}/2(1+1/\sqrt{2+1}/2+2/2\sqrt{2+\ldots}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c - \sqrt{a}(1+\sqrt{2}/2) \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Hence, due to zigzagging between the constraints $x_1 \geq 0$ and $x_2 \geq 0$, we could not reach the maximum but stopped at a non-stationary point.

Global convergence is guaranteed by allowing consideration of constraints that are almost-binding (also called ε-active constraints) in the computation of the search direction d_k.

Because MFD's degenerate to first-order gradient methods when the number of contraints is zero, these algorithms cannot converge better than linearly in the general case. However, linear convergence is not ensured by the behavior of a constrained optimization algorithm on unconstrained problems as was shown by Pironneau and Polak (1973). They showed that MFD's in general converge sublinearly. However, forcing the directions to the conjugate in a method of feasible directions can result in accelerating their convergence. This was suggested by Zoutendijk (1960, 1974) and was successfully applied to some special direction generators by Ritter (1975) and Best (1975) to obtain superlinearly convergent procedures. These methods will be discussed in more details in Section 4.

Finally, we note that a lot of other methods use feasible direction strategies but were not included in this class, such as projection methods (gradient projection and reduced-gradient methods). We included in this class only those methods of the Zoutendijk type or inspiration.

3.4. Relationship Between Projection Methods and Feasible Direction Methods

3.4.1. First-Order Methods

Methods of feasible directions involve the solution of a sequence of problems of the type:

$$\begin{aligned} &\text{Minimize} && \nabla f(x_k)^t \, (x - x_k) \\ &\text{Subject to} && \bar{A}(x - x_k) \leq 0 \\ &\text{And} && \| x - x_k \| \leq 1 \end{aligned} \tag{3.4}$$

where x_k if a feasible point.

The relationship of this problem to projection methods as described in Section one is stated in the following result to be found in Zoutendijk (1960) and Lemke (1961):

<u>Theorem 3</u>. If $(\bar{x} - x_k)$ solves the minimum problem (3.4) and $\nabla f(x_k)^t\ (\bar{x} - x_k) < 0$ then for $\mu > 0$, $\mu(\bar{x} - x_k)$ also solves the problem:

$$\begin{aligned} &\text{Minimize} && \|x - x_k\|^2 \\ &\text{Subject to} && \bar{A}(x - x_k) \leq 0 \qquad\qquad (3.5) \\ &\text{And} && -\nabla f(x_k)^t\ (x - x_k) \geq 1 \end{aligned}$$

The relationship can be made even more explicit by observing the following:

$$\|(x - x_k) + \nabla f(x_k)\|^2 = \|x - x_k\|^2 + \|\nabla f(x_k)\|^2 + 2\nabla f(x_k)^t\ (x - x_k)$$

from which, and using:

$$-\nabla f(x_k)^t(x - x_k) \geq 1$$

we have:

$$\|x - x_k\|^2 = \|[x - x_k + \nabla f(x_k)]\|^2 - \|\nabla f(x_k)\|^2 + 2$$

$$= \|x - [x_k - \nabla f(x_k)]\|^2 + \text{constant term.}$$

And therefore, the quadratic problem (3.5) in the theorem finds the projection of the unconstrained step $[x_k - \nabla f(x_k)]$ onto the set:

$$\{x \in E^n \mid Ax < b, -\nabla f(x_k)^t(x - x_k) > 1\}$$

where x_k is such that $\bar{A}x_k < b$.

3.4.2. Second-Order Methods

The second order feasible direction methods described by Polak (1971) can also be interpreted as projection methods since they solve quadratic subproblems.

4. Methods of Conjugate Directions

4.1. Definitions

4.1.1. Conjugate Directions

The concept of conjugate directions has played a key role in the development of unconstrained optimization techniques as pointed out by Fletcher (1972-b).

A set of n nonzero direction vectors $(d_1,\ldots,d_n)$, $d_i \in E^n$, are said to be mutually conjugate with respect to the nxn symetric matrix G if they are linearly independent and if:

$$d_i^t G d_j = 0 \quad i \neq j,\ i,\ j=1,\ldots,n \tag{4.1}$$

If we consider the problem of minimizing the quadratic function of n variables:

$$q(x) = a + b^t x + 1/2\ x^t G x$$

Then, given a set of n directions conjugate with respect to G and a starting point x_o, the location of the minimum of q(x), say $\bar{x}$, may be found in one of two ways:

i) By searching for the one-dimensional minimum of q(x) along each of the n directions in turn.

ii) By taking one Newton step using the fact that:

$$G^{-1} = \sum_{i=1}^{n} \frac{d_i d_i^t}{d_i^t G d_i} \tag{4.2}$$

where G is assumed to be positive definite.

Clearly, conjugate directions are really only well defined in the case of quadratic functions. The application of conjugate direction methods to the optimization of more general functions is based on the assumption that near the solution the function to be minimized will be nearly quadratic. As the algorithm proceeds, generating a sequence of points $\{x_k\}$ which converge to $\bar{x}$, the set of directions is constantly modified, producing a corresponding sequence of directions $(d_1^{(k)},\ldots,d_n^{(k)})$.

McCormick and Ritter (1972-a) were the first to formally extend the definition of conjugate directions to the case of general functions f(x), as having the following properties:

i) At each iteration k, the new point x_{k+1} is obtained as:

$x_{k+1} = x_k + \lambda_k d_k$ where λ_k is the smallest local minimizer to the one-dimensional step size problem:

$$\text{Minimize} \quad f(x_k + \lambda d_k)$$
$$\text{Subject to} \quad \lambda > 0$$

ii) If the sequence $\{x_k\}$ generated converges to a point $\bar{x}$ such that $\nabla f(\bar{x}) = 0$, $f(x)$ is twice continuously differentiable in a neighborhood of $\bar{x}$ and $G(\bar{x})$ is positive definite, then for all k:

$$\|d_k\| = O(\|g_k\|); \quad \|g_k\| = O(\|d_k\|) \qquad (4.3)$$

and for $k=1,\ldots,n-1$; $i=0,\ldots,k-1$

$$|d_{k+j}\, G(\bar{x}) d_{k+i}| = o(\|d_{k+\delta j}\| \, \|d_{k+1}\|) \qquad (4.4)$$

where $\delta = 0$ or 1.

For methods defined as above, McCormick and Ritter (1972-a) prove the following general result, which establishes an n or (n-1) step superlinear rate of convergence to an isolated local minimizer:

Theorem 4. Suppose the sequence $\{x_k\}$ is generated according to properties i) and ii) above. Suppose also that $G(\bar{x})$ is positive definite. Then, as $k\to\infty$

$$\frac{\|g_{k+n}\|}{\|g_{k+\delta}\|} \to 0 \quad \text{and} \quad \frac{\|x_{k+n} - \bar{x}\|}{\|x_{k+\delta} - \bar{x}\|} \to 0$$

where $\delta = 0$ or 1

We note that the relationship (4.4) is the most important defining relation because it ensures that an almost conjugate property holds in the sense that:

Given the sequence of directions $(d_1^{(k)}, \ldots, d_n^{(k)})$ produced by the algorithm, then:

$$\frac{d_i^{(k)} G(\bar{x}) d_j^{(k)}}{\|d_i^{(k)}\| \; \|d_j^{(k)}\|} \to 0 \quad \text{for } i \neq j;\ i,\ j = 1, \ldots, n$$

Also we note that the above theorem remains valid if inexact line searches are used, as long as the following holds:

$$\frac{|g_{k+1}^t \; d_k|}{\|g_{k+1}\| \; \|d_k\|} \to 0 \quad \text{as} \quad k \to \infty$$

4.1.2. Conjugate Gradients

In practice the objective function is not normally quadratic and the searach directions d_i are not known in advance. However, conjugate gradient algorithms are constructed so that if the function were quadratic then a subset of search directions would be conjugate.

In the early conjugate gradient algorithms by Hestenes (1952) and

Fletcher and Reeves (1964) it was assumed that the initial direction of search in each subset was along the direction of steepest descent:

$$d_1 = -\nabla f(x_o) = -g_o$$

Consecutive directions were built up iteratively by letting:

$$d_{k+1} = -g_k + \sum_{i=1}^{k} \beta_i d_i \tag{4.5}$$

and solving the conjugacy conditions (4.1):

$$d_{k+1}^t G d_i = 0; \; i=1,\ldots,k$$

to determine β_i.

If the line searches are exact and the function is quadratic, then this implies that:

$$\beta_j = 0; \; j < k \tag{4.6}$$

$$d_{k+1} = -g_k + \beta_k d_k \tag{4.7}$$

with

$$\beta_k = \frac{g_k^t(g_k - g_{k-1})}{d_k^t(g_k - g_{k-1})} \tag{4.8}$$

The method is efficient mainly due to (4.6) - (4.8) which is a great simplification over the more general formula (4.5). This makes the conjugate gradient method very attractive for large problems compared to Quasi-Newton methods because of its very low storage requirements.

Numerous ways of simplifying formula (4.8) have been proposed which are identical for quadratic functions and perfect line searches. Of these, the most well known are:

i) The Fletcher-and-Reeves (1964) formula:

$$\beta_k = [g_k^t g_k]/[g_{k-1}^t g_{k-1}] \tag{4.9}$$

ii) The Polak-Ribiere (1969) form:

$$\beta_k = [g_k^t (g_k - g_{k-1})]/[g_{k-1}^t g_{k-1}] \tag{4.10}$$

We note that for general functions there seems to be some numerical evidence, as pointed out by Powell (1977), in favor of the Polak-Ribiere form.

But in general, the most effective way to use conjugate gradients is with restarts along the steepest descent direction every n iterations. This makes the method effectively convergent. Also, the method is shown to be superlinearly convergent [see for example Cohen (1972), Dixon (1975-a)].

Most recent research in conjugate gradient algorithms has been in three main areas:

i) The relaxation of exact line searches as suggested by different authors: Kawamura and Volz (1973), Dixon (1975b), Nazareth (1977), and Mukai (1979).

ii) The improvement of restarting procedures. That is, instead of restarting automatically every n iterations, other possibilities such as checking the size of $|d_k^t g_k|$ might be more appropriate. This is a question studied by Powell (1977).

iii) The possibility of starting each cycle along directions other than the steepest descent one. Beale (1972) derived a simple conjugate gradient formula given any general starting direction. The corresponding results to (4.6) and (4.7) are:

$$\beta_j = 0; \; 2 < j < k \tag{4.11}$$

$$d_{k+1} = -g_k + \beta_1 d_1 + \beta_k d_k \tag{4.12}$$

where

$$\beta_1 = \frac{g_k^t(g_1 - g_o)}{d_1^t(g_1 - g_o)}, \text{ and } \beta_k \text{ is as in (4.8)} \tag{4.13}$$

The use of (4.12) allows the last direction of one conjugate subset to be used as the first direction of the next one.

An alternative way of allowing an arbitrary starting direction was introduced by Allwright (1972) and called "the preconditioned conjugate gradient" approach:

Supposing:

$$f(x) = 1/2x^tGx$$

If the metric (not necessarily equal to the Hessian G)

$$H = LL^t$$

is introduced, a new set of variables is defined as:

$$x = Lz$$

The problem could then be posed as:

$$\text{Minimize} \quad f(z) = 1/2z^tLGLz$$

for which the steepest descent at a point is:

$$\nabla_z f(x) = L^tGLz = -L^tg$$

which when transformed back into x-space becomes the direction:

$$d = -Hg$$

The equivalent of (4.6) and (4.7) for this method are given by:

$$d_{k+1} = -Hg_k + \beta_k d_k \tag{4.14}$$

where β_k corresponding to the Fletcher-Reeves formula (4.9) is given by:

$$\beta_k = \frac{g_k^t H g_k}{g_{k-1}^t H g_{k-1}} \tag{4.15}$$

Because of the success and the popularity of conjugate directions in solving unconstrained problems, some methods modifying them to solve linearly constrained problems have been developed.

4.2 Conjugate Direction Methods for LCP's

Three methods could be distinguished:

4.2.1. A Projection Method

Zoutendijk (1960) made the suggestion that his methods of feasible directions, which were discussed in section three, could be made to converge faster if one requires each newly constructed direction of search to be conjugate to an unspecified number of previous directions. This suggestion is based on the following observation:

$$\text{For} \quad f(x) = a + b^t x + 1/2 x^t G x \tag{4.16}$$

$$\text{If} \quad x_{k+1} = x_k + \lambda_k d_k$$

$$\text{Then} \quad g_{k+1} - g_k = \lambda_k G d_k \tag{4.17}$$

Hence, a new direction, d_{k+1}, would be required to be conjugate to d_k simply by imposing the following linear restriction:

$$(g_{k+1} - g_k)^t \, d_{k+1} = 0 = \lambda_k d_{k+1}^t G d_k \tag{4.18}$$

This approach was used by Ritter (1971), and McCormick and Ritter (1972-b) to develop a superlinearly convergent method for unconstrained minimization called the "projection method." Later this method was extended to handle linear constraints by Ritter (1975) and by Best and Ritter (1976). For the unconstrained method, a set of n search directions:

$$(d_1, \ldots, d_n) \equiv D$$

is maintained such that:

$$D^t = Y^{-1}$$

where Y is a matrix composed of the differences in the gradient of the objective function corresponding to steps taken along the directions in D. For the quadratic function (4.16):

$$Y = GD$$

so the requirement:

$$D^tY = I_{(nxn)}$$

is equivalent to:

$$D^tGD = I_{(nxn)}$$

Initially, $D = Y = I_{(nxn)}$. A typical iteration consists of choosing one direciton d_j; $1 \leq j \leq n$, from the columns of D and taking a step along that direction from the current point x_k to a point $x_{k+1} = x_k + \lambda_k d_j$ having a lower objective function value.

The difference in the gradient vector is computed:

$$\bar{\gamma}_j = \nabla f(x_{k+1}) - \nabla f(x_k)$$

and $\bar{\gamma}_j$ replaces γ_j, the jth column of the matrix Y. The corresponding update of D is:

$$\bar{d}_i = d_i^t[I_{(nxn)} - (\bar{\gamma}_j^t d_j)^{-1} \bar{\gamma}_j d_j^t]; \; i=1,\ldots,n; \; i \neq j$$

$$\bar{d}_j = d_j^t / d_j^t \bar{\gamma}_j$$

After n such steps, or more frequently if possible, an accelerating step is taken in the direction:

$$d^s = \sum_{j=1}^{n} \{d_j d_j^t\} \nabla f(x_k) \qquad (4.19)$$

As observed earlier, if $D^t GD = I_{(nxn)}$, then $G^{-1} = DD^t$, so the direction t is an approximation to a Newton step.

In the modification of this method for linear constraints, $\bar{A}$ whose columns are the normals to the active constraints, replaces some of the columns of Y. Thus, when applied to linearly constrained problems, Y is initialized to any nonsingular matrix whose first r columns are the columns of $\bar{A}$. Also, the accelerating step is taken after every (n-r) conjugate direction steps.

This method is essentially an active set strategy. When a new constraint is added to $\bar{A}$, one of the search directions, say d_ℓ is dropped and the other rows of D are updated as follows:

$$\bar{d}_j^t = d_j^t \; [I - (a^t d_\ell)^{-1} a d_\ell^t]$$

Theoretically then, the ℓth column of Y, γ_ℓ, is replaced by a. Essentially, this is a projection of the search direction d_j, and it is still true that:

$$D^t Y = I_{(rxr)} \text{ so that } \bar{D}^t GD = I_{(rxr)}$$

We also note that an anti-zigzagging provision is used to drop constraints from the active set. The method is shown to enjoy the following properties under appropriate assumptions:

i) $\{x_k\}$ converges to $\bar{x}$, where $\bar{x}$ is the unique minimize of f(x) over $F = \{x \in E^n \mid Ax \leq b\}$.

ii) If:

$$a_i^t\bar{x} = b_i;\ i=1,\ldots,r,$$

$$a_i^t\bar{x} < b_i;\ i=r+1,\ldots,m, \text{ and}$$

$$\nabla f(\bar{x}) + \sum_{i=1}^{r} u_i a_i = 0 \text{ with } u_i > 0;\ i=1,\ldots,r$$

that is, if the strict complementary slackness condition is satisfied, then there exists a k_o such that for $k \geq k_o$.

$$a_i^t x_k = b_i;\ i=1,\ldots,r,$$

$$a_i^t x_k < b_i;\ i=r+1,\ldots,m, \text{ and}$$

$$\frac{\| x_{k+n-r} - \bar{x} \|}{\| x_k - \bar{x} \|} \to 0 \quad \text{as } k \to \infty$$

4.2.2. A Feasible Direction Method

Best (1975) and Best and Ritter (1975) combined the conjugacy requirements defined by (4.18) with a particular feasible direction method of Zoutendijk to develop a superlinearly convergent procedure for problems with linear constraints.

Given a point x generated by the algorithm, a direction is found by solving the dual of the following linear program for x_{k+1}:

$$\begin{aligned} \text{Minimize} \quad & \nabla f(x_k)^t \, (x - x_k) \\ \text{Subject to} \quad & Ax \leq b \\ & Y_k(x - x_k) = 0 \end{aligned} \tag{4.20}$$

giving $d_k = x_{k+1} - x_k$

The matrix Y_k is formed by columns made up of differences of gradients of the objective function to ensure that the direction d_k is conjugate to (k-1) previously defined directions.

This method, combined with an inexact line search procedure and acceleration steps of the form (4.19) seems to be very attractive because of its linear direction finding subproblem. An extension of it could be sought where only near-binding constraints would be included in problem (4.20).

4.2.3. A Reduced Gradient Approach

This method is due to Shanno and Marsten (1979). It is designed to solve the problem:

$$\begin{aligned} \text{Minimize} \quad & f(x) \\ \text{Subject to} \quad & Ax = b \\ & l \leq x \leq u \end{aligned}$$

by combining the generalized reduced-gradient approach of Murtagh and

Saunders (1978) and a new conjugate gradient algorithm.

By partitioning the matrix A as:

$$Ax = [B|S|N]\begin{bmatrix} x_B \\ x_S \\ x_N \end{bmatrix} = b \tag{4.21}$$

where the basic variables x_B are used to satisfy the constraint set, the superbasic variables x_S are allowed to vary to minimize f(x), and the non-basic variables x_N are fixed at their bounds, the problem reduces to a sequence of unconstrained subproblems in terms of the superbasic variables x_S.

The new conjugate gradient method used to solve the unconstrained subproblems is modified to preserve information about good search directions when superbasic variables are dropped and when basis changes occur on manifolds. This method is called the "memoryless variable metric" method because it uses the following directions:

$$d_{k+1} = -Q_{k+1}g_{k+1} \tag{4.22}$$

where Q_{k+1} is the positive definite matrix defined by:

$$Q_{k+1} = I - \frac{p_k y_k^t + y_k p_k^t}{p_k^t y_k} + \left(1 + \frac{y_k^t y_k}{p_k^t y_k}\right) \frac{p_k p_k^t}{p_k^t y_k} \tag{4.23}$$

with $p_k = \lambda_k d_k$, $x_{k+1} = x_k + \lambda_k d_k$, $y_k = g_{k+1} - g_k$

The method is a conjugate gradient method because it is shown that if the line searches are exact, then (4.22) is equivalent to:

$$d_{k+1} = -g_{k+1} + \beta_k d_k \qquad (4.24)$$

where:

$$\beta_k = \frac{g_{k+1}^t (g_{k+1} - g_k)}{d_k^t (g_{k+1} - g_k)}$$

Also, inspecting (4.23) one notices that if I is replaced by Q_k, the BFGS Quasi-Newton updating formula is obtained.

The actual version used by Shanno and Marsten (1979) for the linearly constrained problem is refined so that it could be used without exact line searches and can be restarted along arbitrary search directions as in Beale's (1972) method. Howeve, a convergence analysis is not provided.

5. Curvilinear Methods

In this area work has been very limited except for the important study of Botsaris (1976, 1978, 1979). The essential idea in the method developed by Botsaris (1979) is to generate an arc along which movement is performed to decrease the objective function. The curvilinear search paths are obtained by solving a linear approximation of the differential equations of the continuous steepest descent curve for the objective function on the equality constrained region defined by the active constraints.

The idea of continuous steepest descent is based on the following consideration:

the change of $f(x)$ at $x = x_k$ along a space curve $x(t)$, $x(0) = x_k$ is given by:

$$df(x(t))dt\Big|_{t=0}$$

where t is a real parameter defined as the distance moved along $x(t)$.

If a maximal decrease in $f(x)$ at x_k is sought, then observing that:

$$\frac{df(x(t)}{dt} = \frac{df(x)}{dx}\ \frac{dx}{dt} = \nabla^t f(x)\ \dot{x}(t)$$

one will choose $\dot{x}(t) = -\nabla f(x)$, giving:

$$\frac{df(x(t))}{dt}\Bigg|_{t=0} = -\|\nabla f(x_k)\|^2$$

In the general case this gives rise to a nonlinear system of differential equations, the solution of which is the continuous steepest descent curve:

$$\dot{x}(t) = -\nabla f(x)$$

Using this basic result, together with a reduced gradient approach and the linear approximation of $\nabla f(x)$ at x_k:

$$\nabla f(x) = \nabla f(x_k) + G_k(x - x_k),$$

Botsaris (1979) determines a direction of move in the space of the nonbasic variables x_N, given by:

$$d_k^N = [e^{-t\Phi(x_k)} - I_{n-r}]\ \Phi^{-1}(x_k)\nabla\phi(x_k)$$

where, assuming that r variables are basic at iteration k, $\Phi(\cdot)$, $\Phi^{-1}(\cdot)$ and $\nabla\phi(\cdot)$ are the restricted (or projected) Hessian, Hessian inverse, and gradient to the space of the (n-r) independent variables.

The computational form of d_k^N require the solution of an eigensystem problem giving:

$$d_k^N = [\sum_{i=1}^{n-r} \frac{e^{-t\lambda_k^i} - 1}{\lambda_k^i}\ v_k^i(v_k^i)^t]\ \nabla\phi(x_k) \qquad (5.1)$$

where λ_k^i and v_k^i, i=1,...,n-r, are respectively the eigenvalue and eigenvectors of the restricted Hessian.

Clearly, this method has the major drawback of requiring the solution of an eigensystem problem at each iteration. However, from a theoretical point of view it has some very attractive properties:

i) For any kind of objective function, it is shown that:

$$\lim_{t\to\infty} x(t) = \bar{x} \text{ (KTP of the original problem)}$$

That is, starting from a feasible point and moving on the continuous steepest descent curve, the solution to the original problem is obtained asymptotically, hence global convergence.

ii) In the expression (5.1) for d_k^N, it is shown that the dominant term corresponds to the smallest eigenvalue of $\Phi(\cdot)$, which will cause a fast decrease in the objective function, clearly a desirable feature especially for nonconvex problems.

iii) The method provides a built-in switching between the steepest descent and Newton type directions as the algorithm proceeds. This is seen from the following observations:

as $t \to 0$, $e^{-t\Phi(x_k)} \simeq I_{n-r} - t\Phi(x_k)$

$$\text{and} \quad d_k^N = -t\nabla\phi(x_k)$$

as $t \to \infty$, and provided that $\Phi(x_k)$ is positive definite,

$$d_k^N = -\Phi^{-1}(x_k)\nabla\phi(x_k)$$

iv) Finally, the convergence rate of this method is shown to be quadratic under the assumption that the set of binding constraints remain unchanged after a certain number of iterations.

6. State of the Existing Software

The existing software for linearly constrained nonlinear programs is not really well documented except for two important studies by Lasdon and Warren (1978) and Murtagh and Saunders (1979), which are implementations of different versions of the Generalized Reduced Gradient (GRG).

Lasdon's version was developed initially for general nonlinear programs (GRG2) and Murtagh and Saunder's version was developed for large scale problems which are mostly linear (MINOS). A brief overview of these two softwares will be given here and further details can be found in the studies cited above plus Lasdon and Warren (1979). We note that a lot of general purpose codes also solve the linearly constrained problem as a special case, but our focus will be on codes that are designed specifically for that problem.

MINOS (Modular In Core Nonlinear Optimization System): This code is designed for large sparce systems. It uses a sparce LU factorization of the basis and a stable updating procedure for these factors. Either the BFGS variable metric method in factored form, or a conjugate gradient method is used to vary the superbasic variables. The system is designed to perform revised simplex iterations in case the initial solution is basic. Murtagh and Saunders (1979) report that problems with 700 variables, 49 being nonlinear, and 300 constraints were solved efficiently. The code also has many desirable input/output features that exist in commercial codes.

GRG2 (Generalized Reduced Gradient): As in MINOS, this code also uses a division of variables into basic, superbasic, and nonbasic. The superbasics are changed to decrease the objective function using either the BFGS variable metric method, in factored form, or the conjugate gradient mehtod modified to deal with bounds on the variables. The code is desaigned to switch from one method to the other if storage requirements warrant it. Lasdon and Warren (1979) report that a version of this code designed to exploit sparsity was under development.

Different versions of the GRG exist commercially which use the same approach but with different implementations of the reduced gradient; for example: GRGA by Abadie (1978), GRG73 by Heltue and Littschwager (1975), and Hamilton and Ragsdell (1982).

Other codes less publicized also solve linearly constrained problems. We now briefly review the most important of them. Table 2-1 below summarizes relevant information about these codes.

GPM (Gradient Projection Method): Rosen and Kreuser (1972) and Rosen and Wagner (1975) report about this code which solves problems of no more than 40 variables and 80 constraints. It uses Goldfarb's (1969) variable metric method to compute the search direction.

QRMNEW: May (1976, 1979). This code is based on an extension of Mifflin's (1975) local variations approximate Newton nonderivative method for unconstrained minimization. The direction vectors are generated from a QR factorization of the currently active constraint matrix in a similar way done by Gill and Murray (1974). The method is essentially a finite difference approximation method. The current version handles up to 45 variables and 80 constraints.

CGMAP (Conjugate Gradient Method of Approximation Programming): Beale (1969, 1974). This code is designed to solve large sparse problems which are mostly linear. It uses the SLP (successive linear programming) approach. It also uses an approach similar to MINOS and GRG2 to vary nonlinear variables acting as superbasics in a reduced gradient or a conjugate gradient direction. This code is reported to have solved less than 1% dense problems with more than 2000 constraints, about 4,000 linear variables and 400 nonlinear variables.

Finally, we note that the most extensive and highly integrated software library for linearly constrained problems to date was developed by the optimization group of the National Physical Laboratory in England and includes 12 different codes for solving the linearly constrained nonlinear problem. The relative performance of different codes has been studied by many authors, but unfortunately, there does not exist, to our knowledge, a unified study of linearly constrained codes. In the now classical study of Collevile (1970), 30 codes were tested on 8 problems, a subset of which were linearly constrained. This study found that reduced gradient and gradient projection-based codes performed better in terms of a standardized time: $t_s = t_r/t_t$ where t_r is the execution time of a code on a given computer, and t_t is the execution time of a run of a timing program. Two studies by Himmeblau (1972) and Newell and Himmeblau (1975) found that GRG and GPMNLC (a gradient projection method which includes any nonlinear constraints in a penalty function and solves the resulting linearly constrained problem) were superior to othe methods.

A more recent study by Saudgren and Ragsdell (1982) considered 17 codes and 30 test problems some of which were linearly constrained, and basically confirmed the overall superiority of the GRG based codes. The problems solved in this study, however were all small with less than 20 varialbes. An important study of software was made recently by Schittkowski (1980) but it mainly investigated general purpose nonlinear programming codes.

Lasdon and Warren (1979) report about three software systems for general NLP problems based on algorithms which solve recursively linearly constrained subproblems, as discussed in Section 3.1. These codes are OPRQP by Biggs (1976), GMP/NLC by Rosen (1977) and FCDPAK and ACDPAK by Best (1975) and Best and Ritter (1976).

Table 2-1. Summary of Existing Software (*)

Code Name	Author	Year	Method Used	Problem Size
MINOS	Murtagh Saunders	1979	Generalized Reduced Gradient	Not Fixed
GRG2	Lasdon Warren	1978	Generalized Reduced Gradient	Not Fixed
GPM	Rosen Krenser	1972	Gradient Projection	40 Variables 80 Constraints
QRMNEW	May	1979	Nonderivative Projected Newton	45 Variables 80 Constraints
CGMAP	Beale	1974	Successive Linear Approximation	Not Given
OPRQP	Biggs	1976	Successive Quadratic Programming	Not Fixed
GMP/NLP	Rosen	1977	Successive Linearly Constrained Problems	Not Given
FCDPAK ACDPAK	Best Ritter	1975, 1976	Successive Linearly Constrained Problems	Not Fixed

(*) This table was extracted from the study on nonlinear programming software made by Lasdon and Warren (1979).

CHAPTER III

A CONJUGATE DIRECTIONS ALGORITHM FOR THE LINEARLY CONSTRAINED NONLINEAR PROGRAM

1. Introduction

In this chapter we present an algorithm for solving the linearly constrained problem:

$$\text{(P):} \quad \text{Minimize} \quad f(x)$$
$$\text{Subject to} \quad a_i^t x - b_i < 0; \; i=1,\ldots,m$$

where f: $E^n \to E^1$ is a continuously differentiable function, $x \in E^n$, and $a_i \in E^n$, $i=1,\ldots,m$, and $b \in E^m$.

The chapter will be organized as follows: In Section 2 the algorithm will be presented. This will include a motivation and an outline of its different steps followed by the details of the algorithm and its flowchart. Some relevant properties will also be presented. In Section 3 we discuss the line search scheme and in Section 4 the direction finding subproblem. The details of the procedure to solve the direction problem will then be presented. The convergence analysis of the algorithm will be the subject of Chapter IV.

For reasons of clarity, we elect to present at this point a summary of the notation which will be used extensively throughout this study.

$x^k \in E^n$: is the kth approximation to the solution vector

$x_p^k \in E^n$: is the kth projection of an unconstrained point.

$x_u^k \in E^n$: is the kth approximation to the solution vector along an unconstrained direction vector.

$x^* \in E^n$: is a solution vector to the problem.

$d^k \in E^n$: is the kth constrained direction vector.

$d_u^k \in E^n$: is the kth unconstrained direction used to compute x_u^k.

$\nabla f(x^k)$, $g(x^k)$, $g_k \in E^n$: is the gradient vector of the objective function computed at x^k.

L_k: is the line segment joining x^k and x^{k+1}.

$\zeta^k \in E^n$: is a vector on the line segment L_k.

A: is the (m x n) dimensional array of constraint coefficients.

b: is the m-dimensional RHS vector of the constraints.

Y_k: is an (ℓ x n) dimensional array of coefficients of the conjugacy constraints: $\ell=1,\ldots, k-1$.

$e_k \in E^\ell$: is the right hand side of the conjugacy constraints; $(e_k)i$ represents the ith component of the vector e_k.

J_k: is a counter set keeping track of the number of consecutive conjugate directions constructed.

$y_k \in E^n$: is a vector representing the difference between the gradients of f(x) evaluated at x^k and x^{k-1}. This difference is normalized.

$u_k \in E^m$: is the kth approximation of the vector of Lagrange multipliers associated with the constraints of the problem, computed from the direction problem. $[u_k]_i$ represents the ith component of the dual vector u_k.

$v_k \in E^{\ell}$: is the vector of Lagrange multipliers associated with the conjugacy constraints of the kth direction-finding subproblem. $[v_k]_i$ represents the ith component of the dual vector v_k.

$G(x^k)$, G_k: is an (n x n) matrix representing the approximation of the Hessian of the objective function at iteration k.

E_k: is an (n x n) matrix representing the error in approximation between $G(x^k)$ and $G(x^*)$.

P_k: is an (n x n) projection operator at iteration k.

α_k, β_k, ν_k, w_k, σ_k, a_k and λ_k: are scalars used in the line search procedure.

$\|\cdot\|$: represents the Euclidean norm, unless otherwise specified.

$z \in E^n$: will denote a column vetor and z^t its transpose.

C, C(ε): represent global and local Lipschitz constants.

2. The Proposed Conjugate Directions Algorithm

In this section we present the conjugate directions algorithm we propose in this study. A detailed description of the method will be given before the algorithm itself is presented.

2.1 Description of the Method

Before proceeding with the description, we need to define the concept of conjugacy.

Definition 1. A set of directions d^i, i=1,...,k<n is said to be conjugate with respect to a matrix G and the associated quadratic function:

$$f(x) = a + b^t x + \frac{1}{2} x^t G x$$

if they are linearly independent and possess the property:

$$(d^i)^t G\, d^j = 0;\ i \neq j$$

It is clear from this definition that the concept of conjugacy is only defined with respect to a quadratic objective function. Its usefulness for optimization comes from the following property:

Property 1. Quadratic Termination (see for example Dixon (1980)). Given a strictly convex quadratic function and a set of directions d^i conjugate with respect to it, then starting at any point x_0 and undertaking perfect line searches along each direction d^i in turn, the minimum of the quadratic function will be achieved.

We note that in general the objective function is not quadratic and the search directions d^i, $i=1,\ldots,k<n$, are not known in advance. But we can define conjugacy in an "approximate" way; the approximation coming from the fact that we use the local quadratic approximation of the objective function.

To our knowledge, the only formal definition of this concept was given by McCormick and Ritter (1972a).

Definition 2. An algorithm to minimize a continuously differentiable f(x) is called a method of conjugate directions if the following properties hold:

i) At each iteration k, the new point x^{k+1} is obtained as:

$$x^{k+1} = x^k + \lambda_k d^k$$

where λ_k is the smallest local minimizer to the one-dimensional "step-size" problem:

$$\text{Minimize} \quad f(x^k + \lambda d^k)$$
$$\lambda > 0$$

ii) If the sequence $\{x^k\}$ converges to a point x^* such that $\nabla f(x^*) = 0$, $f(x)$ is twice continuously differentiable in a neighborhood of x^* and $G(x^*) = G$ is positive definite then, for all k,

$$\|d^k\| = O(\|\nabla f(x^k)\|), \ \|\nabla f(x^k)\| = O(\|d^k\|) \tag{2.1}$$

And for $\ell=1,\ldots,n-1$; $i=0,\ldots,\ell-1$

$$|(d^{k+\ell})^t \; G \; d^{k+i}| = o(\|d^{k+\mu\ell}\| \; \|d^{k+i}\|) \tag{2.2}$$

where $\mu = 0$ or 1.

iii) If in addition to the above assumptions, $G(x)$ satisfies a Lipschitz condition in a neighborhood of x^*, for all k and $\ell=1,\ldots,n-1$, $i=0,\ldots,\ell-1$

$$|(d^{k+\ell})^t G d^{k+i}| = o(\|d^{k+\mu\ell}\| \; \|d^k\| \; \|d^{k+1}\|) \qquad (2.3)$$

where $\mu = 0$ or 1.

Notes: The requirements given by (2.1) can be shown to be true for most methods where the direction vectors are computed using gradient information. Some examples are: steepest descent, conjugate gradient, and deflected gradient. Equations (2.2) and (2.3) will insure that for k large enough, for $\ell=1,\ldots,n-1$; $i=0,\ldots,\ell-1$.

$$\frac{(d^{k+\ell})^t G d^{k+i}}{\|d^{k+\ell}\| \; \|d^{k+i}\|} \to 0$$

With these definitions, we now give a general description of the algorithm.

The algorithm presented in this study is a conjugate directions method which uses a projection approach to determine the direction of movement. An initial step-size is used which is then adjusted by an Armijo-like scheme to insure enough decrease in the objective function. The algorithm perform two types of iterations.

i) A conjugate directions iteration during which an unconstrained step is projected onto the feasible region with the requirements of satisfying some approximate conjugacy restrictions. This will result in a feasible direction of descent. An inexact line-search is then performed along this direction to determine the next approximating point.

ii) A restarting iteration in which all the previous conjugate directions constructed are discarded and a new set is started.

The underlying iterative procedure can be summarized as follows:

$$x^{k+1} = x^k + \lambda_k | P_{[R]} (x^k + \alpha_k d_u^k) - x^k |$$

$$= x^k + \lambda_k (x_p^k - x^k) = x^k + \lambda_k d^k$$

where

$x_p^k = P_{[R]} (x^k + \alpha_k d_u^k)$ is the projection of the unconstrained step:

$x_u^k = x^k + \alpha_k d_u^k$ on the set $R_k = \{x \quad Ax - b < 0 \text{ and } Y_k x = e_k\}$

α_k = is the step size used along the unconstrained direction d_u^k.

At this point, it is enough to say that the only requirement on d_u^k is that it is a descent direction, (for example, $d_u^k = -\nabla f(x_k)$).

λ_k: is the adjusted step size; $\lambda_k \in (0,1]$

$Y_k x = e_k$: is a set of linear restriction enforcing the conjugacy requirements between the new direction and a finite number of previously constructed ones. This set is written in the form:

$$y_\ell^t \, x = (e_k)_\ell ; \quad \ell=1,\dots,k$$

To see that these constraints satisfy the approximate conjugacy property ii) of definition 2, we observe that:

$$y_\ell^t x = y_\ell^t x^k = (e_k)\ell \quad \text{or,}$$

$$y_\ell^t(x - x^k) = y_\ell^t \bar{d}, \; \bar{d} = d/\|d\|$$

Using the facts:

$$y_\ell = (g_\ell - g_{\ell-1})/\lambda_{\ell-1}\|d^{\ell-1}\|$$

$$(g_\ell - g_{\ell-1})^t d = \lambda_{\ell-1}(d^{\ell-1})^t \, G(\zeta^{\ell-1})d$$

where $\zeta^{\ell-1} \varepsilon L(x^\ell, x^{\ell-1})$, we have:

$$y_\ell^t \bar{d} = \frac{(d^{\ell-1})^t G(\zeta^{\ell-1})d}{\|d^{\ell-1}\| \, \|d\|}$$

which will tend to zero as $\zeta^{\ell-1} \to x^*$.

This is seen by observing the following: Since we impose $y_\ell^t \bar{d} = 0$ as an additional constraint, then for ℓ large enough, $\zeta^{\ell-1} \to x^*$ and:

$$\lim_{\ell\to\infty} y_\ell^t \bar{d} = \lim_{\ell\to\infty} \frac{(d^{\ell-1})^t G(\zeta^{\ell-1})d}{\|d^{\ell-1}\| \, \|d\|}$$

$$= \lim_{\ell\to\infty} \frac{(d^{\ell-1})^t G(x^*)d}{\|d^{\ell-1}\| \, \|d\|} = 0$$

so that d will be conjugate to $d^{\ell-1}$ with respect to the hessian of the objective function at x*.

2.2 The Proposed Algorithm

For solving the previously defined problem:

$$\text{Minimize} \quad f(x)$$
$$\text{Subject to} \quad Ax \leq b$$

the following algorithm, we call the "basic algorithm" for future reference, is proposed. But for the sake of clarity, we will first present an outline of the method, explaining what each step performs and the different parameters involved. We will then illustrate the workings of the procedure with a graphical example, and give the steps of the algorithm in compact form, followed by its flowchart (Figure 3.2).

Outline of the Method

At the start of the algorithm, an initial feasible point x^0 is given. If one is not available, a procedure like Phase I of the simplex method for linear programming, or linear complementary pivoting is used to obtain a point x^0 such that $Ax^0 \leq b$.

Also, to start the procedure, bounds on the step-size parameters need to be specified to insure global convergence as will be shown later on. These bounds are the following:

- For the unconstrained step that the algorithm takes at every iteration, the step size parameter α_k (k referring to the kth iteration) need only be bounded away from zero and finite, so that α_0, α_1: $0 < \alpha_0 < \alpha_1$ are chosen initially and $\alpha_k \varepsilon [\alpha_0, \alpha_1]$ for all k.

The unconstrained step is then projected onto the set $\{Ax < b, Y_k x = e_k\}$ made up of the feasible region and the conjugacy requirements. The result of this projection is a feasible conjugate direction of descent along which a step size is computed. This step-size has the form:

$$\lambda_k = (\beta_k)^{\nu_k} w_k, \text{ with } w_k = \min \{a_k, 1\}$$

where a_k is the initial approximation to the kth step-size. This initial approximation can be supplied by the user or taken to be equal to one. However, as k gets sufficiently large, a_k has to be an approximation to the exact step-size. So, in general, if $a > 0$ is chosen, then $a_k > a$ is all that is required. Also, from the definition of w_k, it is seen that the step-size will remain feasible.

To insure enough decrease in f, the scalar σ_k is required to be bounded away from zero and strictly less than one-half. For this, σ_0 and σ_1 with $0 < \sigma_0 < \sigma_1 < 1/2$ are initially specified and at each iteration σ_k need only be in the set $[\sigma_0, \sigma_1]$. For example if $\sigma_0 = \sigma_1 = 1/3$, then $\sigma_k = 1/3$ for all k will insure the feasibility of the line search.

Finally, $(\beta_k)^{\nu_k}$ represents a sequence of reduction factors applied to the initial feasible approximation w_k at each iteration. Here, ν_k, the "Armijo Number" is in the set $[0,1,2,....]$ so that the requirement on β_k is that it is non-zero and strictly less than one. To achieve that, we specify initially $\beta > 0$ and $\beta_k \in [\beta,1)$ for all iterations. Clearly, if $\beta_k = 1/2$ for all k, the reduction sequence will be of the form: $\{1, (1/2), (1/2)^2, (1/2)^3, \ldots\}$.

Furthermore, at the start of the algorithm, for the conjugacy requirements the matrices Y_0 and P_0 are initialized to zero and the identity matrix, respectively. In subsequent iterations, Y_k will contain as its rows the coefficients of the conjugacy constraints, and P_k will be the operator which projects any vector in E^n onto the set which satisfies conjugacy requirements. A counter J_0 is initialized to zero and is used to keep track of the number of conjugate directions constructed between restarts.

Now, at the start of a general iteration k, a feasible point x^k is given together with the matrices Y_k, P_k and the counter J_k. The step size parameters are then chosen within their required bounds: $\alpha_k \in [\alpha_0, \alpha_1]$ for the unconstrained step, and $\beta_k \in [\beta_0,1)$, $\sigma_k \in [\sigma_0,\sigma_1]$, and $a_k \in [a,\infty)$ for the constrained step. To determine the kth direction vector, d^k, the unconstrained point $x_u^k = x^k - \alpha_k \nabla f(x^k)$ is projected into the constrained point x_p^k such that:

$$x_p^k = \text{Arg}\min_x \{1/2 \, \| x - x_u^k \|^2 \mid Ax \leq b,\ Y_k x = e_k\}$$

resulting in $d^k = x_p^k - x^k$, and it is seen that when d^k is not zero, in which case $x_p^k \neq x^k$, we have:

$$Y_k x_p^k = e_k = Y_k x^k$$

or, equivalently:

$$Y_k(x_p^k - x^k) = Y_k d^k = 0$$

thus ensuring the conjugacy of d^k and d^{k-1}, d^{k-2}, etc...

Clearly, the above projection problem may not have a solution. This can happen for two reasons: the first reason is that no more conjugate directions can be constructed, thus resulting in the restart of the procedure and the construction of a new set of conjugate directions. This case is identified by checking if Y_k and J_k are not empty when $x_p^k = x^k$ (i.e., $d^k = 0$). The second reason is that a solution has been achieved and the procedure terminates with a Kuhn-Tucker point. This case is identified by checking if Y_k is empty and J_k is equal to zero when $x_p^k = x^k$.

To summarize, we observe that x_p^k, the solution to the projection problem satisfies the Kuhn-Tucker equation of the projection problem:

$$x_p^k - x^k + {}_k\,[\nabla f(x^k) + A^t \frac{u_k}{k}] + Y_k^t\, v_k = 0$$

from which we see that when $x_p^k = x^k$, a solution to the original problem is achieved only when $Y_k = 0$, otherwise, a restart is necessary. If on the other hand, $x_p^k \neq x^k$, then a conjugate direction has been constructed and the procedure continues.

When d^k is at hand, the inexact line search is used to compute a step size λ_k and the next approximation to the solution:

$$x^{k+1} = x^k + \lambda_k d^k$$

From x^{k+1}, to obtain the next conjugate direction vector d^{k+1}, we observe the following: For a direction vector d to be conjugate to d^k, it is necessary that the approximate conjugacy requirement:

$$\frac{(d^k)^t G(\zeta^k) d}{\|d^k\| \ \|d\|} = 0; \ \zeta^k \in L(x^{k+1}, x^k)$$

holds. But since:

$$\frac{(d^k)^t G(\zeta^k) d}{\|d^k\|} = \frac{(g_{k+1} - g_k)^t d}{\lambda_k \|d^k\|}$$

the above requirement is satisfied by imposing the condition:

$$\frac{(g_{k+1} - g_k)^t d}{\lambda_k \|d^k\|} = 0$$

or, equivalently, by letting $y_{k+1} = (g_{k+1} - g_k)/\lambda_k \|d^k\|$ and noting that $d = x - x^{k+1}$, we must have:

$$y_{k+1}^t d = y_{k+1}^t (x - x^{k+1}) = 0$$

That is:

$$y_{k+1}^t x = y_{k+1}^t x^{k+1} \triangleq (e)_{k+1}$$

Therefore, to compute d^{k+1} so that it is conjugate to d^k, the following linear restriction:

$$y_{k+1}^t x = (e)_{k+1}$$

is added before the (k+1)th projection problem is solved. The updating step of the algorithm adds y_{k+1}^t as a row vector to the matrix of conjugacy requirements Y_k to obtain Y_{k+1}, and J_k is updated to $J_{k+1} = (J_k) + 1$. We note here that d^{k+1} will be conjugate not only to d^k, as explained above, but also to d^{k-1}, d^{k-2}, etc..., since we are also improving the restrictions:

$$y_k x = e_k,\ y_{k-1}\ x = e_{k-1},\ \text{etc...}$$

whose coefficients are contained in Y_k.

The following numerical example is solved graphically illustrating the different steps of the proposed algorithm (see Figure 3.1).

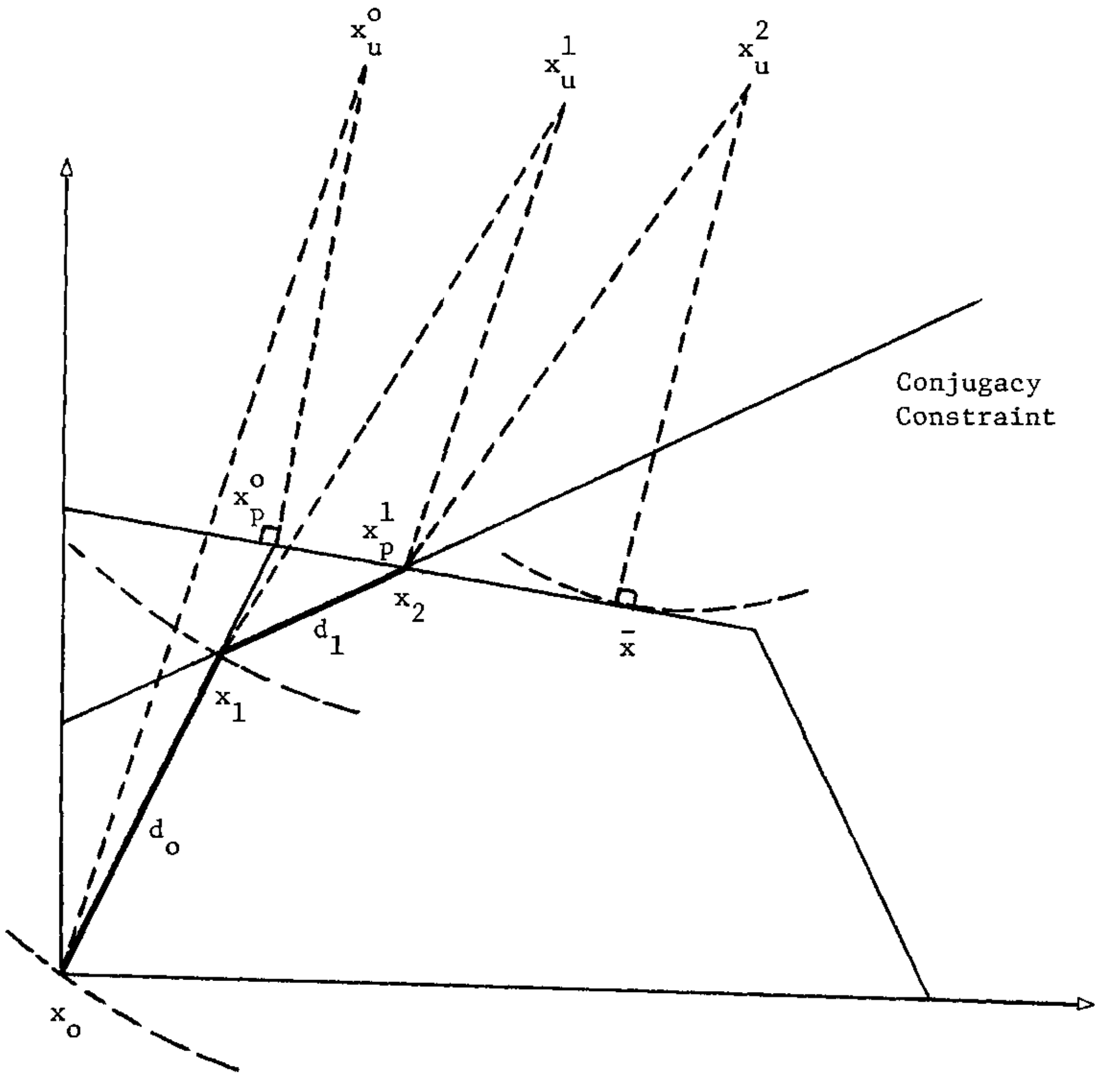

Figure 3-1: Graphical Illustration of the Proposed Method

Minimize $f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2$

Subject to $x_1 + x_2 \leq 2$

$x_1 + 5x_2 \leq 5$

$x_1, x_2 \geq 0$

The Algorithm

Step 0: Initialization

Given x^0; α_0, α_1: $0 < \alpha_0 \leq \alpha_1$; σ_0, σ_1: $0 < \sigma_0 \leq \sigma_1 < 1/2$;

$\beta > 0$, $a > 0$;

Let $J_0 = 0$ be a counter for the conjugacy constraints

Let $Y_0 = \phi$ be the initial matrix of conjugacy contraints

Let $P_0 = I$ be an initial projection matrix.

If x^0 is such that: $Ax^0 \leq b$, let $k = 1$, go to step 1.

Otherwise, replace x^0 by:

$$x^0 = \text{Arg} \{\text{Min: } \frac{1}{2} \| x - x^o \|^2 \quad | Ax \leq b\}$$

Let $k = 1$, go to step 1.

Step 1: Direction Finding

Given x^k: $Ax^k \leq b$, $\alpha_k \in [\alpha_0, \alpha_1]$, $\beta_k \in [\beta, 1)$, $\sigma_k \in [\sigma_0, \sigma_1]$, $a_k \in [a, \infty)$, Y_k, P_k, J_k and $x_u^k = x^k - \alpha_k \nabla f(x^k)$:

Let x_p^k be the solution to the following projection problem:

$$\text{PP:} \quad \text{Minimize} \quad \frac{1}{2}\|x - x_u^k\|^2$$

$$\text{Subject to} \quad Ax \leq b$$

$$Y_k x = e_k$$

where $e_k = Y_k x_k$. Go to step 2.

<u>Step 2: Restarting Step</u>

Given x_p^k, set $d^k = (x_p^k - x^k)$

Three possible cases may occur:

i) If $x_p^k \neq x^k$, go to step 3

ii) If $x_p^k = x^k$ and $Y_k = \phi$, stop with the a stationary point.

iii) If $x_p^k = x^k$ and $Y_k \neq 0$, restart: set $Y_k = 0$, $J_k = 0$, go to step 1.

<u>Step 3: Approximate Line Search</u>

Let $w_k = \text{Min}\,\{a_k, 1\}$, $\sigma_k \in [\sigma_0, \sigma_1]$

set $\lambda_k = (\beta_k)^{\nu_k} w_k$ where:

$$\nu_k = \text{Min}\,\Big\{_{\text{integer}}^{\nu \geq 0} \quad f|x^k + (\beta_k)^{\nu} w_k d^k| - f(x^k) \leq -\sigma_k (\beta_k)^{\nu} w_k \frac{\|d^k\|^2}{2\alpha_k}\Big\}$$

$$\text{set } x^{k+1} = x^k + \lambda_k (x_p^k - x^k) = x^k + \lambda_k d^k$$

Go to step 4.

Step 4: Updating Step

Let $y_{k+1} = (g_{k+1} - g_k)/\lambda_k \| x_p^k - x^k \|$

Set $Y_{k+1} = Y_k$ with row J_{k+1} replaced by the vector $(y_{k+1})^t$.

Set $J_{k+1} = J_k + 1$; $k \leftarrow k+1$; go to step 1.

Note 1: This algorithm, even though designed for constrained problems, can accomodate unconstrained problems as follows:

If $A = \phi$, in step 1, the solution to the projection problem is given in closed-form by:

$$x_p^k = x^k - \alpha_k P_k \nabla f(x^k)$$

where:

$$P_k = [I - Y_k^t (Y_k Y_k^t)^{-1} Y_k].$$

Figure 3.2 gives a flowchart for the "basic algorithm."

2.3 Properties of the Algorithm

We briefly present here some properties of the algorithm which are relevant for further analysis.

i) If x_p^k is the solution to the kth projection problem, then x_p^k is unique because the projection problem is strictly convex. Also, if $x_p^k \neq x^k$, then $d^k = (x_p^k - x^k)$ is a descent direction, as will be shown in lemma 2 (see global convergence analysis section in Chapter IV).

ii) The step size $\lambda_k = (\beta_k)^{\nu_k} w_k$ is well defined as will be shown in lemmas 3 and 4 (Chapter IV)

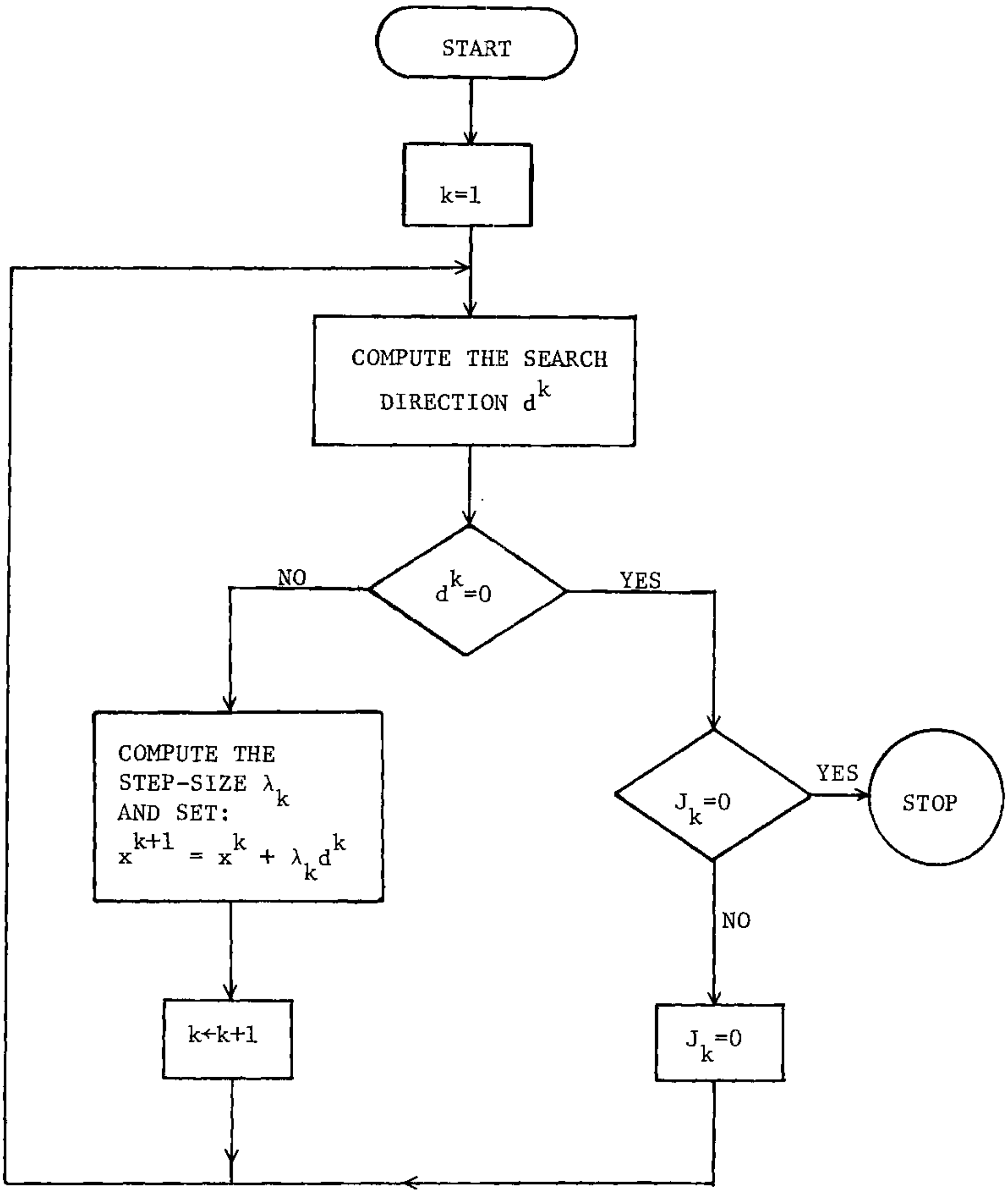

Figure 3-2: A Flowchart of the Proposed Method

iii) Let $M = \{k \mid Y_k = \phi,\ J_k = 0\}$

The set M is defined to be the set of restarting iterations.

Suppose $k \in M$ and $k+1, k+2, \ldots, k+q \notin M$, that is, the algorithm constructed q successive conjugate directions. Then from the optimality conditions of the projection problem, we have:

For the kth projection problem:

$$(x_p^k - x_u^k) + \sum_{i=1}^{m} [u_k]_i a_i = 0$$

$$a_i^t x_p^k - b_i \leq 0; \qquad i=1,\ldots,m$$

$$[u_k]_i \,(a_i^t x_p^k - b_i) = 0; \quad i=1,\ldots,m$$

$$[u_k]_i \geq 0; \qquad i=1,\ldots,m$$

For the (k+q)th projection problem:

$$(x_p^{k+q} - x_u^{k+q}) + \sum_{i=1}^{m} [u_{k+q}]_i \; a_i$$

$$+ \sum_{i=1}^{q} [v_{k+q}]_i \frac{[g_{k+i} - g_{k+i-1}]}{\lambda_{k+i-1} \| (x_p^{k+i-1} - x_{k+i-1}) \|} = 0$$

$$a_i^t x_p^{k+q} - b_i \leq 0; \qquad i=1,\ldots,m$$

$$(g_{k+i} - g_{k+i-1})^t x_p^{k+q} = (e_k)_i; \quad i=1,2,\ldots,q$$

$$[u_{k+1}]_i (a_i^t x_p^{k+q} - b_i) = 0; \qquad i=1,\ldots,m$$

$$[u_{k+q}]_i > 0; \qquad i=1,\ldots,m$$

$$[v_{k+q}]_i: \text{ Unrestricted}; \qquad i=1,\ldots,q$$

iv) The iterates: $x^{k+1} = x^k + \lambda_k(x_p^k - x^k)$ constructed by the algorithm will always be feasible to the constraints of the original problem. To see that we consider two cases that may occur:

Case 1: $\lambda_k = 1 \Rightarrow x^{k+1} = x_p^k$, therefore:

$$Ax^{k+1} = Ax_p^k < b$$

Case 2: $\lambda_k < 1$, then

$$x^{k+1} = x^k + \lambda_k x_p^k - \lambda_k x^k = (1 - \lambda_k)x^k + \lambda_k x_p^k$$

and

$$\begin{aligned} Ax^{k+1} &= (1 - \lambda_k)\, Ax^k + \lambda_k\, Ax_p^k \\ &< (1 - \lambda_k)\, b + \lambda_k\, b = b. \end{aligned}$$

3. The Line Search Scheme

In this section we explain how the line search procedure used in the algorithm works. It will be shown in Chapter IV that this procedure is well defined in the sense that it produces a step size in a finite number of iterations and a monotonic decrease in the objective function value. The procedure is based on the approximate search method of Armijo (1966) for unconstrained optimization. Since the solution of the projection problem will produce a descent direction satisfying:

$$\nabla f(x^k)^t \, d^k \leq - \frac{\|d^k\|^2}{2\alpha_k} < 0, \text{ for } d^k \neq 0, \tag{3.1}$$

then the step size is computed based on the linear prediction of the function around x^k:

$$f(x^k + \lambda d^k) \leq f(x^k) + \lambda \nabla f(x^k)^t d^k \tag{3.2}$$

This prediction will be used to insure enough decrease in f(x) for overall convergence.

The right hand side of inequality (3.2) above represents the tangent line to the curve $f(x^k + \lambda d^k)$ at $\lambda = 0$, as shown in Figure 3.3. Thus, for an arbitrary $\sigma_k \in (0,1)$, the line $f(x^k) - \sigma_k \lambda \|d^k\|^2/2\alpha_k$ will lie above the tangent line since:

$$f(x^k) + \lambda \nabla f(x^k)^t \, d^k < f(x^k) - \sigma_k \lambda \frac{\|d^k\|^2}{2\alpha_k}, \text{ for } \lambda > 0$$

and we have:

$$f(x^k + \lambda d^k) - f(x^k) < \sigma_k \lambda \nabla f(x^k)^t d^k < -\sigma_k \lambda \frac{\|d^k\|^2}{2\alpha_k} < 0$$

Thus the first λ_k to satisfy the inequality will produce enough decrease in $f(x)$.

Figure 3.3 illustrates how successive trials are computed: $\lambda_k^1 = (\beta_k^0 w_k)$, $\lambda_k^2 = (\beta_k^1 w_k)$, $\lambda_k^3 = (\beta_k^2 w_k)$ giving a successful trial with $\nu_k = 2$ and $\lambda_k = (\beta_k^2 w_k)$.

Finally, we note that bold-faced sets represent regions of acceptable step size. In addition, the choice of the line-search parameters will insure that a step-size bounded away from zero will be computed so that enough decrease in the objective function will be achieved away from a stationary point.

4. The Direction Finding Problem

In this section we discuss the properties of the direction problem and present a procedure for its solution based on Lemke's (1968) procedure for linear complementarity problems.

4.1 Properties of the Direction Finding Problem

Here we show that the direction vector can be obtained by solving a dual problem associated with the projection problem (PP).

Lemma 1. At every iteration, if the constraint set of the projection problem is consistent, the direction vector will be given by:

$$d^k = x_p^k - x^k = -\alpha_k P_k \nabla f(x^k) - P_k A^t u_k$$

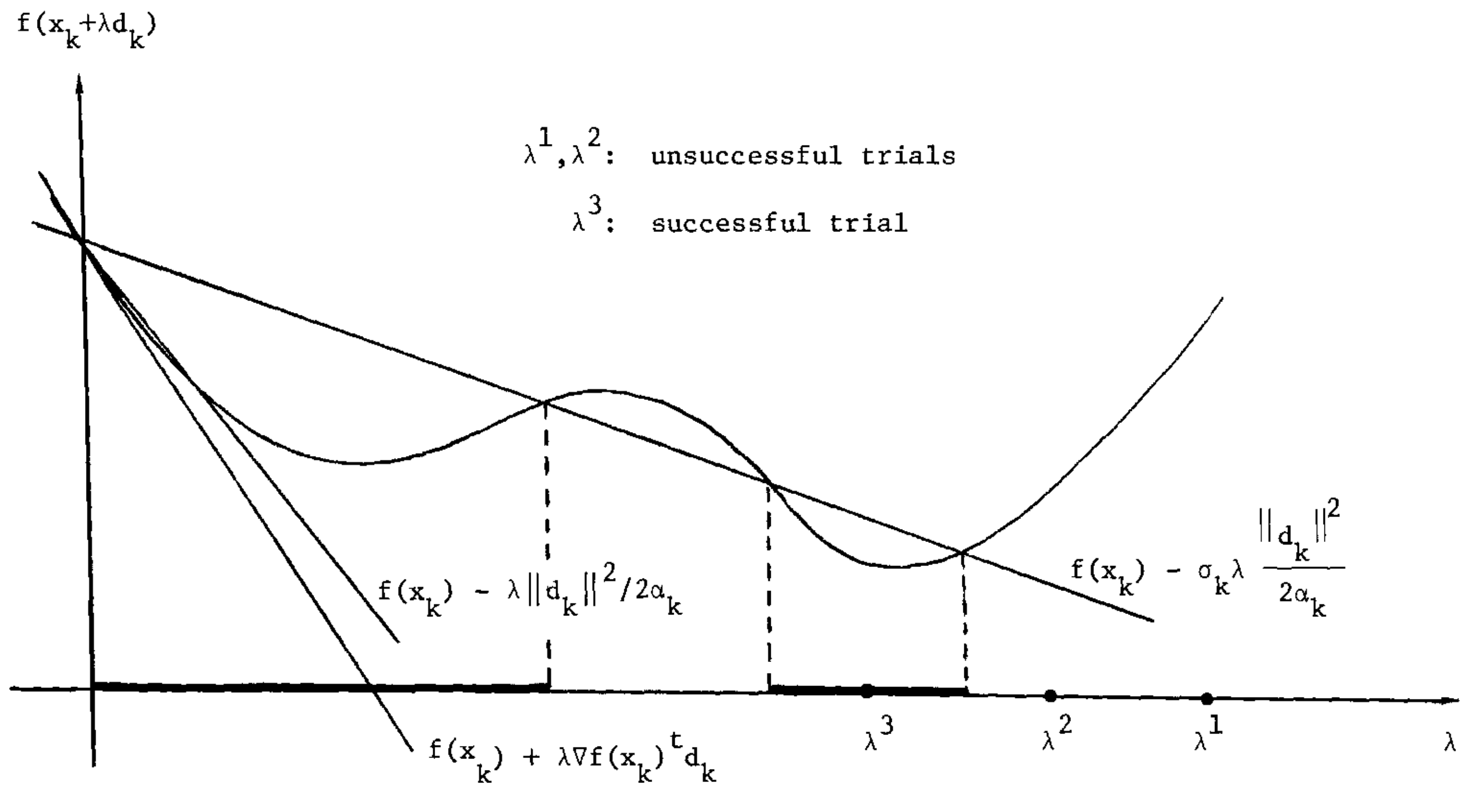

Figure 3-3: Step-Size Determination

where: $$P_k = [I - Y_k^t (Y_k Y_k^t)^{-1} Y_k]$$

and u_k is the vector of Kuhn-Tucker multipliers associated with the inequality constraints of the kth projection problem.

<u>Proof</u>. The projection problem (PP) is rewritten in the form:

$$\text{(PP)}': \quad \text{Minimize} \quad (-x_u^k)^t x + \frac{1}{2} x^t x$$
$$\text{Subject to} \quad Ax \le b$$
$$Y_k x = e_k$$

The dual associated with (PP)' is given by:

$$\text{Maximize} \quad -\frac{1}{2} x^t x - b^t u + e_k^t v \tag{4.1}$$

$$\text{Subject to} \quad x - x_u^k + A^t u + Y_k^t v = 0 \tag{4.2}$$

$$u \ge 0, \ v: \ \text{unrestricted.} \tag{4.3}$$

From (4.2) we derive

$$v = -(Y_k Y_k^t)^{-1} Y_k [\alpha_k \nabla f_k + A^t u]$$

using

$$x_u^k = x^k - \alpha_k \nabla f(x^k)$$

Upon substituting back for v in (4.2) we get:

$$x = x^k - [I - Y_k^t (Y_u Y_k)^{-1} Y_k] [\alpha_k \nabla f(x^k) + A_u^t] \qquad (4.4)$$

Also, substituting expression (4.4) for x in the dual problem we get the problem:

$$\text{Maximize} \quad -\frac{1}{2} [x^k - \alpha_k P_k \nabla f(x^k) - P_k A_u^t]^t [x^k - \alpha_k P_k \nabla f(x^k) - P_k A^t u]$$
$$- b^t u + e_k^t (Y_k Y_k^t)^{-1} Y_k [\alpha_k \nabla f(x^k) + A^t u]$$
$$\text{Subject to} \quad u \geq 0$$

Which is also equivalent to the problem:

$$\text{Minimize} \quad \frac{1}{2} u^t A P_k P_k A^t u + \alpha_k \nabla f(x^k)^t P_k A^t u + b^t u$$
$$- \alpha_k A^t u + \frac{1}{2} u^t A P_k P_k A^t u + \alpha_k \nabla f(x^k)^t P_k A u$$
$$+ u^t [b - A x^k]^t$$
$$\text{Subject to} \quad u \geq 0$$

After rearranging, we obtain the following dual problem:

$$\text{Minimize} \quad \frac{1}{2} u^t A P_k A^t + (s_k + \alpha_k A_k \nabla f(x^k)^t u \qquad (4.5)$$
$$\text{Subject to} \quad u \geq 0$$

where $s_k = (b - Ax^k)$ is the slack vector associated with the constraints of the original problem at x^k.

The solution of (4.5) will be u_k which when used in expression (4.4) gives:

$$d^k = (x_p^k - x^k) = -\alpha_k P_k \nabla f(x^k) - P_k A^t u_k$$

which completes the proof.

Lemma 2. Assuming that the projection problem (PP) is consistent, then it can be solved as a linear complementary problem.

Proof. Consider the projection problem (PP) at iteration k:

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\|x - x_u^k\|^2 \\ \text{Subject to} \quad & Ax \leq b \\ & Y_k x = e_k \end{aligned}$$

The Kuhn-Tucker conditions for x_p^k to be a solution to (PP) are:

$$-x_u^k + x_p^k + A^t u + Y_k^t v = 0 \tag{4.6}$$

$$Ax_p^k + y = b \tag{4.7}$$

$$Y_k x_p^k = e_k \tag{4.8}$$

$$u^t y = 0 \tag{4.9}$$

$$u \geq 0, \; y \geq 0$$

Multiplying (4.6) by Y_k and using (4.8) we get:

$$v = -(Y_k \, Y_k^t)^{-1} \, [-Y_k x_u^k + e_k + Y_k A^t u] \tag{4.10}$$

Now, multiplying (4.6) by A and using (4.7) and (4.10) we get:

$$-AP_k A^t u + y = -AP_k x_u^k + b + A(P_k - I) \, x_k \tag{4.11}$$

where: $$P_k = [I - Y_k^t \, (Y_k Y_k^t)^{-1} \, Y_k]$$

Using the fact that P_k is a projection operator such that: $P_k = P_k^t$ and $P_k P_k = P_k$, (4.11) is written as:

$$-AP_k P_k A^t u + y = -AP_k (x_u^k - x_k) + s_k$$

with $s_k = b - Ax^k$

Finally, by letting $x_u^k = x^k - \alpha_k \, \nabla f(x^k)$, the Kuhn-Tucker optimality conditions of the projection problem are equivalent to the following linear complementarity problem (LCP):

$$\text{LCP:} \quad \begin{aligned} &-AP_kA^tu + y = \alpha_k AP_k \nabla f(x^k) + s_k \\ &u^ty = 0 \\ &u \geq 0, \; y \geq 0 \end{aligned} \tag{4.12}$$

From Lemma 1, it is seen that the Kuhn-Tucker conditions of the dual problem (4.5) are:

$$\begin{aligned} AP_kA^tu + (s_k + \alpha_k AP_k \nabla f(x^k)) - \delta = 0 \\ u^t\delta = 0 \\ u \geq 0, \; \delta \geq 0 \end{aligned}$$

with $\delta = y$, the above system is exactly equal to (LCP).

Clearly, the solution (u_k, y_k) to (LCP) will produce the optimal solution to the projection problem:

$$x_p^k = x^k - \alpha_k P_k \nabla f(x^k) - P_k A^t u_k \tag{4.13}$$

from which $d^k = x_p^k - x^k$ is at hand.

Lemma 3. If the constraints to the projection problem are consistent, then a linear complementary pivoting algorithm will produce a complementary basic feasible solution in a finite number of steps.

Proof. Since AP_kA^t is positive semi-definite, it is copositive plus, and the result follows from known theory.

Notes:

1) If (4.13) above is written as:

$$x_p^k = x^k - \alpha_k P_k[\nabla f(x^k) + A^t \tilde{u}_k]$$

where $\tilde{u}_k = u_k/\alpha_k$. Then:

$$x^{k+1} = x^k + \lambda_k(x_p^k - x^k)$$

$$= x^k - \lambda_k \alpha_k P_k[\nabla f(x^k) + A^t \tilde{u}_k] \qquad (4.14)$$

Now, for comparison purposes with other projection methods, suppose that only the binding constraints are considered in our projection problem. Denoting by $\bar{A}_k$ the matrix of binding constraints at x^k, this problem becomes:

$$\begin{aligned} &\text{Minimize} && \frac{1}{2}\|x - x_u^k\|^2 \\ &\text{Subject to} && \bar{A}_k x = b \\ & && Y_k x = e_k \end{aligned} \qquad (4.15)$$

Its Kuhn-Tucker conditions are:

$$x_p^k - x^k + \alpha_k \nabla f(x^k) + \bar{A}_k^t w_k + y_k^t Y_k = 0$$

$$\bar{A}_k x_p^k = \bar{b} \qquad (4.16)$$

$$Y_k x_p^k = e_k$$

From (4.16) we obtain after substituting for v_k:

$$x_p^k = x^k - \alpha_k P_k [\nabla f(x^k) + \bar{A}_k^t \tilde{w}_k] \qquad (4.17)$$

where $\tilde{w}_k = w_k/\alpha_k$

Now, at nonoptimal points of problem (4.15), the vector of Kuhn-Tucker multipliers $\tilde{w}_k$, associated with $\bar{A}_k x = \bar{b}$, is the least-squares solution to the Kuhn-Tucker system (4.16). That is:

$$\tilde{w}_k = \text{Arg } \{\text{Min } \frac{1}{2} \| P_k (\nabla f(x^k) + \bar{A}_k^t \tilde{w}) \|^2\}$$

$$= -(\bar{A}_k P_k \bar{A}_k^t)^{-1} \bar{A}_k P_k \nabla f(x^k)$$

Upon substituting in (4.17), we get:

$$x_p^k = x^k - \alpha_k P_k [\nabla f(x^k) - \bar{A}_k^t (\bar{A}_k P_k \bar{A}_k^t)^{-1} \bar{A}_k P_k \nabla f(x^k)]$$

$$= x^k - \alpha_k [I - P_k \bar{A}_k^t (\bar{A}_k P_k \bar{A}_k^t)^{-1} \bar{A}_k] P_k \nabla f(x^k)$$

And since:

$$x^{k+1} = x^k + \lambda_k(x_p^k - x^k)$$

we get:

$$x^{k+1} = x^k - \alpha_k\lambda_k\{[I - P_k\bar{A}_k^t(\bar{A}_kP_k\bar{A}_k^t)^{-1}\ \bar{A}_k]P_k\}\ \nabla f(x^k)$$

From this last expression we observe two things:

i) α_k, the unconstrained step-size parameter can be interpreted as an approximation to the constrained step-size λ_k. Clearly, if α_k is chosen small enough, λ_k would be equal to one and the Armijo number ν_k would be equal to zero, thus requiring only one trial.

ii) The operator $\{I - P_k\bar{A}_k^t(\bar{A}_kP_k\bar{A}_k^t)^{-1}\ \bar{A}_k\}$ is seen to be a non-orthogonal projection operator weighted by P_k, which projects any direction vector in E^n onto the subspace defined by the approximate conjugacy requirements. In fact, this operator fills the gap between the Newton-type operator:

$$\{I - G_k^{-1}\bar{A}_k^t(\bar{A}_k\ G_k^{-1}\ \bar{A}_k^t)^{-1}\bar{A}_k\}$$

where G_k is either the Hessian matrix of the objective function at x^k or an approximation to it (Quasi-Newton type), and the orthogonal projection operator:

$$\{I - \bar{A}_k^t(\bar{A}_k\bar{A}_k^t)^{-1}\bar{A}_k\}$$

as in Rosen's (1966) method.

It is that weighting matrix P_k which will eventually insure the desired fast convergence beyond steepest descent type methods.

2) From the Kuhn-Tucker conditions of the projection problem:

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\|x - x_u^k\|^2 \\ \text{Subject to} \quad & Ax \leq b \\ & Y_k x = e_k \end{aligned}$$

and assuming that x_u^k is a general unconstrained point in a descent direction, that is, $x_u^k = x^k + \alpha_k d_u^k$ we have:

$$x_p^k - x^k - \alpha_k[-d_u^k + A^t\tilde{u}_k] + Y_k^t v_k = 0$$

$$Ax_p^k \leq b$$

$$\tilde{u}_k^t(Ax_p^k - b) = 0$$

$$Y_k x = e_k$$

$$u_k \geq 0$$

Now, from step 2 of the algorithm, if $x_p^k = x^k$ and $Y_k = 0$, it is not directly clear that the Kuhn-Tucker conditions of the original problem will be satisfied, since we will have:

$$-d_u^k + A^t\tilde{u}_k = 0$$

$$Ax_k \geq b$$

$$\tilde{u}_k(Ax_k - b) = 0$$

$$\tilde{u}_k \geq 0$$

If $d_u^k = -\nabla f(x^k)$, that is, the unconstrained step is taken along the steepest descent, the result is obvious.

If for example, $d_u^k = -\nabla f(x^k) + \beta_k d_u^{k-1}$, that is the unconstrained conjugate gradient direction is used, the Kuhn-Tucker conditions of the original problem will be satisfied only if $\beta_k = 0$, which corresponds to a restart point for the conjugate gradient iterations. This means that each time the matrix Y_k is reinitialized, corresponding to a restarting point of the constrained algorithm, the unconstrained step has to be taken along the negative gradient to insure convergence of the sequence of restarting points to a Kuhn-Tucker point of the original problem.

4.2 Solution of the Projection Problem

4.2.1 Forumulation as a Complementary Problem

The complementary pivoting method of Lemke (1968) for solving the system of dimension m:

$$-Mu + y = q$$

$$y^t u = 0$$

$$u \geq 0$$

$$v \geq 0$$

will be used in this study. We give here a brief description of this method. If $q \geq 0$, a solution is readily available. If $q \not\geq 0$, by introducing an artificial variable z_0, the above system reduces to the following:

$$-Mu + y - 1\, z_0 = q$$
$$y^t u = 0$$
$$u,\ y,\ z \geq 0$$

with 1 denoting an m-dimensional vector whose elements are equal to one. A solution to the above system is now available by letting $z_0 = \text{Max}\{-q_i:\ i=1,\ldots,m\} = q_r$, $u_j = 0$ and $y_j = q_j + 1_j\, z_0$, for $j=1,\ldots,m$, $j \neq r$; and only one variable from each complementary pair (u_i, y_i) is basic. Lemke's algorithm attempts to drive z_0 out of the basis while maintaining the property that exactly y_i or u_i is basic for (m-1) components. When z_0 leaves the basis, a solution to the complementary problem is at hand.

With M copositive plus, Lemke's procedure is finite assuming that degeneracy is not present or, if it is, it can be removed by a lexicographical method as in the studies by Eaves (1971) and Pann (1974).

For the projection problem in this study the following is a summary of the steps taken to set up the complementary problem:

0. Given x^k, $\nabla f(x^k)$, P_k, and α_k:

1. Solve the problem for z_p^k:

$$\left.\begin{array}{ll} \text{Minimize} & \frac{1}{2}\|z - x_u^k\|^2 \\ \\ \text{Subject to} & Y_k z = e_k \end{array}\right\} \Rightarrow z_p^k = x^k - \alpha_k P_k \nabla f(x^k)$$

2. Let $M = AP_k A^t$

And $q = b - Az_p^k$

3. Given (u_k, y_k) a basic feasible complementary solution obtained by solving (LCP), then:

$$d^k = z_p^k - x^k - P_k A^t u_k$$

To see why d^k is given by this last expression, we note the following:

Given $z_p^k = x^k - \alpha_k P_k \nabla f(x^k)$ from Step 1 above, then

$$q = b - Az_p^k = b - A[x^k - \alpha_k P_k \nabla f(x^k)]$$

$$= b - Ax^k + \alpha_k AP_k \nabla f(x^k)$$

$$= s_k + \alpha_k AP_k \nabla f(x^k)$$

From (4.13) in proposition 2 and the above, we have:

$$d^k = x_p^k - x^k = x^k - \alpha_k P_k \nabla f(x^k) - P_k A^t u_k - x^k$$

$$= z_p^k - x^k - P_k A^t u_k.$$

The above forms of q, z_p^k and d^k will be used for computational purposes as they are clearly more efficient forms.

4.2.2 Consideration of Bounds and Equality Constraints

When the original problem includes equality constraints, the natural way to handle them is to initialize the projection matrix P to:

$$P_0 = [I - D^t(DD^t)^{-1} D] \qquad (4.18)$$

where D is the coefficient matrix of the equality constraints. During subsequent iterations, P_k is updated by the following formula:

$$P_{k+1} = P_k - \frac{P_k y_k y_k^t P_k}{y_k^t P_k y_k}$$

where $y_k = (g_k - g_{k-1})/\|\lambda_{k-1} d^{k-1}\|$ represents the vector associated with a new conjugacy constraint. On restart P_0, given by (4.18), is used.

When explicit lower and upper bounds are present, the projection problem will have the form:

$$\text{Minimize} \quad \frac{1}{2}\|x - x_u^k\|^2$$

$$\text{Subject to} \quad Ax \leq b$$

$$Ix \leq u$$

$$-Ix \leq -\ell$$

$$Y_k x = e_k$$

In this case, the linear complementary problem will have the form:

$$\begin{bmatrix} A \\ \overline{I} \\ -\overline{I} \end{bmatrix} [P_k] \; [A^t \;\; I \;\; -I] \; u + y = s_k + \alpha_k \begin{bmatrix} A \\ \overline{I} \\ -\overline{I} \end{bmatrix} [P_k] \; \nabla f(x^k)$$

$$u^t y = 0; \; u \geq 0, \; y \geq 0$$

In Table 3-1 we summarize the computational forms that M and q can take depending on the particular problem under consideration, and the associated direction vector.

Table 3-1: Lemke's Arrays and Corresponding Direction Vectors			
PROBLEM TYPE	LEMKE'S ARRAY $-M$	LEMKE'S RHS q	ASSOCIATED DIRECTION d^k
$A\neq\phi$ $u\neq\phi$ $\ell\neq\phi$	$-AP_kA^t \mid -AP_k \mid AP_k$ $-P_kA^t \mid -P_k \mid P_k$ $P_kA^t \mid P_k \mid -P_k$	$b - Ax_p^k$ $u - x_p^k$ $x_p^k - \ell$	$z_p^k - x^k - [P_kA^t \mid P_k \mid -P_k]u_k$
$A\neq\phi$ $u\neq\phi$ $\ell=\phi$	$-AP_kA^t \mid -AP_k$ $-P_kA^t \mid -P_k$	$b - Ax_p^k$ $u - x_p^k$	$z_p^k - x^k - [P_kA^t \mid P_k]u_k$
$A\neq\phi$ $u=\phi$ $\ell\neq\phi$	$-AP_kA^t \mid AP_k$ $P_kA^t \mid -P_k$	$b - Ax_p^k$ $x_p^k - \ell$	$z_p^k - x^k - [P_kA^t \mid -P_k]u_k$
$A\neq\phi$ $u=\ell=\phi$	$[-AP_kA^t]$	$[b - Ax_p^k]$	$z_p^k - x^k - [P_kA^t]u_k$
$A=\phi$ $u\neq\phi$ $\ell\neq\phi$	$-P_k \mid P_k$ $P_k \mid -P_k$	$u - x_p^k$ $x_p - \ell$	$z_p^k - x^k - [P_k \mid -P_k]u_k$
$A=\phi$ $u\neq\phi$ $\ell=\phi$	$[-P_k]$	$[u - x_p^k]$	$z_p^k - x^k - P_ku_k$
$A=\phi$ $u=\phi$ $\ell\neq\phi$	$[-P_k]$	$[x_p^k - \ell]$	$z_p^k - x^k + P_ku_k$

CHAPTER IV

GLOBAL CONVERGENCE ANALYSIS

1. Introduction

In this chapter and the next one the convergence properties of the conjugate directions algorithm presented in Chapter III will be studied. We recall that the algorithm in question performs two major steps. The first is a conjugate direction step which is accomplished by solving a projection problem followed by an inexact line search. The second is a restarting step which initiates the construction of a fresh set of conjugate directions. It will be shown subsequently that these two steps are at the heart of the convergence behavior of the algorithm. Starting from any feasible point, a sequence of approximation is generated which convergences to a Kuhn-Tucker Point. When a solution point is approached, this sequence exhibits a second order rate of convergence.

The main section of this chapter will be devoted to the global converge analysis of the algorithm. Before we start the analysis we need the following definitions which will be used extensively throughout the chapter.

$$\text{Let } \; R_k = \{x \mid Ax < b,\ Y_k x = e_k\},\ S = \{x \mid Ax < b\}.$$

Let $x^k \in S$ and $x_p^k \in R_k$ the projection of the unconstrained point

$x_u^k = x^k - \alpha_k \nabla f(x^k)$ onto R_k. Let x^* be a stationary point and u^* the vector of Kuhn-Tucker multipliers associated with the inequality constraints at x^*. In addition, let r be the number of binding constraints at x^*, and correspondingly $\bar{A}x^* = \bar{b}$. Finally, we let $M = \{k \mid Y_k = 0, J_k = 0\}$ be the set of restarting iterations. The gradient of the objective function at x^k will be denoted equivalently by $\nabla f(x^k)$, $g(x^k)$, and g_k.

2. Global Convergence Analysis

The main result of this section will be to show that each accumulation point of the sequence generated by the algorithm is a stationary point. In addition, under appropriate assumptions, it will be shown that the whole sequence will converge.

To accomplish this, some intermediate results will be needed which we organize in four parts. The first part will be concerned with properties of the direction vector. More specifically, it will be shown that:

i) The direction vector is the unique optimal solution of the projection problem (lemma 1).

ii) The direction vector is a feasible direction of descent (lemma 2).

iii) Near a point which is not a stationary point, the direction vector is bounded above and below away from zero (lemma 3).

The second part will be concerned with the step-size properties. Here it will be shown that:

i) Under the assumption of twice continuous differentiability of f, the step-size is uniformly bounded (lemma 4).

ii) Near a point which is not a stationary point the result in i) will be sharpened by doing away with the twice continuous differentiability assumption (lemma 5).

The third part will be concerned with the convergence properties of the subsequence of restarting points, $\{x^k, k \in M\}$. In particular, it will be shown that:

i) Each accumulation point of this subsequence is a stationary point and that the differences in the tail of the subsequence will converge to zero (theorem 1).

ii) The entire subsequence will converge to a single stationary point based on the results of i) above and the assumption that the subsequence has a finite number of accumulation points (theorem 2).

iii) Under the stronger assumption of convexity of f, the subsequence will converge to the unique solution of the problem (theorem 3).

Finally, the last part will be concerned with showing that the sequence of all points (restarting and conjugate steps) is a convergent sequence (theorem 4).

2.1 Properties of the Direction Vector

We start by showing that x_p^k is the unique optimal solution to the kth projection problem, thus establishing uniqueness of the direction vector. We note that this turns out to be a variation of a standard result characterizing the minimum distance between a closed convex set and a point not in the set (see for example Bazaraa and Shetty (1979)).

<u>Lemma 1</u>. Given $x^k \varepsilon S$, $x_u^k \varepsilon E^n$ and

$$x_p^k = \text{Arg } \{\text{Min } \frac{1}{2} \| x - x_u^k \|^2 \mid x \varepsilon R_k\}$$

Then the relationship:

$$(x_p^k - y)^t (x_p^k - x_u^k) < 0 \qquad (2.1)$$

holds for any vector $y \varepsilon R_k$.

<u>Proof</u>. Let $h(x) = 1/2 \| x - x_u^k \|^2$. Then $x_p^k = \text{Arg}\{\text{Min } h(x) \mid x \varepsilon R_k\}$. Since h(x) is strictly convex, the necessary and sufficient conditions for a minimum of h over R_k are:

$$\nabla h(x_p^k)^t (y - x_p^k) > 0, \qquad \text{for all } y \varepsilon R_k$$

This is equivalent to:

$$(x_p^k - x_u^k)^t (y - x_p^k) > 0, \quad \text{for all } y \varepsilon R_k$$

Thus, the desired result:

$$(x_p^k - y)^t (x_p^k - x_u^k) < 0, \quad \text{for all } y \varepsilon R_k$$

Now, we show uniqueness of x_p^k. Using (2.1), we have:

$$\|x_u^k - y\|^2 = \|x_u^k - x_p^k\|^2 + \|x_p^k - y\|^2 + 2(x_p^k - y)^t\ (x_u^k - x_p^k)$$

which implies that:

$$\|x_u^k - y\|^2 > \|x_u^k - x_p^k\|^2 \quad \text{for all } y \in R_k$$

In particular let $y = x_p^{k'}$ where $x_p^{k'}$ is such that:

$$(x_u^k - x_p^{k'}) = (x_p^k - x_u^k)$$

Then, using (2.1) we have:

$$\|x_p^k - x_p^{k'}\|^2 + 2(x_p^k - x_p^{k'})^t(x_u^k - x_p^k) = 0$$

or, equivalently:

$$(x_p^k - x_p^{k'})^t\ [(x_p^k - x_p^{k'}) + 2(x_u^k - x_p^k)] = 0$$

then, either:

$$x_p^k - x_p^{k'} = 0 \quad \Rightarrow \quad x_p^k \text{ is unique, or:}$$

$$(x_p^k - x_p^{k'}) + 2(x_u^k - x_p^k) = 0$$

which implies that:

$$2(x_u^k - x_p^k) = 0 \quad \text{and hence} \quad x_u^k = x_p^{k'} - x_p^k$$

and again x_p^k is unique. This completes the proof.

We note that since $d^k = (x_p^k - x^k)$, it is the unique optimal solution. The next result shows that the direction vector produced by solving the projection problem is a direction of descent.

Lemma 2. Given $x^k \in S$, $x_u^k \in E^n$ and x_p^k as in lemma 1 the following properties hold:

i) $\alpha_k \nabla f(x^k)^t(y - x_p^k) > (x_p^k - x^k)^t(x_p^k - y)$, for all $y \in R_k$

ii) The direction vector $(x_p^k - x_k)$ produced by the algorithm is a direction of descent.

Proof. i) From lemma 1 we have:

$$(x_u^k - x_p^k)^t(y - x_p^k) < 0, \text{ for all } y \in R_k$$

Substituting for x_u^k, we have:

$$(x^k - \alpha_k\nabla f(x_k) - x^k) < \alpha_k\nabla f(x^k)^t(y - x_p^k), \text{ for all } y \in R_k$$

from which:

$$(x^k - x_p^k)^t(y - x_p^k) < \alpha_k\nabla f(x^k)^t(y - x_p^k), \text{ for all } y \in R_k \qquad (2.2)$$

ii) The direction vector is produced by solving the following problem for x_p^k:

$$\text{Minimize} \quad \frac{1}{2}\|x - x_u^k\|^2 = h(x)$$

$$\text{Subject to} \quad Ax \leq b$$

$$Y_k x = e_k$$

If $x = x^k$, the constraints are satisfied and

$$h(x^k) = \frac{1}{2}\|x^k - x_u^k\|^2 = \frac{1}{2}\|x^k - x^k + \alpha_k \nabla f(x^k)\|^2 = \frac{1}{2}\alpha_k^2 \|\nabla f(x^k)\|^2$$

Therefore, we must have:

$$\|x_p^k - x_u^k\|^2 \leq \alpha_k^2 \|\nabla f(x^k)\|^2$$

Expanding the left hand side, we get:

$$(x_p^k - x^k + \alpha_k \nabla f(x^k))^t (x_p^k - x^k + \alpha_k \nabla f(x^k)) = (x_p^k - x^k)^t (x_p^k - x^k)$$

$$+ 2\alpha_k \nabla f(x^k)^t (x_p^k - x^k) + \alpha_k^2 \|\nabla f(x^k)\|^2 \leq \alpha_k^2 \|\nabla f(x^k)\|^2$$

from which:

$$\|x_p^k - x^k\|^2 + 2\alpha_k \nabla f(x^k)^t (x_p^k - x^k) \leq 0$$

$$\nabla f(x^k)^t (x_p^k - x^k) \leq - \frac{\|x_p^k - x^k\|^2}{2\alpha_k} \tag{2.3}$$

Hence, $$\nabla f(x^k)^t d^k \leq - \frac{\|d^k\|^2}{2\alpha_k} < 0$$

which implies that $d^k = (x_p^k - x^k)$ is a descent direction at nonoptimal points since $\alpha_k > 0$ for all k.

The next lemma shows that near a point $\bar{x}$ which is not a stationary point, the direction vector is bounded above and below away from zero.

Lemma 3. Let f be continuously differentiable. In addition, let S be bounded and suppose that $\bar{x} \in S$ is not a stationary point. Then there is a neighborhood N of $\bar{x}$ and positive numbers c_1 and c_2 so that for $x^k \in N \cap S$, the direction d^k produced by the algorithm satisfies $c_1 \leq \|d^k\| \leq c_2$.

Proof. By definition $\|d^k\| = \|x_p^k - x^k\|$. Since x_p^k, $x^k \in S$ and S is bounded, then $\|d^k\|$ is bounded above. Now, to show that $\|d^k\|$ is bounded below away from zero, we first consider the case of $k \in M$.

Let $$h(x^k) = \{\text{Min } \frac{1}{2} \|x - x_u^k\|^2 \mid Ax \leq b\}$$

Since $x = x^k$ is feasible, then $h(x^k) \leq \frac{1}{2} \alpha_k^2 \|\nabla f(x^k)\|^2$. By contradiction, suppose that $\|d^k\|$ is not bounded away from zeo in a neighborhood of $\bar{x}$. Then there is a sequence $\{x^k\} \in S$ converging to $\bar{x}$ and a sequence $\{\alpha_k\}$ converging to $\bar{\alpha}$ such that:

$$h(x^k) \to \frac{1}{2} \bar{\alpha}^2 \|\nabla f(\bar{x})\|^2 \qquad (2.4)$$

since $h(x^k) = \frac{1}{2} \|(x_p^k - x^k) + \alpha_k \nabla f(x^k)\|$ and the gradient is continuous.

By continuity of optimal solutions of positive definite quadratic programs (Daniel (1973)) it follows that:

$$h(x^k) \to h(\bar{x}) < \frac{1}{2} \bar{\alpha}^2 \|\nabla f(\bar{x})\|^2 \qquad (2.5)$$

From (2.4) and (2.5) we have:

$$h(\bar{x}) = \frac{1}{2} \bar{\alpha}^2 \|\nabla f(\bar{x})\|^2$$

which implies that $\bar{x}_p = \bar{x}$ and hence $\bar{d} = 0$, therefore $\bar{x}$ is a stationary point contradicting the assumption that it is not. This shows that for each $s > 0$ there are numbers c_1, c_2 with $0 < c_1 < c_2$ such that $\|x^k - \bar{x}\| < s$ and $c_1 < \|d^k\| < c_2$.

2.2 Uniform Boundedness of the Step-Size

In this part it will be shown that the step-size computed by the algorithm is bounded above and below away from zero. This is very important for the finiteness of the procedure. Close to a point $\bar{x}$ which is not a stationary point only a weaker assumption will be necessary.

Lemma 4. Let f be twice continuously differentiable and assume that S is bounded. Let λ_k be the step-size generated by the algorithm. Then there exist scalars $\underline{\lambda}$ and $\bar{\lambda}$ such that:

$$0 < \underline{\lambda} < \lambda_k < \bar{\lambda}, \qquad \text{for all } k > 0$$

Proof. By twice continuous differentiability of f and the boundedness of S, for each $x \in S$ there exists a real number $C > 0$ such that:

$$\|\nabla f(x) - \nabla f(y)\| \leq C \|x - y\|, \quad \text{for all } y \in S \tag{2.6}$$

Now, let $x^k \in S$, $x_u^k \in E^n$, and $x_p^k \in R_k$. In addition, let $z = x^k + \lambda(x_p^k - x^k)$, $z \in S$, $\lambda > 0$. Then, using a Taylor Series Expansion formula from Polak (1971, p. 293), we have:

$$f(z) - f(x^k) = \nabla f(x^k)^t(z - x^k) + \int_o^1 \{\nabla f(x^k + t(z - x^k)) - \nabla f(x^k)\}^t (z - x^k) dt$$

Using (2.6) and the expression for z, we have:

$$f(z) - f(x^k) \leq \nabla f(x^k)^t [x^k + \lambda(x_p^k - x^k] + C \int_o^1 t\|z - x^k\|^2 dt$$

Using (2.3) from lemma 2, we obtain:

$$f(z) - f(x^k) \leq \frac{\lambda}{2\alpha_k} \|x_p^k - x^k\|^2 + \frac{C\lambda^2}{2} \|x_p^k - x^k\|^2$$

$$= \frac{-\lambda}{2\alpha_k} + \frac{C\lambda^2}{2} \quad \|x_p^k - x^k\|^2 \tag{2.7}$$

Now, from step 3 of the algorithms we can have one of the following cases:

i) $\nu_k = 0$ which implies that $\lambda_k = w_k = \text{Min}\{1, a_k\}$. Since $a_k \geq a > 0$ we have:

$$0 < a = \underline{\lambda} \leq \lambda_k \leq \bar{\lambda} = 1.$$

ii) $\nu_k > 0$ and the following holds:

$$f[x^k + (\beta_k)^{\nu_k - 1} w_k d^k] - f(x^k) > -\sigma_k (\beta_k)^{\nu_k - 1} w_k \frac{\|d^k\|^2}{2\alpha_k} \qquad (2.8)$$

Using (2.7) and (2.8) we have:

$$-\sigma_k (\beta_k)^{\nu_k - 1} w_k \frac{\|d^k\|^2}{2\alpha_k} < f[x^k + (\beta_k)^{\nu_k - 1} w_k d^k] - f(x^k)$$

$$\leq -\frac{(\beta_k)^{\nu_k - 1}}{2\alpha_k} + \frac{C}{2} (\beta_k)^{2(\nu_k - 1)} w_k w_k \|d^k\|^2$$

$$= (\beta_k)^{\nu_k - 1} w_k \left[-\frac{1}{2\alpha_k} + \frac{C}{2} (\beta_k)^{\nu_k - 1} w_k \right] \|d^k\|^2$$

From which we get:

$$-\frac{\sigma_k}{2\alpha_k} < -\frac{1}{2\alpha_k} + \frac{C}{2} (\beta_k)^{\nu_k - 1} w_k$$

Multiplying both sides by (β_k), we get:

$$\frac{(\beta_k)\ \sigma_k}{2\alpha_k} > \frac{(\beta_k)}{2\alpha_k} - \frac{C}{2}\,(\beta_k)^{\nu_k}\, w_k$$

And finally:

$$(\beta_k)^{\nu_k} > \frac{\beta_k(1-\sigma_k)}{C\alpha_k w_k} > \frac{\beta(1-\sigma_1)}{1.C.\alpha_1} > \frac{\beta}{2C\alpha_1} > 0$$

since $0 < \alpha_o < \alpha_k < \alpha_1$, $a < w_k < 1$, and $0 < \sigma_k < \sigma_1 < 1/2$. Therefore, we have:

$$\bar{\lambda} = 1 > \lambda_k > \mathrm{Min}\{w_k, \frac{2\beta(1-\sigma_1)}{C\alpha_1}\} = \underline{\lambda} > 0$$

and the proof is complete.

Note. From (2.7) and lemma 2 we have for

$$z = x^k + \lambda(x_p^k - x^k),\ \lambda > 0$$

$$f(z) - f(x^k) < (\lambda - C\alpha_k\lambda^2)\ \nabla f(x^k)^t(x_p^k - x^k)$$

Since $(x_p^k - x^k)$ is a descent direction, to insure that $f(z) < f(x^k)$, λ must satisfy:

$$\lambda < \frac{1}{C\alpha_k} \quad \text{or} \quad \alpha_k < \frac{1}{C}$$

Because $0 < C < \infty$, an appropriate choice of α_k will produce a step-size $\lambda = 1$ thus requiring only one trial step-size evaluation. This shows that the values for α_k used in the algorithm will effect the computational effort required for λ_k.

The next lemma will only use continuous differentiability to show that near a point $\bar{x}$ which is not a stationary point, the Armijo integer computed by the algorithm is bounded above and therefore the step-size is uniformly bounded.

Lemma 5. Let f be continuously differentiable. Let $\bar{x} \in S$ and suppose that $\bar{x}$ is not a stationary point. Then there is a neighborhood N of $\bar{x}$ and a number ν so that for $x^k \in S \cap N$, the Armijo integer ν_k in step 3 of the algorithm satisfies $0 < \nu_k < \nu$.

Proof. Let $x^k \in S$, then from lemma 2 we have:

$$\nabla f(x^k)^t d^k < - \frac{\|d^k\|^2}{2\alpha_k}$$

By lemma 3 $\|d^k\|$ is bounded away from zero and α_k is uniformly bounded, hence, there exists a positive number δ such that:

$$\nabla f(x^k)^t d^k < - \frac{\|d^k\|^2}{2\alpha_k} < -\delta \qquad (2.9)$$

for x^k sufficiently close to $\bar{x}$.

Since ∇f is continuous, there exist γ and $s > 0$ such that:

$$|\nabla f(x^k + td^k)^t d^k - \nabla f(x^k)^t d^k| < \frac{1}{2}\delta, \quad \text{for } t \in [0, \gamma] \qquad (2.10)$$

for all $x^k \in S$ such that $\|x^k - \bar{x}\| < s$

Now, given $0 < \beta \leq \beta_k < 1$, let ν be the smallest nonnegative integer such that $(\beta_k)^\nu \leq \gamma$. Let $x^k \in S$ N, then there exists $\theta \in [0,1]$ such that:

$$f[x^k + (\beta_k)^\nu w_k d^k] - f(x^k) + \sigma_k(\beta_k)^\nu w_k \frac{\|d^k\|^2}{2\alpha_k}$$

$$= (\beta_k)^\nu w_k \nabla f[x^k + \theta(\beta_k)^\nu w_k d^k]^t d^k + \sigma_k(\beta_k)^\nu w_k \frac{\|d^k\|^2}{2\alpha_k}$$

$$\leq (\beta_k)^\nu w_k \nabla f[x^k + \theta(\beta_k)^\nu w_k d^k]^t d^k - \sigma_k(\beta_k)^\nu w_k \nabla f(x^k) d^k$$

$$= (\beta_k)^\nu w_k [\nabla f(x^k + \theta(\beta_k)^\nu w_k d^k)^t d^k - \nabla f(x^k)^t d^k] + (1 - \sigma_k)(\beta_k)^\nu w_k \nabla f(x^k)^t d^k$$

Using (2.9) and (2.10) we have:

$$f[x^k + (\beta_k)^\nu w_k d^k] - f(x^k) + \sigma_k(\beta_k)^\nu w_k \frac{\|d^k\|^2}{2\alpha_k}$$

$$\leq (\beta_k)^\nu w_k \left(\frac{\delta}{2}\right) - (1 - \sigma_k)(\beta_k)^\nu w_k(\delta)$$

$$= (\beta_k)^\nu w_k \ \delta\left(\sigma_k - \frac{1}{2}\right) \leq \delta\left(\sigma - \frac{1}{2}\right) \leq 0$$

Since $(\beta_k)^{\nu} w_k > 0$, $\delta > 0$ and $0 < \sigma_o < \sigma_k < \sigma_1 < 1/2$. This shows that $\nu_k < \nu$ and the proof is complete.

<u>Note</u>. It follows from the above lemma that since $\lambda_k = (\beta_k)^{\nu_k} w_k$ we have:

$$\nu_k = 0 \Rightarrow \lambda_k = w_k < 1 \Rightarrow \bar{\lambda} = 1$$

$$\nu_k > 0 \Rightarrow (\beta_k)^{\nu_k} w_k > (\beta_k)^{\nu} w_k > (\beta)^{\nu} \text{Min}\{1,a\} = \underline{\lambda} > 0$$

and hence $0 < \underline{\lambda} < \lambda_k < \bar{\lambda} = 1$.

Finally, from lemmas 4 and 5 we can see that λ_k is obtained by a finite number of function evaluations, which demonstrates the feasibility of the step-size procedure. Now that we have established the properties of uniqueness, descent and boundedness of the direction vector, and the finiteness of the step-size procedure, we are ready to consider accumulation points of the sequence of iterates generated by the algorithm. The subsequence of restarting points, $\{x^k, k \in M\}$, will be considered first.

2.3 <u>Convergence of the Subsequence of Restarting Points</u>

We recall that the restarting subsequence is made up of points x^k, $k \in M$. It is generated along projected gradient directions and its behavior determines the global convergence properties of the algorithm. We will first establish that this subsequence is well-defined in the sense that if the algorithm generates an infinite sequence, then it also generates infinitely many restarting points.

Lemma 6. Let M be the set of restarting iterations. If M is infinite then for any given $i \in M$ there exists an integer ℓ such that

$$1 \leq \ell \leq n \quad \text{and} \quad i + \ell \in M$$

Proof. Let $i \in M$, and $i+1, \ldots, i+q_i \notin M$. Then by step 4 of the algorithm $J_i = q_i$ and $q_i < n$. Since M is infinite step 2-ii) of the algorithm cannot occur. Hence step 2-i) will be performed at most n times leading to step 2-iii). That is, for some $q_i < n$ and $i+q_i \notin M$ we must have $i+q_i + 1 = i+\ell \in M$, which completes the proof.

Note that this lemma also shows that from any point generated by the algoirthm, a restarting point is reached in a finite number of steps.

We now show that when the algorithm produces an infinite subsequence of restarting points, then its accumulation points are stationary points. For this we will use the properties of the direction vector and the step-size previously established.

Theorem 1. Let f be twice continuously differentiable and bounded below on the bounded set S. Let $x^k \in S$, $x_u^k \in E^n$ and $x_p^k \in R_k$. In addition, let $x^{k+1} = x^k + \lambda_k(x_p^k - x^k)$ with $\lambda_k \in (0,1]$ be the sequence of points generated by the algorithm. Denote $\{x^k: k \in M\}$ by $\{x^{k1}, x^{k2}, x^{k3}, \ldots, x^{kn} \ldots\}$, and define $z^n = x^{kn}$. Then the following statements will hold.

i) The sequence $\{f(x^k)\}$ converges to some limit f*.

ii) $$\lim_{k\to\infty} \|x_p^k - x^k\| = 0 \tag{2.11}$$

iii) $\lim_{k\to\infty} \nabla f(x^k)^t(x_p^k - x^k) = 0$ (2.12)

iv) $\lim_{n\to\infty} \|z^{n+1} - z^n\| = 0$ (2.13)

v) The limit of any convergent subsequence of $\{z^n\}$ is a stationary point of the problem:

$$\{\text{Min } f(x) \mid Ax \leq b\}$$

<u>Proof</u>. i) By construction, and moving lemma 4, the algorithm is well defined in that λ_k is obtained by a finite number of function evaluations (ν_k is bounded above). Hence $f(x^k)$ is monotonically decreasing and since it is bounded below, it must converge. This implies that:

$$\lim_{k\to\infty} [f(x^{k+1}) - f(x^k)] = 0$$

ii) From step 3 of the algorithm, we have:

$$f(x^k) - f(x^{k+1}) \geq \sigma_k \lambda_k \frac{\|x_p^k - x^k\|^2}{2\alpha_k} \geq 0$$

$$\frac{2\alpha_k}{\sigma_k\lambda_k} [f(x^k) - f(x^{k+1})] \geq \|x_p^k - x^k\|^2 \geq 0$$

By construction $\sigma_k \geq \sigma_o > 0$ and $\alpha_k < \alpha_1 (0 < \alpha_1 < \infty)$, and from lemma 4

$\lambda_k > \underline{\lambda} > 0$. Therefore $\delta_k = 2\alpha_k/\sigma_k\lambda_k < 2\alpha_1/\sigma_o\underline{\lambda}$ for all k. Therefore, we have:

$$\lim_{k\to\infty} \frac{2\alpha_k}{\sigma_k\lambda_k} \quad [f(x^k) - f(x^{k+1})] > \overline{\lim_{k\to\infty}} \|x_p^k - x^k\|^2 > 0$$

And using part i) above: $\lim\limits_{k\to\infty} \|x_p^k - x^k\|^2 = 0$.

iii) To show this part, we use the following from lemma 4:

$$f(z) - f(x^k) < \lambda\nabla f(x^k)^t(x_p^k - x^k) + \frac{C\lambda^2}{2} \|x_p^k - x^k\|^2$$

$$\text{where } z = x^k + \lambda(x_p^k - x^k),\ C > 0$$

In particular, given λ_k, $x^{k+1} = x^k + \lambda_k(x_p^k - x^k)$, and:

$$f(x^{k+1}) - f(x^k) < \lambda_k\nabla f(x^k)^t\ (x_p^k - x^k) + \frac{C\lambda_k^2}{2} \|x_p^k - x^k\|^2$$

Now, using part ii) above:

$$0 = \lim_{k\to\infty} [f(x^{k+1}) - f(x^k)] < \underline{\lim_{k\to\infty}}\ \lambda_k\nabla f(x^k)^t(x_p^k - x^k)$$

$$+ \lim_{k\to\infty} \frac{C\lambda_k^2}{2} \|x_p^k - x^k\|^2$$

Since $\lambda_k \in (0,1]$ and $0 < C < \infty$, we have:

$$0 < \lim_{l\to\infty} \lambda \nabla f(x_p^k)^t(x_p^k - x^k) < \underline{\lim}_{k\to\infty} \nabla f(x^k)^t(x_p^k - x^k) \tag{2.14}$$

Since $\nabla f(x^k)^t\ (x_p^k - x^k) < 0$ for all k, we have:

$$0 < \overline{\lim_{k\to\infty}}\ \nabla f(x^k)^t(x_p^k - x^k) < 0 \tag{2.15}$$

which implies that:

$$\lim_{k\to\infty} \nabla f(x^k)^t(x_p^k - x^k) = 0$$

Before we show the next part, we note that since $\{f(x^k)\} \to f^*$ and $\{f(z^n)\}$ is a subsequence of $\{f(x^k)\}$, we have $\{f(z^n)\} \to f^*$.

iv) Part ii) above showed that $\lim_{i\to\infty} \|x_p^i - x_i\| = \lim_{i\to\infty} \|d_i\| = 0$.

Using this fact with the definition of $\{z^n\}$, we have:

$$\|z^{n+1} - z^n\| = \|x^{k_{n+1}} - x^{k_n}\| = \|\sum_{i=k_n}^{k_{(n+1)}-1} \lambda_i d_i\| < \sum_{i=k_n}^{k_n+n} \|d_i\|$$

since $\lambda_i < 1$ for all i from lemma 4.

Now, taking limits we have:

$$0 < \lim_{n\to\infty} \|z^{n+1} - z^n\| < \lim_{k_n\to\infty} \sum_{i=k_n}^{k_{(n+1)}-1} \|d_i\| = 0$$

and it follows that: $\lim\limits_{n\to\infty} \| z^{n+1} - z^{n} \| = 0.$

v) To prove this part we assume that the subsequence $\{z^n\}$ converges to z^* in the set $\{x \mid Ax \leq b\}$. We now show that z^* is a stationary point of the problem $\{\text{Min } f(x) \mid Ax \leq b\}$.

Let y be any point in $\{x \mid Ax \leq b\}$. Then:

$$\nabla f(z^n)^t (y - z^n) = \nabla f(z^n)^t (y - z_p^n + z_p^n - z^n)$$

where z_p^n is the projection of z_u^n onto $\{x \mid Ax \leq b\}$.

$$\nabla f(z^n)^t (y - z^n) = \nabla f(z^n)^t (y - z_p^n) + \nabla f(z^n)^t (z_p^n - z^n)$$

$$\geq \frac{1}{2\alpha_n} (z_p^n - z^n)^t (z_p^n - y) + \nabla f(z^n)^t (z_p^n - z^n)$$

(from lemma 2-i)

Using the Schwartz inequality and lemma 2, we have:

$$-(z^n - z_p^n)^t (z_p^n - y) \geq - \left[2\alpha_n \nabla f(z^n)^t (z^n - z_p^n) \| z_p^n - y \|\right]^{\frac{1}{2}}$$

from which we get:

$$\nabla f(z^n)^t (y - z^n) \geq \frac{1}{\alpha_n} \; -\left[2\alpha_n \nabla f(z^n)^t (z^n - z_p^n)\right]^{\frac{1}{2}} \| z_p^n - y \|$$

$$+ \nabla f(z^n)^t (z_p^n - z^n) \qquad (2.16)$$

Now, since the set $\{x \mid Ax \le b\}$ is assumed to be bounded, and z_p^n, $y \varepsilon \{x \mid Ax \le b\}$, there exists a positive number B such that $\| z_p^n - y \| \le B$ for all n and for all $y \varepsilon \{x \mid Ax \le b\}$. In addition, using the fact that $0 < \alpha_o \le \alpha_n$, n, (2.16) can be written as:

$$\nabla f(z^n)^t (y - z^n) \ge \frac{-B\sqrt{2}}{\sqrt{\alpha_o}} [\nabla f(z^n)^t (z^n - z_p^n)]^{\frac{1}{2}} + \nabla f(z^n)^t (z_p^n - z^n)$$

Now, as $n \to \infty$ and by continuity of $\nabla f(x)$, we have:

$$\lim_{n\to\infty} \nabla f(z^n)^t (y - z^n) \ge \frac{-B\sqrt{2}}{\sqrt{\alpha_o}} [\lim_{n\to\infty} [\nabla f(z^n)^t (z^n - z_p^n)]^{\frac{1}{2}}$$

$$+ \lim_{n\to\infty} \nabla f(z^n)^t (z_p^n - z^n)$$

Finally, from part iii) of this theorem, the right hand side of the last expression goes to zero and we get:

$$\nabla f(z^*)(y - z^*) \ge 0 \qquad (2.17)$$

Since y is any point in $\{x \mid Ax \le b\}$, and since $\{x \mid Ax \le b\}$ is closed and convex, expression (2.17) shows that z* is a stationary point of f in $\{x \mid Ax \le b\}$

Remark. If the twice continuous differentiability assumption is dispensed with, then under the weaker conditions of lemmas 3 and 5 we can show the following: Any accumulation point of the subsequence $\{z^n\}$

is a stationary point.

Clearly, if we assume $\{z^n\}$ $K \to z^*$, $K \subset \{0,1,2...\}$ and by contradiction we suppose that z^* is not a stationary point, then we have:

$$f(z^n) - f(z^{n+1}) \geq \sum_{i=k_n}^{k_{(n+1)}-1} [\sigma_i \lambda_i \frac{\|d_i\|^2}{2\alpha_i}]$$

$$\geq \sum_{i=k_n}^{k_{(n+1)}-1} \sigma_o \underline{\lambda} \delta \geq \sigma\lambda\delta > 0$$

Since for k large enough $\sigma_i \geq \sigma_o$, $\lambda_i \geq \underline{\lambda}$, $\|d_i\|^2/2\alpha_i \geq \delta$ and $(k^{n+1} - k_n)$ is finite.

Now, since f decreases at each iteration if follows that $f(z^n) \to -\infty$. This is impossible since $f(z^n)$ is bounded below and hence converges to f^*.

As a result of the last theorem, it was shown that the differences in the tail of the subsequence of restarting points converge to zero. This property together with the assumption that the subsequence has a finite number of accumulation points will enable us to show that it converges to a single stationary point. This is the object of the next theorem.

Theorem 2. Let f be continuously differentiable and bounded below on the bounded set S. In addition, assume that the subsequence $\{z^n\}$ defined in theorem 1 has a finite number of accumulation points. Then $\{z^n\}$ converges to a single stationary point.

Proof. First, we note that z^n is feasible for all $n > 0$, and that $f(z^n) < f(x_o)$, for all $n > 0$. Therefore $\{z^n\} \in S$. Also, since $\lim_{n\to\infty} \|z^{n+1} - z^n\| = 0$, this sequence is "strongly downward" as defined by Orteg and Rheinboldt (1970, theorem 14.1.5).

Now, let the set of accumulation points of $\{z^n\}$ be

$$\Phi = \{\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^\ell\}, \ \ell > 0$$

then:

$$\delta = \text{Min}\{\|\bar{x}^i - \bar{x}^j\| \mid i \neq j;\ i,j = 1,\ldots,\ell\} > 0$$

Note that since ℓ is finite δ is well defined. We can choose $n_o > 0$ such that z^n is within $\delta/4$ of one of the ℓ accumulation points and at the same time $\|z^{n+1} - z^n\| < \delta/4$, for all $n > n_o$. To see that this choice is possible suppose that $\|z^n - \bar{x}^i\| < \delta/4$ for a given i and that $\|z^{n+1} - z^n\| > \delta/4$ for all $n > n_o$. This implies that $\|z^{n+1} - z^n\| \not\to 0$, a contradiction.

Now, consider z_j where $j > n_o$. Then there exists an $1 < i < \ell$ so that $\|z_j - \bar{x}^i\| < \delta/4$. Without loss of generality suppose $i = 1$. Then we have:

$$\|\bar{x}^{i'} - \bar{x}^1\| < \|\bar{x}^{i'} - z_{j+1}\| + \|z_{j+1} - \bar{x}^1\|$$

$$< \|\bar{x}^{i'} - z_{j+1}\| + \|\bar{x}^1 - z_j\| + \|z_j - z_{j+1}\|, \ i' > 2$$

which implies that:

$$\|\bar{x}^{i'} - z_{j+1}\| > \|\bar{x}^{i'} - \bar{x}^{1}\| - (\|\bar{x}^{1} - z_{j}\| + \|z_{j} - z_{j+1}\|)$$

$$> \delta - 2\ \delta/4 = \delta/2,\ i' > 2$$

In other words, z_{j+1} is not within $\delta/4$ of the accumulation points $\bar{x}^2, \bar{x}^3, \ldots, \bar{x}^{\ell}$. Therefore it must be within $\delta/4$ of $\bar{x}^1$ for all $n > j$. This means that $\bar{x}^2, \ldots, \bar{x}^{\ell}$ cannot be limit points of $\{z^n\}$ which implies that $\ell=1$ and that $\{z^n\}$ converges to a single point in Φ. Since all points in Φ are stationary points, the proof is complete.

Finally, to complete the study of the subsequence of restarting points we show that global converges will take place under convexity of the objective function. This is the object of the next theorem.

Theorem 3. Let f be continously differentiable and bounded below on the bounded set S. In addition, assume that there exist scalars m_1 and m_2 with $0 < m_1 < m_2$ such that:

$$m_1 \|y\|^2 < y^t G(x) y < m_2 \|y\|^2,\ x \in S \text{ and } y \in E^n.$$

Let $\{z^n\}$ be the sequence defined in theorem 1. Then:

i) $\lim_{n\to\infty} f(z^n) = f^* = \text{Min}\{f(x) \mid x \in S\}$

ii) $\{z^n\} \to x^*$, the unique minimizer of f over S.

Proof. i) By assumption, f(x) is convex which implies that it has a unique minimizer x* on the set S. Hence:

$$0 < f(z^n) - f(x^*) \qquad (2.18)$$

Also, from the Taylor Series Expansion:

$$f(x^*) = f(z^n) + \nabla f^t(z^n)(x^* - z^n) + \frac{1}{2}(x^* - z^n)^t G(\zeta^*)(x^* - z^n)$$

where $\zeta^* \in L(x^*, z^n)$, we have:

$$f(x^*) - f(z^n) > -\nabla f(z^n)^t(z^n - x^*) + \frac{1}{2} m \|x^* - z^n\|^2$$

$$> -\nabla f(z^n)^t(z^n - x^*)$$

$$\Rightarrow f(z^n) - f(x^*) < \nabla f(z^n)^t(z^n - x^*) \qquad (2.19)$$

Now, using lemma 2, part i), we have:

$$\nabla f(z^n)^t(z^n - x^*) < \nabla f(z^n)^t(z^n - z_p^n) - \frac{1}{2\alpha_n}(z_p^n - z^n)^t(z_p^n - x^*)$$

$$= \nabla f(z^n)^t(z^n - z_p^n) + \frac{1}{2\alpha_n}(z^n - z_p^n)^t(z_p^n - x^*)$$

$$< \nabla f(z^n)^t(z^n - z_p^n) + \frac{1}{2\alpha_n}\|z^n - z_p^n\| \|z_p^n - x^*\|$$

Also, using the fact that:

$$\|z^n - z_p^n\|^2 < 2\alpha_n \nabla f(z^n)^t(z^n - z_p^n)$$

we get:

$$\nabla f(z^n)^t(z^n - x^*) \le \nabla f(z^n)^t(z^n - z_p^n)$$

$$+ \frac{1}{2\alpha_n} \left[2\alpha_n \nabla f(z^n)^t(z^n - z_p^n)\right]^{\frac{1}{2}} \| z_p^n - x^* \| \qquad (2.20)$$

Now, since z_p^n, x^* are in the set $\{x \mid Ax \le b\}$ and this set is assumed to be bounded, there exist a positive number D such that:

$$\| z_p^n - x^* \| \le D \quad \text{for all } n \ge 0$$

In addition, since $0 < \alpha_o \le \alpha_n$ for all $n \ge 0$, we obtain from (2.18), (2.19) and (2.20) the following expression:

$$0 \le f(z^n) - f(x^*) \le \nabla f(z^n)^t(z^n - x^*)$$

$$\le \nabla f(z^n)^t(z^n - z_p^n) + \frac{D}{\sqrt{2\alpha_o}} \left[\nabla f(z^n)^t(z^n - z_p^n)\right]^{\frac{1}{2}}$$

Now, as n goes to infinity, we have:

$$0 \le \lim_{n\to\infty} \{f(z^n) - f(x^*)\} \le \lim_{n\to\infty} \{\nabla f(z^n)^t(z^n - z_p^n)$$

$$+ \lim_{n\to\infty} \frac{D}{\sqrt{2\alpha_o}} \quad \nabla f(z^n)^t(z^n - z_p^n) \qquad (2.21)$$

The right hand side of (2.21) goes to zero from part iii) of theorem 1 and we finally get:

$$0 \leq \lim_{n \to \infty} \{f(z^n) - f(x^*)\} \leq 0$$

$$\Rightarrow \lim \{f(z^n)\} = f(x^*) = \text{Min}\{f(x) \mid Ax \leq b\}$$

which comples the proof of part i)

ii) To prove this part we note that the set Φ defined in theorem 2 contains a single point. That is, $\Phi = \{x^*\}$ since f(x) is convex. Therefore, the entire sequence $\{z^n\}$ converges to x* since S is bounded.

This now completes the convergence study of the subsequence of restarting points. In the next part we will consider the sequence of all points (both restarting and conjugate) and show that it essentially has the same convergence properties.

2.4 Convergence of the Sequence Generated by the Algorithm

This part is a direct consequence of the convergence results established so far. We first show that cluster points of the sequence are stationary points and that the sequence is "strongly downward." Under additional assumptions the whole sequence will converge.

Theorem 4. Let f be continuously differentiable and bounded below on the bounded set S. The algorithm will produce a sequence $\{x^k\}$ with the following properties:

i) $x^k \in S$ and $\{f(x^k)\}$ is strictly decreasing

ii) $\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0$

iii) Every accumulation point of $\{x^k\}$ is a stationary point.

<u>Proof</u>. i) This part holds by definition.

ii) From the expression:

$$x^{k+1} = x^k + \lambda_k(x_p^k - x^k)$$

we have:

$$0 < \|x^{k+1} - x^k\| < \lambda\|x_p^k - x^k\|$$

By lemma 5, $\lambda_k < 1, \forall k$, which implies that:

$$0 < \lim_{k\to\infty} \|x^{k+1} - x^k\| < \lim_{k\to\infty} \|x_p^k - x^k\|$$

And by part ii) of theorem 1, it follows that:

$$\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0 \qquad (2.22)$$

iii) To show this part we let $\bar{x}$ be an accumulation point of $\{x^k\}$. That is $\{x^k\}\kappa \to \bar{x}$, $\kappa \subset \{0,1,2,...\}$. The three following cases may arise:

<u>Case 1</u>: The elements of κ correspond to restarting iterations. That is κ M. By part 2.3 above $\bar{x}$ is a stationary point.

Case 2: Some elements of κ are in M and some others are not in M (that is κ is a set of mixed iterations). Then we can extract a further subsequence $\{x^k\}\kappa' \subset K$ with elements all in M, and $\{x^k\}\kappa' \to \bar{x}$. Since $\kappa' \subset M$ if follows again that $\bar{x}$ is a stationary point.

Case 3: All the elements of κ are not in M. Here we assume by contradiction that $\bar{x}$ is not a stationary point. Then by lemma 3 there is a member $\delta > 0$ so that $\|d_k\|^2/2\alpha_k \geq \delta$ for $k \in \kappa$ large enough. Also, by lemma 5 there exist a member $\underline{\lambda} > 0$ so that $\lambda_k \geq \underline{\lambda}$ for $k \in \kappa$ large enough. Then, by step 3 of the algorithm we have:

$$f(x^k) - f(x^{k+1}) \geq \sigma_k \lambda_k \frac{\|d_k\|^2}{2\alpha_k} > \frac{\sigma_o \underline{\lambda}}{2\alpha_1} \delta > 0 \qquad (2.23)$$

since $0 < \sigma_o \leq \sigma_k$ and $0 < \alpha_o \leq \alpha_k \leq \alpha_1$ for all $k \geq 0$.

By part i) of this theorem, (2.23) implies that $f(x^k) \to -\infty$. This contradicts the fact the $\{f(x^k)\}$ converges to f^* and hence $\bar{x}$ is a stationary point, completing the proof of the theorem.

Corollary 4.1. Under the assumptions of theorem 4, if $\{x^k\}$ has a finite number of accumulation points, it converges to a single one.

Proof. Using the facts that $\{f(x^k)\}$ is decreasing and that $\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0$, the result follows from the argument used in theorem 2.

Corollary 4.2. Under the assumptions of theorem 4 and assuming that f is strictly convex, the sequence $\{x^k\}$ generated by the algorithm converges to the unique minimizer of f over S.

Proof. Here also the proof follows from the same argument used in theorem 2.

CHAPTER V

LOCAL CONVERGENCE ANALYSIS

1. Introduction

In this chapter we show that the sequence of points generated by the algorithm proposed in Chapter III will exhibit a superlinear rate of convergence in the vicinity of a solution point. For that we will need some stronger assumptions which we now state and will refer to whenever required as assumption A.

Assumption A. Given the function f and the sequence $\{x^k\}$ generated by the algorithm, the following properties are assumed:

i) The sequence $\{x^k\}$ generated by the algorithm converges to x^*.

ii) f is twice continuously differentiable in a neighborhood of x^*, $N_{\varepsilon_o}(x^*)$, $\varepsilon_o > 0$.

iii) There exist members m_1 and m_2 with $0 < m_1 < m_2$ such that:

$$m_1 \|y\|^2 < y^t G(x^*) y < m_2 \|y\|^2$$

for all $y \in E^n$ with $a_i^t y = 0$; $i=1,\ldots,r$

iv) The strict complementary slackness condition holds at x^*. That is, assuming that $\{a_1,\ldots,a_r\}$ are linearly independent, we have:

$$a_i^t x^* = b_i; \; i=1,\ldots,r$$

$$a_i^t x^* < b_i; \; i=r+1,\ldots,m$$

$$u_i^* > 0 \quad ; \; i=1,\ldots,r$$

In addition, three intermediate results will be need to prove the rate of convergence. Therefore, the remainder of the chapter will be organized as follows:

i) The first part will be concerned with showing that after a while the set of binding constraints will not change (theorem 5). The importance of this result is that it shows that once the algorithm gets close enough to x*, it eventually becomes equivalent to a conjugate direction method on the subspace $\{y \mid a_i^t y = 0; \; i=1,\ldots,r\}$. Consequently, the rate of convergence of the method is determined by the Hessian of f restricted to the subspace $\{y \mid a_i^t y = 0; \; i=1,\ldots,r\}$ rather than the whole space E^n. That the set of binding constraints will stabilize will result from lemmas 7, 8, and 9. These show that the sequence of Kuhn-Tucker multipliers generated by solving the projection problem converges to the Kuhn-Tucker multipliers u* at the solution point x*.

ii) This part will show that the initial approximation to the step-size of the Mukai-type will have the necessary properties required by the conjugate directions algorithm. First, the step-size is uniformly bounded (lemma 10). Second, the initial approximation will be close enough to the exact step-size so that it will always be taken without further trials (theorem 6).

iii) The third part will show that eventually the directions constructed by the algorithm will satisfy the following conjugacy property:

Given d^k and d^{k+1}, then:

$$\frac{(d^k)^t G(x^*) d^{k+1}}{\|d^k\| \, \|d^{k+1}\|} \to 0 \quad \text{as } k \to \infty$$

To see how that will be achieved by the algorithm we note the following:

Assuming that f is twice continuously differentiable, by Taylor's Theorem we have:

$$d^t[\nabla f(x^{k+1}) - \nabla f(x^k)] = d^t \left[\int_0^1 G[x^k + t(x^{k+1} - x^k)] \, (x^{k+1} - x^k) dt\right]$$

$$= d^t G(\zeta^k)(x^{k+1} - x^k), \; \zeta^k \in L_k(x^k, x^{k+1})$$

$$= \lambda_k d^t G(\zeta^k) d^k$$

$$= \lambda_k d^t G(x^*) d^k + \lambda_k d^t [G(\zeta^k) - G(x^*)] \, d^k$$

Now if we denote the error term $[G(\zeta^k) - G(x^*)]$ by E_k, we have:

$$[\nabla f(x^{k+1}) - \nabla f(x^k)]^t d^{k+1} = \lambda_k (d^k)^t G(x^*) d^{k+1} + \lambda_k (d^k)^t E_k d^{k+1}$$

from which:

$$\frac{[\nabla f(x^{k+1}) - \nabla f(x^k)]^t d^{k+1}}{\lambda_k \|d^k\| \|d^{k+1}\|} = \frac{\lambda_k (d^k)^t G(x^*) d^{k+1}}{\lambda_k \|d^k\| \|d^{k+1}\|} + \frac{\lambda_k (d^k)^t E_k d^{k+1}}{\lambda_k \|d^k\| \|d^{k+1}\|} \quad (1.1)$$

Now for $k \to \infty$, $x^k \to x^*$ and $E_k \to 0$ which gives:

$$\frac{(d^k)^t G(x^*) d^{k+1}}{\|d^k\| \|d^{k+1}\|} \to 0 \quad \text{as} \quad k \to \infty$$

since in the algorithm the left-hand side of (1.1) is always equal to zero. This result will be established in lemma 11.

iv) Finally, in the last part we establish the superlinear rate of convergence (theorem 7). The basic idea behind the result is as follows:

From the Kuhn-Tucker conditions of the projection problem we have:

$$(x_p^k - x^k) + \alpha_k [\nabla f(x^k) + A^t \frac{u_k}{\alpha_k}] + Y_k^t v_k = 0$$

$$\text{or,} \qquad d^k + \alpha_k [\nabla f(x^k) + A^t \frac{u_k}{\alpha_k}] = -Y_k^t v_k \qquad (1.2)$$

As $d^k \to 0$, $\nabla f(x^k) \to \nabla f(x^*)$ and $A^t u_k / \alpha_k \to \bar{A}^t u^*$ where $\bar{A}$ is the matrix of binding constraints, the left-hand side of (1.2) will go to zero. How fast will that happen depends on how fast $Y_k^t v_k$ goes to zero. We show in lemma 13 that $Y_k^t v_k$ is governed by second order information. That is,

$Y_k^t v_k = O[G(x^k) - G(x^*)]$ because the approximate conjugacy property defined above holds. Hence the second order rate of convergence will be achieved. Lemma 12 will establish some intermediate results which are needed for lemma 13 and theorem 7.

2. Stabilization of the Set of Binding Constraints

To show that the binding constraints will eventually remain the same we will first show that $u_k \to u^*$, the vector of Kuhn-Tucker multipliers at x^*. For this we first need the following intermediate result establishing a property of multipliers associated with linear independent vectors.

Lemma 7. Suppose that $\{a_1,\ldots,a_r\}$ are linearly independent. Then there exists $\gamma_1 > 0$ such that if $z \in E^n$ is given by $z = \sum_{i=1}^{r} \Pi_i a_i$, then we have:

$$\underset{1\leq i\leq r}{\text{maximum}} \{|\Pi_i|\} \leq \gamma_1 \|z\|$$

Proof. Let $A = [a_1,\ldots,a_r]$ an (nxr) matrix whose column are the column vectors a_i, i=1,...,r. Given $z = \sum_{i=1}^{r} \Pi_i a_i$ and letting $\Pi = (\Pi_1,\ldots,\Pi_r)^t$ we have:

$$z = A\Pi \quad \text{or} \quad A^t z = (A^t A)\Pi$$

from which $$\Pi = (A^t A)^{-1} A^t z$$

Clearly, $$\pi_i = [(A^tA)^{-1} A^t]_i \; z$$

And

$$|\pi_i| = |[(A^tA)^{-1}A^t]_i \; z| < \|(A^tA)^{-1}A^t\|_i \; \|z\|$$

$$< \underset{1<i<r}{\text{Maximum}} \{\|(A^tA)^{-1}A^t\|_i \; \|z\|\} = \gamma_1 \|z\|$$

where $$\gamma_1 = \underset{1<i<r}{\text{Maximum}} \{\|(A^tA)^{-1}A^t\|_i\}$$

which completes the proof.

Let $k \in M$ and $k+1, k+2, \ldots, k+q_k \notin M$. That is q_k conjugate directions are contructed after the kth restart. Let $0 < \ell < q_k$ and note that ℓ is dependent on k. The next result will show that for $i=r+1,\ldots,m$ $[u_{k+\ell}]_i \to 0$. Clearly, if $\ell=0$ it corresponds to a restarting iteration and if $\ell>1$ it corresponds to a conjugate direction iteration.

Lemma 8. Let f be continuously differentiable and bounded below on the bounded set S. In addition, let assumption A defined in section one be satisfied. Let $[u_{k+\ell}]_i$ be the ith component of the Kuhn-Tucker vector associated with the inequality constraints of the $(k+\ell)$th projection problem. Then for k large enough we have:

$$[u_{k+\ell}]_i \to 0; \; i=r+1,\ldots,m, \; 0 < \ell < q_k$$

Proof. From the Kuhn-Tucker conditions of the projection problem (PP) with $0 < \ell < q_k$, we have:

$$(x_p^{k+\ell} - x^{k+\ell}) + \alpha_{k+\ell}[g_{k+\ell} + \sum_{i=1}^{m} \frac{[u_{k+\ell}]_i}{\alpha_{k+\ell}} a_i] + \sum_{j=0}^{\ell} [v_{k+\ell}]_j y_{k+j} = 0 \quad (2.1)$$

$$[u_{k+\ell}]_i \ (a_i^t x_p^{k+\ell} - b_i) = 0; \ i=1,\dots,m \quad (2.2)$$

$$y_{k+j}^t x_p^{k+\ell} = (e)_{k+j} \quad ; \ j=0,\dots,\ell \quad (2.3)$$

where if $\ell=0$, from the algorithm we have $y_k = 0$, and the above conditions reduce to the Kuhn-Tucker conditions of the kth projection problem, with $k \in M$. First, we note that (2.3) is equivalent to:

$$y_{k+j}^t (x_p^{k+\ell} - x^{k+\ell}) = 0 \quad ; \ j=0,\dots,\ell \quad (2.4)$$

Now, multiplying (2.1) through by $(x_p^{k+\ell} - x^{k+\ell})$ we get:

$$\|x_p^{k+\ell} - x^{k+\ell}\|^2 + \alpha_{k+\ell}[g_{k+\ell}^t (x_p^{k+\ell} - x^{k+\ell}) + \sum_{i=1}^{m} \frac{[u_{k+\ell}]_i}{\alpha_{k+\ell}} a_i^t (x_p^{k+\ell} - x^{k+\ell})]$$

$$+ \sum_{j=0}^{\ell} [v_{k+\ell}]_j y_{k+j}^t (x_p^{k+\ell} - x^{k+\ell}) = 0$$

Using (2.4) and letting:

$$\delta_{k+\ell} = \|x_p^{k+\ell} - x^{k+\ell}\|^2 + \alpha_{k+\ell} \ g_{k+\ell}^t (x_p^{k+\ell} - x^{k+\ell})$$

we have:

$$\sum_{i=1}^{m} [u_{k+\ell}]_i a_i^t (x_p^{k+\ell} - x^{k+\ell}) + \delta_{k+\ell} = 0 \tag{2.5}$$

Now, from (2.2) we have:

$$[u_{k+\ell}]_i \; a_i^t x_p^{k+\ell} = [u_{k+\ell}]_i \; b_i$$

and (2.5) can be written as:

$$\sum_{i=1}^{m} ([u_{k+\ell}]_i a_i^t x_p^{k+\ell} - [u_{k+\ell}]_i a_i^t x^{k+\ell}) + \delta_{k+\ell} =$$

$$\sum_{i=1}^{m} ([u_{k+\ell}]_i b_i - [u_{k+\ell}]_i a_i^t x^{k+\ell}) + \delta_{k+\ell} =$$

$$\sum_{i=1}^{m} [u_{k+\ell}]_i (b_i - a_i^t x^{k+\ell}) + \delta_{k+\ell} = 0$$

Now, as $k \to \infty$, by theorem 1 in Chapter III, $\delta_{k+\ell} \to 0$ and

$$\sum_{i=1}^{m} [u_{k+\ell}]_i (b_i - a_i^t x^{k+\ell}) \to 0 \tag{2.6}$$

But since $b_i - a_i^t x^{k+\ell} \geq 0$, $[u_{k+\ell}]_i \geq 0$ for all k and $0 \leq \ell \leq q_k$, (2.6) is equivalent to:

$$[u_{k+\ell}]_i \; (b_i - a_i^t x^{k+\ell}) \to 0; \quad i=1,\dots,m$$

Also, as $k \to \infty$, $x^{k+\ell} \to x^*$ and for $i=r+1,\ldots,m$, using assumption A:

$$b_i - a_i^t x^{k+\ell} \to b_i - a_i^t x^* > 0$$

which implies that there exists a k_0 such that for all $k > k_0$, $b_i - a_i^t x^{k+1} > 0$ for $i=r+1,\ldots,m$ and therefore:

$$[u_{k+\ell}]_i \to 0; \quad i=r+1,\ldots,m; \quad 0 < \ell < q_k.$$

which complete the proof of this lemma.

We next show that for k large enough the Kuhn-Tucker multipliers associated with the binding constraints will be strictly positive.

Lemma 9. Let f be continuously differentiable and bounded below on the bounded set S. In addition, let assumption A defined in section one be satisfied. Let $[u_{k+\ell}]_i$ be as defined in lemma 8. Then for k large enough we have:

$$\frac{[u_{k+\ell}]_i}{\alpha_{k+\ell}} \to [u^*]_i \ ; \quad i=1,\ldots,r; \quad 0 < \ell < q_k$$

Proof. Here we will use an induction argument. Let $\ell=0$ then from (2.1) in lemma 8 we have:

$$(x_p^k - x^k) + \alpha_k[g_k + \sum_{i=1}^{m} \frac{[u_k]_i}{\alpha_k} a_i] = 0 \qquad (2.7)$$

Also, at x^* we have:

$$\nabla f(x^*) + \sum_{i=1}^{r} [u^*]_i \, a_i = 0 \tag{2.8}$$

where $[u^*]_i$, $i=1,\ldots,r$ are unique since a_i, $i=1,\ldots,r$ are linearly independent.

From (2.8) and (2.7) we have:

$$[g_k - \nabla f(x^*)] + \frac{1}{\alpha_k}(x_p^k - x^k) + \sum_{i=r+1}^{m} \frac{[u_k]_i}{\alpha_k} a_i = \sum_{i=1}^{r} \left([u^*]_i - \frac{[u_k]_i}{\alpha_k} \right) a_i$$

Since $\{a_1,\ldots,a_r\}$ are linearly independent, from lemma 7 there exists $\gamma_1 > 0$ such that:

$$0 < \underset{1<i<l}{\text{maximum}} \left\{ \left| [u^*]_i - \frac{[u_k]_i}{\alpha_k} \right| \right\} < \gamma_1 \left(\|g_k - \nabla f(x^*)\| + \frac{1}{\alpha_k} \|x_p^k - x^k\| + \sum_{i=r+1}^{m} \left| \frac{[u_k]_i}{\alpha_k} \right| \|a_i\| \right)$$

Now, by continuity of ∇f, theorem 1 in Chapter III, lemma 8, and since α_k is uniformly bounded, as $k \to \infty$, we get:

$$0 < \underset{1<i<l}{\text{maximum}} \left\{ \left| [u^*]_i - \frac{[u_k]_i}{\alpha_k} \right| \right\} \to 0$$

which proves the desired result fro $\ell=0$.

Now assume that the result holds for $0 < \ell < q_k$. That is

$$\frac{[u_{k+\ell}]_i}{\alpha_{k+\ell}} \to [u^*]_i, \quad \text{for } i=1,\ldots,r \tag{2.9}$$

We need to establish that it also holds true for $(\ell+1)$. From (2.1) in lemma 8 and (2.8), after rearranging terms, we have:

$$\sum_{i=1}^{\ell} [v_{k+\ell}]y_{k+j} = -\Big[d^{k+\ell} + \alpha_{k+\ell}(g_{k+\ell} - \nabla f(x^*)) + \alpha_{k+\ell}\sum_{i=1}^{r}\Big(\frac{[u_{k+\ell}]_i}{\alpha_{k+\ell}} - [u^*]_i\Big)\, a_i + \sum_{i=r+1}^{m} [u_{k+\ell}]a_i\Big]$$

Denoting the right hand side of the above expression by $\delta_{k+\ell}$ and noting that by theorem 1 of Chapter III, lemma 8, the continuity of g, (2.9) and the uniform boundedness of $\alpha_{k+\ell}$, $\delta_{k+\ell} \to 0$ as $k \to \infty$, and we have:

$$\delta_k = \sum_{j=1}^{\ell} [v_{k+\ell}]_j y_{k+j} \Rightarrow \underset{0<j<\ell}{\text{maximum}} \{|v_{k+\ell}|_j\} < \gamma_2 \|\delta_k\| \tag{2.10}$$

which follows from the linear independence of $\{y_{k+1},\ldots,y_{k+\ell}\}$ and lemma 7.

Finally, from (2.10), as $k \to \infty$, we get:

$$0 < \underset{1<j<\ell}{\text{maximum}} \{|v_{k+\ell}|_j\} \to 0 \text{ and hence } \|v_{k+\ell}\| \to 0 \tag{2.11}$$

Using (2.11), the (k+ℓ+1)th projection problem and lemma 7 one more time, the same argument implies that:

$$\frac{[u_{k+\ell+1}]_i}{\alpha_{k+\ell+1}} \to [u^*]_i > 0; \quad i=1,\dots,r.$$

and the proof is complete.

We note that as a consequence of this lemma, there exists a k_o such that for all $k > k_0$, we have:

$$[u_{k+\ell}]_i > 0; \quad i=1,\dots,r; \quad 0 < \ell < q_k$$

Using lemmas 8 and 9 we next show that eventually the set of binding constraints will stablize.

<u>Theorem 5</u>. Let f be continuously differentiable and bounded below on the bounded set S. In addition, let assumption A defined in section one be satisfied. Let $[u_{k+\ell}]_i$ be as defined in lemma 8. Then there exists an integer k_0 such that for $k > k_0$:

$$a_i^t x^k = b_i; \quad i=1,\dots,r$$

$$a_i^t x^k < b_i; \quad i=r+1,\dots,m$$

<u>Proof</u>. By lemma 9, there exists an integer k_o such that for $k > k_0$:

$$\frac{[u_{k+\ell}]_i}{\alpha_{k+\ell}} > 0, \text{ for } 0 \leq \ell \leq q_k \text{ and } i=1,\ldots,r.$$

Also, from:

$$[u_{k+\ell}]_i \; (a_i^t x_p^{k+\ell} - b_i) = 0; \quad 0 \leq \ell \leq q_k; \quad i=1,\ldots,r$$

we have that for $k \geq k_0$:

$$a_i^t x_p^{k+\ell} = b_i; \quad 0 \leq \ell \leq q_k; \; i=1,\ldots,r$$

From the iterates generated by the algorithm:

$$x^{k+\ell+1} = x^{k+\ell} + \lambda_{k+\ell} \; (x_p^{k+\ell} - x^{k+\ell}); \quad 0 \leq \ell \leq q_k$$

Two cases can occur:

i) $\lambda_{k+\ell} = 1$. This implies that $x^{k+\ell+1} = x_p^{k+\ell}$ and

$a_i^t x^{k+\ell+1} = b_i$, $i=1,\ldots,r$ and the result follows.

ii) $\lambda_{k+\ell} < 1$. In this case we note that by construction, $x_p^k = x^k$ for restarting iterations, and we have:

$$a_i^t x^k = a_i^t x_p^k = b_i$$

$$a_i^t x^{k+q_k+1} = a_i^t x_p^{k+q_k+1} = b_i \qquad i=1,\ldots,r \qquad (2.12)$$

Now, by definition:

$$x^{k+q_k+1} - x^k = \sum_{j=1}^{q_k+1} (x^{k+j} - x^{k+j-1})$$

Using (2.12), we have:

$$0 = a_i^t(x^{k+q_k+1} - x^k) = \sum_{j=1}^{q_k+1} a_i^t(x^{k+j} - x^{k+j-1}); \quad i=1,\ldots,r \qquad (2.13)$$

In addition, we observe that:

$$a_i^t(x^{k+j} - x^{k+j-1}) = \lambda_{k+j-1}\, a_i^t(x_p^{k+j-1} - x^{k+j-1}) =$$

$$\lambda_{k+j-1}\,(b_i - a_i^t x^{k+j-1}) \geq 0, \qquad i=1,\ldots,r \quad j=1,\ldots,q_k+1 \qquad (2.14)$$

(2.13) and (2.14) imply that:

$$a_i^t(x^{k+j} - x^{k+j-1}) = 0, \qquad i=1,\ldots,r \quad j=1,\ldots,q_k+1 \qquad (2.15)$$

And finally, (2.15) is written as:

$$\lambda_{k+j-1}\, a_i^t(x_p^{k+j-1} - x^{k+j-1}) = 0, \quad \begin{matrix} i=1,\ldots,r \\ j=1,\ldots,q_k+1 \end{matrix}$$

Since λ_{k+j-1} is bounded away from zero for all k, it follows that:

$$a_i^t(x_p^{k+j-1} - x^{k+j-1}) = 0$$

$$\Rightarrow a_i^t x_p^{k+j-1} = a_i^t x^{k+j-1} = b_i, \quad \begin{matrix} i=1,\ldots,r \\ j=1,\ldots,q_k+1 \end{matrix}$$

And hence, there is a k_0 such that for all $k \geq k_0$: $a_i^t x^k = b_i$, $i=1,\ldots,r$. Finally we note that for $i=r+1,\ldots,m$ for k large enough, $a_i^t x^k - b_i \to a_i^t x^* - b_i < 0$ which completes the proof.

3. Properties of the Initial Step-Size Approximation

As we have mentioned previously, when k gets large enough the conjugate directions algorithm requires that the step-sizes taken be close to the exact ones. This requirement will be used in lemma 13. In this part we describe an extension of Mukai's (1978) approximation to the intitial step-size for unconstrained problems. The idea is to use as an initial trial step-size an estimate based on the local quadratic approximation. We first explain its derivation.

Given x^k and d^k, then a good estimate of the step-size along d^k

can be obtained by minimizing the quadratic approximation of f at x^k along d^k:

$$f(x^k + ad^k) \simeq f(x^k) + a\nabla f(x^k)^t d^k + \frac{1}{2} a^2 (d^k)^t G(x^k) d^k$$

whenever the Hessian matrix $G(x^k)$ is positive definite. That is,

$$\text{Minimize } \{f(x^k + ad^k) \mid a \in [0,\infty)\}$$

yields the unique solution:

$$a_k = \frac{-\nabla f(x^k)^t d^k}{(d^k)^t G(x^k) d^k}$$

The above formula will only be valid if the denominator is positive, and since $(d^k)^t G(x^k) d^k$ is not readily available, it can be approximated as follows:

Given x^k, d^k, for a small $\varepsilon > 0$ we have:

$$\tau_k = f(x^k + 2\varepsilon d^k) - 2f(x^k + \varepsilon d^k) + f(x^k) \simeq \varepsilon^2 (d^k)^t G(x^k) d^k$$

or, equivalently:

$$\tau_k = 2[f(x^k + \varepsilon d^k) - f(x^k) - \varepsilon\nabla f(x^k)^t d^k] \simeq \varepsilon^2 (d^k)^t G(x^k) d^k$$

Now, given $\delta > 0$, an approximate way to check for positive curvature is to test if:

$$(d^k)^t G(x^k) d^k \geq \delta \| d^k \|^2 > 0$$

or, since $(d^k)^t G(x^k) d^k = \tau_k/\varepsilon^2$, then if $\tau_k \geq \varepsilon^2 \delta \, \| d^k \|^2$, the approximation a_k of the step-size will be acceptable. Otherwise, if $\tau_k < \delta^2 \delta \| d^k \|^2$, then either $(\tau_k/\varepsilon^2) < 0$ or δ is not small enough. In this case δ is reduced by half for the purpose of the next iteration, and the step-size will be taken equal to one.

We note that even though a_k gives an estimate of the step-size when the local quadratic approximation is valid, there is no guarantee that in general $f(x^k + a_k d^k) < f(x^k)$ is achieved. The use of the Armijo scheme, however, will insure that it will happen as demonstrated by lemma 5 in Chapter III. As k gets sufficiently large, it will be seen that conjugate directions require a step-size that closely approximates the exact one in the sense that:

$$g_{k+1}^t \, d^k \to 0$$

The above Mukai-type approximation will insure that the estimated step-size has this property, and in addition, as $x^k \to x^*$, the Armijo number will be equal to zero, so that the initial trial will be the only trial needed.

We modify step 3 of the algorithm in the following way:

- Given δ_0 and ε positive numbers

- Compute $\tau_k = f(x^k + 2\varepsilon d^k) - 2f(x^k + \varepsilon d^k) + f(x^k)$

- Let

$$a_k = \begin{cases} -\dfrac{\varepsilon^2 \nabla f(x^k)^t d^k}{\tau_k}, & \text{if } \tau_k > \varepsilon^2 \delta_k \|d^k\|^2 \qquad (3.1) \\ 1 & \text{, otherwise} \qquad (3.2) \end{cases}$$

- Let w_k = minimize $\{a_k, 1\}$ and set $\lambda_k = (\beta_k)^{\nu_k} w_k$, where

$$\nu_k = \text{minimum} \left\{ \begin{matrix} \nu > 0 \\ \text{integer} \end{matrix} \;\middle|\; f(x^k + (\beta_k)^{\nu} w_k d^k] - f(x^k) \le -\sigma_k(\beta_k)^{\nu} w_k \frac{\|d^k\|^2}{2\alpha_k} \right\}$$

- $\delta_{k+1} = \delta_k$ if a_k is given by (3.1) and $\delta_k = \delta_k/2$ is a_k is given by (3.2) above.

We next show that if the Mukai-type approximation is used to compute the initial step-size a_k, then the step-size λ_k will remain bounded away from zero.

Lemma 10. Let $\bar{x} \in S$ and suppose that $\bar{x}$ is not a stationary point. If a_k is defined as in (3.1) and (3.2) then the step-size λ_k will remain bounded above and below away from zero for $x^k \in S$ in a neighborhood of $\bar{x}$.

Proof. From lemma 5 in Chapter III there exists $\underline{\lambda} > 0$ and $\bar{\lambda}$ such that $\underline{\lambda} \le \lambda_k \le \bar{\lambda}$ if it is shown here that w_k is bounded above and below away from zero for $x^k \in S$ close to $\bar{x}$. By definition w_k = minimize$\{1, a_k\}$ so that $w_k \le 1$. Now we need to show that a_k is bounded. By definition $a_k \le 1$. Also, two possible cases may arise:

- If $\tau_k < \varepsilon^2\delta_k \|d^k\|^2$, then $a_k = 1 > 0$

- If $\tau_k \geq \varepsilon\delta_k \|d^k\|^2$, then for x^k close $\bar{x}$, by lemma 3 in Chapter III there exists a positive number s such that $\|d^k\|^2 \geq s$. In addition since the test passes we have $\tau_k > 0$ and by continuity of f, as x^k is close to $\bar{x}$, τ_k is bounded above. So that we have:

$$a_k = \frac{-\varepsilon^2\nabla f(x^k)^t d^k}{\tau_k} \geq \frac{\varepsilon^2\|d^k\|^2}{2\alpha_k\tau_k} \geq \frac{\varepsilon^2 s}{2\alpha_1\tau_k} > 0$$

This shows that for $x^k \in S$ sufficiently close to $\bar{x}$, the scalars $\underline{\lambda}$ and $\bar{\lambda}$ exist such that: $0 < \underline{\lambda} \leq \lambda_k \leq \bar{\lambda}$.

In the next result we show that for k large enough, under appropriate assumptions, the Mukai-Type initial approximation will result in a good estimate of the step-size in a single trial. That means that the Armijo member ν_k will be equal to zero.

Theorem 6. Let f be continuously differentiable and bounded below on the bounded set S. In addition, let assumption A defined in section one be satisfied. Let λ_k be the step-size determined by the algorithm with a_k determined by the Mukai-Type approximation. Then there exists an integer k_0 such that for all $k \geq k_0$, $\nu_k = 0$.

Proof. We first need to show that the test for positive curvature will eventually always be satisfied and that $\lambda_k = w_k$.

a) For k large enough, we show that $\tau_k \geq \varepsilon^2\delta_k \|d^k\|^2$.

We use the following Hessian approximations defined by Mukai (1978, p. 993):

$$G_k^{\varepsilon} = 2 \int_0^1 (1 - t)\, G(x^k + t\varepsilon d^k)dt$$

$$G_k^{2\varepsilon} = 2 \int_0^1 (1 - t)\, G(x^k + \varepsilon d^k + t\varepsilon d^k)dt, \; t \in [0,1)$$

$$G_k^{\varepsilon'} = \int_0^1 G(x^k + t\varepsilon d^k)dt$$

And from Mukai (1978, eq. A23) we have:

$$\tau_k = f(x^k + 2\varepsilon d^k) - 2f(x^k + \varepsilon d^k) + f(x^k)$$

$$= \varepsilon^2 (d^k)^t G_k^{\varepsilon'} d^k + \frac{1}{2} \varepsilon^2 (d^k)^t [G_k^{2\varepsilon} - G_k^{\varepsilon}] d^k$$

For k large enough, and since f is assumed to be twice continuously differnentiable and $x^k \to x^*$, we have $G_k^{2\varepsilon}$, G_k^{ε}, $G_k^{\varepsilon'} \to G(x^*)$. In addition from theorem 5, there is a k_0 such that for all $k > k_0$, $a_i^t(x_p^k - x^k) = a_i^t d^k = 0$ for $i=1,\ldots,r$. Thus, there exists $m_1 > 0$ such that:

$$\tau_k > \varepsilon^2 \frac{m_1}{2} \|d^k\|^2 + \frac{1}{2} \varepsilon (d^k)^t [G_k^{2\varepsilon} - G_k^{\varepsilon}] d^k$$

Now, for any k > 0 if the positive curvature test fails we have:

$$\varepsilon^2 \delta_k \|d^k\|^2 > \tau_k > \varepsilon^2 \frac{m_1}{2} \|d^k\|^2 + \frac{1}{2} \varepsilon^2 (d^k)^t [G_k^{2\varepsilon} - G_k^{\varepsilon}] d^k$$

from which it follows that:

$$\delta_k > \frac{m_1}{2} + \frac{1}{2} (\bar{d}^k)^t [G_k^{2\varepsilon} - G_k^{\varepsilon}]\bar{d}^k, \text{ where } \bar{d}^k = \frac{d^k}{\|d^k\|}$$

As k gets large enough, we get:

$$\delta_k > m_1 > 0 \tag{3.3}$$

Now, suppose that the test for positive curvature fails infinitely often, then δ_k will be halved infinitely often, that is, $\delta_k \to 0$, contradicting (3.3) above. Therefore, there is a k_0 such that for all k $> k_0$, we will have:

$$\tau_k > \varepsilon\delta_k \|d^k\|^2.$$

b) For k large enough, we show that:

$$f(x^{k+1}) - f(x^k) < -\sigma_k w_k \frac{\|d^k\|^2}{2\alpha_k}$$

By definition, we have:

$$a_k = \frac{-\varepsilon^2 \nabla f(x^k)^t d^k}{\tau_k} = \frac{-\nabla f(x^k)^t d^k}{(d^k)^t G_k^{\varepsilon'} d^k + \frac{1}{2} (d^k)^t (G_k^{2\varepsilon} - G_k^{\varepsilon}) d^k} \tag{3.4}$$

Two possible cases may arise:

i) $a_k < 1$, then $w_k = \text{minimum}\{1, a_k\} = a_k$.

Using the definition:

$$f(x^k + \alpha d^k) - f(x^k) = \alpha \nabla f(x^k)^t d^k + \frac{1}{2} \alpha^2 (d^k)^t G_k^{\alpha} d^k$$

We have:

$$f(x^k + w_k d^k) - f(x^k) + \sigma_k w_k \frac{\|d^k\|^2}{2\alpha_k} = w_k \nabla f(x^k)^t d^k$$

$$+ \frac{1}{2} w_k^2 (d^k)^t G_k^{w_k} d^k + \sigma_k w_k \frac{\|d^k\|^2}{2\alpha_k}$$

$$< w_k \nabla f(x^k)^t d^k + \frac{1}{2} w_k^2 (d^k)^t G_k^{w_k} d^k - \sigma_k w_k \nabla f(x^k)^t d^k \quad \text{(By lemma 2, Chapter IV)}$$

$$= (1 - \sigma_k)\, w_k \nabla f(x^k)^t d^k + \frac{1}{2} w_k^2 (d^k)^t G_k^{w_k} d^k$$

$$= -(1 - \sigma_k)\, w_k^2\, [(d^k)^t G_k^{\varepsilon'} d^k + \frac{1}{2} (d^k)^t (G_k^{2\varepsilon} - G_k^{\varepsilon}) d^k] + \frac{1}{2} w_k^2 (d^k)^t G_k^{w_k} d^k$$

$$= w_k^2\, [-(\frac{1}{2} + \frac{1}{2} - \sigma_k)\, [(d^k)^t G_k^{\varepsilon'} d^k + \frac{1}{2} (d^k)^t (G_k^{2\varepsilon} - G_k^{\varepsilon}) d^k] + \frac{1}{2} (d^k)^t G_k^{w_k} d^k]$$

$$= w_k^2 [- \frac{1}{2} (d^k)^t G_k^{\varepsilon'} d^k - (\frac{1}{2} - \sigma_k)(d^k)^t G_k^{\varepsilon'} d^k$$

$$- \frac{1}{2}(1 - \sigma_k)(d^k)^t(G_k^{2\varepsilon} - G_k^{\varepsilon})d^k + \frac{1}{2}(d^k)^t G_k^{w_k} d^k]$$

$$= w_k^2 \, [-(\frac{1}{2} - \sigma_k)(d^k)^t G_k^{\varepsilon'} d^k - \frac{1}{2}(1 - \sigma_k)(d^k)^t(G_k^{2\varepsilon} - G_k^{\varepsilon})d^k$$

$$+ \frac{1}{2}(d^k)^t(G_k^{w_k} - G_k^{\varepsilon'})d^k]$$

$$< w_k^2 \, \|d^k\|^2 \, [\frac{m_1}{2}(\sigma_k - \frac{1}{2}) + \frac{1}{2}(1 - \sigma_k)\, \|G_k^{2\varepsilon} - G_k^{\varepsilon}\| + \frac{1}{2}\|G_k^{w_k} - G_k^{\varepsilon'}\|]$$

$$< w_k^2 \, \|d^k\|^2 \, \frac{m_1}{2}(\sigma_k - \frac{1}{2}) < 0, \text{ since } \sigma_k < \frac{1}{2} \text{ for all } k > 0.$$

Thus, we showed that, in this case:

$$f(x^k + w_k d^k) - f(x^k) + \sigma_k w_k \frac{\|d^k\|^2}{2\alpha_k} < 0$$

ii) Suppose $a_k > 1$, then $w_k = 1$. And from (3.4) we have:

$$\nabla f(x^k)^t d^k < - [(d^k)^t G_k^{\varepsilon'} d^k + \frac{1}{2}(d^k)^t(G_k^{2\varepsilon} - G_k^{\varepsilon})d^k]$$

Using the exact same derivation as in part i) above we get:

$$f(x^k + w_k d^k) - f(x^k) + \sigma_k w_k \frac{\|d^k\|^2}{2\alpha_k} < \frac{m_1}{2}(\sigma_k - \frac{1}{2})\, \|d^k\|^2 < 0$$

Therefore, it follows that there is a k_0 such that for $k > k_0$, $\lambda_k = (\beta_k)^{\nu_k} w_k = w_k$, which implies that $\nu_k = 0$.

Finally, we note that for k large enough, since the positive curvature test was shown to always be satisfied, we have:

$$f(x^k + a_k d^k) - f(x^k) = a_k \nabla f(x^k)^t d^k + \frac{1}{2} a_k^2 (d^k)^t G(x^k) d^k$$

$$= a_k \nabla f(x^k)^t d^k - \frac{1}{2} a_k \nabla f(x^k)^t d^k = \frac{1}{2} a_k \nabla f(x^k)^t d^k < 0$$

This, together with theorem 6 shows that for k large enough $\lambda_k = a_k$, which means that the step-size estimate based on the local quadratic approximation will be used as the actual step-size in the line search. As $x^k \to x^*$, this step-size closely approximates the exact one.

<u>Note</u>. If $\tau_k = 2[f(x^k + \varepsilon d^k) - f(x^k) - \varepsilon \nabla f(x^k)^t d^k]$ is used to compute a_k, then:

$$a_k = \frac{-\varepsilon^2 \nabla f(x^k)^t d^k}{\tau_k} = \frac{-\nabla f(x^k)^t d^k}{(d^k)^t G_k^{\varepsilon'} d^k}$$

and the same results shown by theorem 6 will hold. That is $\nu_k = 0$ and $\lambda_k = a_k$ for $k > k_0$. The above form for τ_k has some computational advantages when $\nabla f(x^k)$ is readily available and thus only $f(x^k + \varepsilon d^k)$ need to be computed, resulting in only one additional function evaluation.

4. Verification of the Approximate Conjugacy of Directions

This part will have a single result. Based on theorem 5, which established that the set of binding constraints will not change for k large enough, and the second order assumptions around x^*, it will be shown that indeed the algorithm constructs conjugate directions. As $x^k \to x^*$ the conjugacy property will hold with respect to the Hessian $G(x^*)$. Assuming that $k \in M$ and $k+q_k \notin M$ we have the following lemma.

Lemma 11. Let $\{x^k\}$ be the sequence generated by the algorithm and let assumption A defined in section one be satisfied. Then, for k sufficiently large, and $1 \leq \ell < q_k$, the following will hold:

$$\text{i)}\quad y^t_{k+\ell+1}\,\bar{d}^{k+i} = \begin{cases} (\bar{d}^{k+\ell})^t(E_{k+\ell} - E_{k+i})\,\bar{d}^{k+i}; & i=0,1,\ldots,\ell-1 \\ \geq \eta > 0 & ; \quad i=\ell \end{cases}$$

$$\text{ii)}\quad y^t_{k+i}\,\bar{d}^{k+\ell} = (\bar{d}^{k+i-1})^t(E_{k+i-1} - E_{k+\ell})\,\bar{d}^{k+\ell} \;;\quad i=\ell+2,\ldots,q_k-1$$

where $\bar{d}_j = d_j/\|d_j\|$

Proof. i) Let us first consider the case where $i=\ell$.

$$y^t_{k+\ell+1}\,\bar{d}^{k+\ell} = \frac{(g_{k+\ell+1} - g_{k+\ell})^t\,\bar{d}^{k+\ell}}{\|\lambda_{k+\ell}\,d^{k+\ell}\|}$$

By Taylor's theorem, we have for $\zeta^{k+\ell} \in L(x^{k+\ell+1}, x^{k+\ell})$

$$\frac{(g_{k+\ell+1} - g_{k+\ell})^t \bar{d}^{k+\ell}}{\|\lambda_{k+\ell}\ d^{k+\ell}\|} = \frac{\lambda_{k+\ell}(d^{k+\ell})^t G(\zeta^{k+\ell})\bar{d}^{k+\ell}}{\lambda_{k+\ell}\ \|d^{k+\ell}\|} = (\bar{d}^{k+\ell})^t G(\zeta^{k+\ell})\bar{d}^{k+\ell}$$

Now, for k large enough, $\zeta^{k+\ell} \in N_\varepsilon(x^*)$ and $\bar{A}\ \bar{d}^{k+\ell} = 0$ by theorem 5, so we obtain:

$$y^t_{k+\ell+1}\bar{d}^{k+\ell} = (\bar{d}^{k+\ell})^t G(\zeta^{k+\ell})\bar{d}^{k+\ell} > \frac{m_2}{2}\ \|\bar{d}^{k+\ell}\|^2 = \frac{m_2}{2} = \eta_0 > 0.$$

We now consider the case where $i=0,1,\ldots,\ell-1,\ 0 < \ell < q_k$.

$$y^t_{k+\ell+1}\bar{d}^{k+i} = \frac{(g_{k+\ell+1} - g_{k+\ell})^t \bar{d}^{k+i}}{\|\lambda_{k+\ell}\ d^{k+\ell}\|} = \frac{\lambda_{k+\ell}(d^{k+\ell})^t G(\zeta^{k+\ell})\bar{d}^{k+i}}{\lambda_{k+\ell}\ \|d^{k+\ell}\|}$$

$$= (\bar{d}^{k+\ell})^t G(\zeta^{k+\ell})\bar{d}^{k+i} = (\bar{d}^{k+\ell})^t\ [E_{k+\ell} + G(x^*)]\ \bar{d}^{k+i} \qquad (4.1)$$

where: $E_{k+\ell} = G(\zeta^{k+\ell}) - G(x^*)$ and $\zeta^{k+\ell} \in L(x^{k+\ell+1},\ x^{k+\ell})$

Now, using the fact that by construction, for $i=0,\ldots,\ell-1$:

$$y^t_{k+i+1}\ \bar{d}^{k+\ell} = \frac{(g_{k+i+1} - g_{k+i})^t\ \bar{d}^{k+\ell}}{\|\lambda_{k+i}\ d^{k+i}\|} = 0$$

we have:

$$(\bar{d}^{k+\ell})^t G(x^*)\bar{d}^{k+i} = (\bar{d}^{k+\ell})^t G(x^*)\bar{d}^{k+i} - \frac{(\bar{d}^{k+\ell})^t (g_{k+i+1} - g_{k+i})}{\|\lambda_{k+i}\; d^{k+i}\|}$$

Or, equivalently:

$$(\bar{d}^{k+\ell})^t G(x^*)\bar{d}^{k+i} = (\bar{d}^{k+\ell})^t G(x^*)\bar{d}^{k+i} - (\bar{d}^{k+\ell})^t \frac{[\lambda_{k+i}\; G(\zeta^{k+i})\; d^{k+i}}{\lambda_{k+i}\; \|d^{k+i}\|}$$

$$= (\bar{d}^{k+\ell})^t G(x^*)\bar{d}^{k+i} - (\bar{d}^{k+\ell})^t G(\zeta^{k+i})\bar{d}^{k+i}$$

where $\zeta^{k+i} \in L(x^{k+i+1}, x^{k+i})$

Therefore:

$$(\bar{d}^{k+\ell})^t G(x^*)\bar{d}^{k+i} = -(\bar{d}^{k+\ell})^t [G(\zeta^{k+i}) - G(x^*)]\bar{d}^{k+i}$$

$$= -(\bar{d}^{k+\ell})^t [E_{k+i}]\; \bar{d}^{k+i} \qquad (4.2)$$

Now, using (4.2) in (4.1) above, we get:

$$y^t_{k+\ell+1}\; \bar{d}^{k+i} = (\bar{d}^{k+\ell})^t [E_{k+\ell}]\; \bar{d}^{k+i} - (\bar{d}^{k+\ell})^t\; [E_{k+1}]\bar{d}^{k+i}$$

$$= (\bar{d}^{k+\ell})^t\; [E_{k+\ell} - E_{k+i}]\; \bar{d}^{k+i}$$

which holds for i=0,1,..., -1, and completes the proof of part i).

ii) Let $i=\ell+2,\ \ell+\ ,\ldots,\ q_k-1$.

$$y_{k+i}^{t}\ \bar{d}^{k+\ell} = \frac{(g_{k+i} - g_{k+i-1})\ \bar{d}^{k+\ell}}{\lambda_{k+i-1}\ \|d^{k+i-1}\|} = \frac{\lambda_{k+i-1}(d^{k+i-1})^{t}G(\zeta^{k+i-1})\ \bar{d}^{k+\ell}}{\lambda_{k+i-1}\ \|d^{k+i-1}\|}$$

$$= (\bar{d}^{k+i-1})^{t}G(\zeta^{k+i-1})\bar{d}^{k+\ell} = (\bar{d}^{k+i-1})^{t}[G(x^*) + E_{k+i-1}]\bar{d}^{k+\ell}$$

$$= (\bar{d}^{k+i-1})^{t}G(x^*)\bar{d}^{k+\ell} + (\bar{d}^{k+i-1})^{t}(E_{k+i-1})\bar{d}^{k+\ell} \qquad .(4.3)$$

Now, using the fact that by construction, for $i=\ell+2,\ldots,q_k-1$:

$$y_{k+\ell+1}^{t}\ \bar{d}^{k+i-1} = \frac{(g_{k+\ell+1} - g_{k+\ell})^{t}\ \bar{d}^{k+i-1}}{\|\lambda_{k+}\ \ d^{k+\ell}\|} = 0$$

by a similar argument used to derive (4.2), we can write:

$$(\bar{d}^{k+i-1})^{t}G(x^*)\bar{d}^{k+\ell} = (\bar{d}^{k+i-1})^{t}G(x^*)\bar{d}^{k+\ell} - \frac{(\bar{d}^{k+i-1})^{t}(g_{k+\ell+1} - g_{k+1})}{\|\lambda_{k+}\ \ d^{k+\ell}\|}$$

$$= (\bar{d}^{k+\ell})^{t}[G(x^*) - G(\zeta^{k+\ell})]\ d^{k+i-1} = (-\bar{d}^{k+\ell})^{t}(E_{k+\ell})\ \bar{d}^{k+i-1} \qquad (4.4)$$

Using (4.4) in conjunction with (4.3), we get:

$$y_{k+i}^t \bar{d}^{k+\ell} = (-\bar{d}^{k+\ell})^t [E_{K+\ell}] \bar{d}^{k+i-1} + (\bar{d}^{k+\ell})^t [E_{k+i-1}] \bar{d}^{k+i-1}$$

$$= (\bar{d}^{k+i-1})^t [E_{k+i-1} - E_{k+\ell}]\, d^{k+\ell} \tag{4.5}$$

which holds for $i=\ell+2,\ldots,q_k$ and completes the proof of part ii) and the lemma.

5. Superlinear Convergence

The rate of convergence of the algorithm will depend crucially on the rate at which $\|v_k\|$ converges to zero. It will be seen that the following elements are important in that regard:

i) The set of binding constraints will not change. This means that terms of the form $[u_k]_i\ a_i$; $i=r+1,\ldots,m$ will not be included in any bound on $\|v_k\|$.

ii) The step-sizes along the conjugate directions are close approximations to the exact ones. This means that $g_{k+1}^t d^k$ should go to zero.

iii) If the directions are conjugate then a bound on $\|v_k\|$ could be derived only in terms of $g_k^t\ d^{k-1}$, $g_k^t\ d^{k-2}$,..., these quantities tending to zero as fast as the error in the Hessian approximation, that is very rapidly. In lemma 13 we will derive such a bound and use it in theorem 7 to establish superlinear convergence. The next lemma, however, establishes some intermediate result which will be needed to obtain the desired bound.

Lemma 12. Let $\{x^k\}$ be the sequence generated by the algorithm and let assumption A defined in section one be satisfied. Then for k large enough the following expressions will hold:

i) $\|x^{k+q_k} - x^*\| = O(\|x^k - x^*\|)$

ii) $\|x^{k+q_k} - x^*\| = O(\|\nabla f(x^{k+q_k}) + r_{k+q_k}\|)$

where $r_{k+q_k} \in \{r \mid r = \sum_{i=1}^{\bar{r}} t_i a_i\}$

iii) $\|\nabla f(x^{k+q_k}) - \nabla f(x^*)\| = O(\|x^{k+q_k} - x^*\|)$

Proof. i) By theorem 5, there exists k_0 that for $k > k_0$, $\bar{A}d^k = A(x_p^k - x^k) = 0$. Also, since x^* is a stationary point, we have:

$$\nabla f(x^*)^t(x^k - x^*) \geq 0, \quad \text{for all } x^k \in S$$

$$\nabla f(x^*) + \bar{A}^t u^* = 0$$

From which we get:

$$0 \leq \nabla f(x^*)^t(x^k - x^*) = -\mu^{*t}\bar{A}\,(x^k - x^*) \leq 0$$

since $\quad \bar{A}(x^k - x^*) \leq \bar{b} - \bar{b} = 0$

Therefore $\quad \nabla f(x^*)^t(x^k - x^*) = 0 \qquad (5.1)$

In addition, for $\zeta^0 \in L_k(x^k, x^*)$

$$f(x^k) - f(x^*) = \nabla f(x^*)^t(x^k - x^*) + \frac{1}{2}(x^k - x^*)^t G(\zeta^0)(x^k - x^*) \qquad (5.2)$$

Since for k large enough, $\zeta_0 \in N_{\varepsilon_0}(x^*)$, there exist numbers m_1, m_2 with $0 < m_1 \leq m_2$ such that:

$$\frac{m_1}{2}\|x^k - x^*\|^2 \leq (x^k - x^*)^t G(\zeta^0)(x^k - x^*) \leq \frac{m_2}{2}\|x^k - x^*\|^2$$

This, together with (5.1) and (5.2) gives:

$$\frac{m_1}{2}\|x^k - x^*\|^2 \leq f(x^k) - f(x^*) \leq \frac{m_2}{2}\|x^k - x^*\|^2 \qquad (5.3)$$

From theorem 1 in Chapter III, $f(x^{k+q_k}) \leq f(x^k)$ for all $q_k \geq 0$, so that for k large enough, we have:

$$f(x^{k+q_k}) - f(x^*) \leq f(x^k) - f(x^*)$$

and it follows that:

$$\frac{m_1}{2}\|x^{k+q_k} - x^*\|^2 \leq f(x^{k+q_k}) - f(x^*) \leq f(x^k) - f(x^*) \leq \frac{m_2}{2}\|x^k - x^*\|^2$$

which gives the desired result:

$$\|x^{k+q_k} - x^*\| \leq \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}}\|x^k - x^*\| \quad \text{or} \quad \|x^{k+q_k} - x^*\| = O(\|x^k - x^*\|)$$

ii) To show this part we let $r_{k+q_k} \varepsilon \{r \mid r = \sum_{i=1}^{r} t_i a_i\}$ where the t_i's are multipliers. Then from theorem 5 for k large enough $a_i^t d^k = 0$, $i=1,\ldots,r$ and hence $r_{k+q_k}^t d^{k+q} = 0$. Also from theorem 5 and the relation $\nabla f(x^*) + \sum_{i=1}^{r} u_i^* a_i = 0$ we have:

$$[\nabla f(x^*) + \sum_{i=1}^{r} u_i^* a_i]^t (x^{k+q_k} - x^*) = 0 \Rightarrow \nabla f(x^*)^t (x^{k+q_k} - x^*) = 0$$

Now,

$$(\nabla f_{k+q_k} + r_{k+q_k})^t (x^{k+q_k} - x^*) = \nabla f_{k+q_k}^t (x^{k+q_k} - x^*) + r_{k+q_k}^t (x^{k+q_k} - x^*)$$

$$= \nabla f_{k+q_k}^t (x^{k+q_k} - x^*) \qquad (5.4)$$

and by Taylor's theorem:

$$[(\nabla f_{k+q_k} - \nabla f(x^*)]^t (x^{k+q_k} - x^*) = (x^{k+q_k} - x^*)^t G(\zeta^{k+q_k})(x^{k+q_k} - x^*)$$

$$= \nabla f_{k+q_k}^t (x^{k+q_k} - x^*) \qquad (5.5)$$

where $\zeta^{k+q_k} \varepsilon L[x^{k+q_k}, x^{k+q_k} + t(x^{k+q_k} - x^*)]$

From (5.4) and (5.5) we have:

$$(\nabla f_{k+q_k} + r_{k+q_k})^t (x^{k+q_k} - x^*) = (x^{k+q_k} - x^*)^t G(\zeta^{k+q_r})(x^{k+q_k} - x^*)$$

For k large enough $\zeta^{k+q_k} \in N_{\varepsilon_0}(x^*)$ and we get:

$$(\nabla f_{k+q_k} + r_{k+q_k})^t (x^{k+q_k} - x^*) \geq \frac{m_1}{2} \| x^{k+q_k} - x^* \|^2$$

from which:

$$\frac{m_1}{2} \| x^{k+q_k} - x^* \|^2 \leq (\nabla f_{k+q_k} + r_{k+q_k})^t (x^{k+q_k} - x^*)$$

$$\leq \| \nabla f_{k+q_k} - \nabla f(x^*) \| \; \| x^{k+q} - x^* \|$$

Hence:

$$\| x^{k+q_k} - x^* \| \leq \left(\frac{2}{m_1}\right) \| \nabla f_{k+q_k} + r_{k+q_k} \| \tag{5.6}$$

which completes the proof of this part.

iii) Here we use Taylor's theorem to get:

$$\|\nabla f(x^{k+q_k}) - \nabla f(x^*)\|^2 = (x^{k+q_k} - x^*)^t G(\zeta^{k+q_k})[\nabla f(x^{k+q_k}) - \nabla f(x^*)]$$

$$\leq \|x^{k+q_k} - x^*\| \; \|G(\zeta^{k+q_k})\| \; \|\nabla f(x^{k+q_k}) - \nabla f(x^*)\|$$

As k get large enough, $\zeta^{k+q_k} \in N_{\varepsilon_0}(x^*)$ and hence there exists $\Delta > 0$ such that: $\|G(\zeta^{k+q_k})\| \leq \Delta$. Therefore we obtain:

$$\|\nabla f(x^{k+q_k}) - \nabla f(x^*)\| \leq \Delta \, \|x^{k+q_k} - x^*\| \tag{5.7}$$

and the proof of the lemmas is complete.

The next lemma is the one in which we show that the bound on $\|v_k\|$ goes to zero rapidly. The details of the proof of this lemma are rather lengthy but the substance of it is based on two key points which were previously discussed:

- The step-size approximates the exact one near x*.

- The stabilization of the binding constraints and the conjugacy of the direction insure that the quantities $g_k^t \, d^{k-2}$, $g_k^t \, d^{k-2}$,..., go to rapidly. These quantities will make up the bound on $\|v_k\|$.

<u>Lemma 13.</u> Let $\{x^k\}$ be the sequence generated by the algorithm and let assumption A defined in section one be satisfied. Then as k gets sufficiently large, the following bounds will hold:

$$\text{i)} \; |g_{k+\ell+1}^t \bar{d}^{k+\ell}| \leq 2 \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}} \|x^k - x^*\| \, \|E_{k+\ell} - E_{k+\ell}^{\varepsilon}\|$$

ii) $|g^t_{k+q_k} \bar{d}^{k+\ell}| < 2 \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}} \|x^k - x^*\| \left[\sum_{i=\ell+2}^{q_k} (\|E_{k+i-1} - E_{k+\ell}\|) + \|E_{k+\ell} - E^{\varepsilon}_{k+\ell}\|\right]$

iii) $\|v_{k+q_k}\| = O\left(\|x^k - x^*\| \left[\sum_{i=\ell+2}^{q_k} (\|E_{k+i-1} - E_{k+\ell}) + \|E_{k+\ell} - E^{\varepsilon}_{k+\ell}\|\right]\right)$

iv) $\|\sum_{i=1}^{q_k} [v_{k+q_k}]_i y_{k+i}\| = O(\|v_{k+q_k}\|)$

<u>Proof</u>. i) From theorem 6 for k large enough,

$$\lambda_{k+\ell} = \frac{-g^t_{k+\ell} d^{k+\ell}}{(d^{k+\ell})^t G^{\varepsilon}_{k+\ell} d^{k+\ell}}, \text{ where } G^{\varepsilon}_k = \int_0^1 G(x^{k+\ell} + t\varepsilon d^{k+\ell})dt$$

$$\Rightarrow + g^t_{k+\ell} d^{k+\ell} = -\lambda_{k+\ell} (d^{k+\ell})^t G^{\varepsilon}_{k+\ell} \; d^{k+\ell} \tag{5.8}$$

Also, from Taylor's theorem:

$$g^t_{k+\ell+1} \bar{d}^{k+\ell} = g^t_{k+\ell} \bar{d}^{k+\ell} + \lambda_{k+\ell} (\bar{d}^{k+\ell})^t G(\zeta^{k+\ell}) d^{k+\ell}$$

Using (5.8) we have:

$$g^t_{k+\ell+1}\bar{d}^{k+\ell} = -\lambda_{k+\ell}(\bar{d}^{k+\ell})^t G^{\varepsilon}_{k+\ell} d^{k+\ell} + \lambda_{k+\ell}(\bar{d}^{k+\ell})^t G(\zeta^{k+\ell}) d^{k+\ell}$$

$$= \lambda_{k+\ell}(d^{k+\ell})^t [G(\zeta^{k+\ell}) - G^{\varepsilon}_{k+\ell}]\bar{d}^{k+\ell}$$

$$= \lambda_{k+\ell}(d^{k+\ell})^t [G(x^*) + E_{k+\ell} - G(x^*) - E^{\varepsilon}_{k+\ell}]\bar{d}^{k+\ell}$$

$$= (x^{k+\ell+1} - x^{k+\ell})^t [E_{k+\ell} - E^{\varepsilon}_{k+\ell}]\ \bar{d}^{k+\ell}$$

In addition,

$$|g^t_{k+\ell+1}\ \bar{d}^{k+\ell}| < \| x^{k+\ell+1} - x^{k+\ell} \|\ \| E_{k+\ell} - E^{\varepsilon}_{k+\ell} \|\ \| \bar{d}^{k+\ell} \|$$

Using the fact that:

$$\| x^{k+\ell+1} - x^{k+\ell} \| < \| x^{k+\ell+1} - x^* \| + \| x^{k+\ell} - x^* \|$$

and from lemma 12-i we have:

$$|g^t_{k+\ell+1}\bar{d}^{k+\ell}| < 2\ \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}} \| x^k - x^* \|\ \| E_{k+\ell} - E^{\varepsilon}_{k+\ell} \|$$

which completes the proof of part i).

ii) To show this part we write g_{k+q_k} as:

$$g_{k+q_k} = (g_{k+q_k} - g_{k+q_k-1}) + (g_{k+q_k-1} - g_{k+q_k-2}) + \dots +$$

$$(g_{k+\ell+2} - g_{k+\ell+1}) + g_{k+\ell+1}$$

Multiplying through by $\bar{d}^{k+\ell}$, we get:

$$g^t_{k+q_k}\ \bar{d}^{k+\ell} = (g_{k+q_k} - g_{k+q_k-1})^t\ d^{k+\ell} + (g_{k+q_k-1} - g_{k+q_k-2})^t\ \bar{d}^{k+\ell} +$$

$$\dots + g^t_{k+\ell+1}\ \bar{d}^{k+\ell}$$

$$= \sum_{i=\ell+2}^{q_k} (g_{k+i} - g_{k+i-1})^t\ \bar{d}^{k+\ell} + g^t_{k+\ell+1}\ \bar{d}^{k+\ell}$$

$$= \sum_{i=\ell+2}^{q_k} (g_{k+i} - g_{k+i-1})^t\ \bar{d}^{k+\ell}\ \frac{\|\lambda_{k+i-1} d^{k+i-1}\|}{\|\lambda_{k+i-1} d^{k+i-1}\|} + g^t_{k+\ell+1}\bar{d}^{k+\ell}$$

$$= \sum_{i=\ell+2}^{q_k} \|x^{k+i} - x^{k+i-1}\|\ y^t_{k+i}\ \bar{d}^{k+\ell} + g^t_{k+\ell+1}\ \bar{d}^{k+\ell}$$

since,

$$\frac{(g_{k+i} - g_{k+i-1})}{\|\lambda_{k+i-1} d^{k+i-1}\|} = y_{k+i} \quad \text{and} \quad \|\lambda_{k+i-1} d^{k+i-1}\| = \|x^{k+1} - x^{k+i-1}\|$$

Now, we note that:

$$\|x^{k+i} - x^{k+i-1}\| \leq \|x^{k+i} - x^*\| + \|x^{k+i-1} - x^*\|$$

and using lemma 12-i, we get:

$$g^t_{k+q_k} \bar{d}^{k+\ell} \leq \sum_{i=\ell+2}^{q_k} (\|x^{k+1} - x^*\| + \|x^{k+i-1} - x^*\|)\, y^t_{k+i} \bar{d}^{k+\ell} + g^t_{k+\ell+1} \bar{d}^{k+\ell}$$

$$\leq \sum_{i=\ell+2}^{q_k} \left(2 \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}} \|x^k - x^*\|\right) y^t_{k+i} \bar{d}^{k+\ell} + y^t_{k+\ell+1} \bar{d}^{k+\ell}$$

Finally, using lemma 11-ii) and part i) of this lemma, we have:

$$|g^t_{k+q_k} \bar{d}^{k+\ell}| \leq 2 \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}} \|x^k - x^*\| \sum_{i=\ell+2}^{q_k} \|(\bar{d}^{k+i-1})^t (E_{k+i-1} - E_{k+\ell}) \bar{d}^{k+\ell}\|$$

$$+ 2 \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}} \|x^k - x^*\| \, \|E_{k+\ell} - E^{\varepsilon}_{k+\ell}\|$$

$$\leq 2 \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}} \|x^k - x^*\| \left[\sum_{i=\ell+2}^{q_k} (\|E_{k+i-1} - E_{k+\ell}\|) + \|E_{k+\ell} - E^{\varepsilon}_{k+\ell}\| \right]$$

which completes the proof of part ii).

iii) To show this part consider the Kuhn-Tucker conditions of the $(k+q_k)$th projection problem. This problem is the last one solved before restarting:

$$d^{k+q_k} + \alpha_{k+q_k} g_{k+q_k} + \sum_{i=1}^{r} [u_{k+q_k}]_i a_i + \sum_{i=r+1}^{m} [u_{k+q_k}]_i a_i$$

$$+ \sum_{i=1}^{q_k} [v_{k+q_k}]_i y_{k+i} = 0$$

Then a necessary condition for restarting is that $x_p^{k+q_k} = x^{k+q_k}$ and $Y_{k+q_k} \neq \phi$, hence $d^{k+q_k} = 0$. Also, using lemma 8 we get:

$$\alpha_{k+q_k} g_{k+q_k} + \sum_{i=1}^{r} [u_{k+q_k}]_i \; a_i + \sum_{i=1}^{q_k} [v_{k+q_k}]_i \; y_{k+i} = 0$$

Multiplying through by $\bar{d}^{k+\ell}$, we have:

$$\alpha_{k+q_k} g_{k+q}^t \bar{d}^{k+\ell} + \sum_{i=1}^{r} [u_{k+q_k}]_i a_i^t \bar{d}^{k+\ell} + \sum_{i=1}^{q_k} [v_{k+q_k}]_i y_{k+i}^t \bar{d}^{k+\ell} = 0$$

Since by theorem 5 the second term is equal to zero, we have:

$$\sum_{i=1}^{q_k} [v_{k+q_k}]_i y_{k+i}^t \bar{d}^{k+\ell} = - \alpha_{k+q_k} g_{k+q_k}^t \bar{d}^{k+\ell} \qquad (5.9)$$

Or, equivalently:

$$\sum_{i=1}^{\ell} [v_{k+q_k}]_i \; y_{k+i}^t \; \bar{d}^{k+\ell} + [v_{k+q_k}]_{\ell+1} \; y_{k+\ell+1}^t \; \bar{d}^{k+\ell}$$

$$+ \sum_{i=\ell+2}^{q_k} [v_{k+q_k}]_i \; y_{k+i}^t \; \bar{d}^{k+\ell} = - \alpha_{k+q_k} \; g_{k+q_k}^t \; \bar{d}^{k+\ell}$$

Since, by construction, $y_{k+i}^t \bar{d}^{k+\ell} = 0$ for $i=1,\ldots,\ell$, we obtain:

$$[v_{k+q_k}]_{\ell+1} y_{k+\ell+1}^t \bar{d}^{k+\ell} = - \sum_{i=\ell+2}^{q_k} [v_{k+q_k}]_i y_{k+i}^t \bar{d}^{k+\ell} - \alpha_{k+q_k} g_{k+q_k} \bar{d}^{k+\ell}$$

Now, from lemma 11-i), $y_{k+\ell+1}^t \bar{d}^{k+\ell} \geq \eta$

And thus:

$$\eta [v_{k+q_k}]_{\ell+1} \leq [v_{k+q_k}]_{\ell+1} y_{k+\ell+1}^t \bar{d}^{k+\ell} = - \sum_{i=\ell+2}^{q_k} [v_{k+q_k}]_i y_{k+i}^t \bar{d}^{k+\ell}$$

$$- \alpha_{k+q_k} \; g_{k+q_k}^t \; \bar{d}^{k+\ell}$$

Consequently,

$$|[v_{k+q_k}]_{\ell+1}| \leq \left(\frac{1}{\eta}\right) \, ||\alpha_{k+q_k} g_{k+q_k}^t \bar{d}^{k+q}| + |\sum_{i=\ell+2}^{q_k} [v_{k+q_k}]_i y_{k+i}^t \bar{d}^{k+\ell}|$$

Using the expression for $y_{k+i}^t \; \bar{d}^{k+\ell}$ from lemma 11-ii) and noting that $\alpha_{k+q_k} \leq \alpha_1$, we have:

$$|[v_{k+q_k}]_{\ell+1}| < \left(\frac{1}{\eta}\right) [\alpha_1 \mid g^t_{k+q_k} \bar{d}^{k+\ell} \mid +$$

$$+ \sum_{i=\ell+2}^{q_k} |[v_{k+q_k}]_i| (\|E_{k+i-1} - E_{k+\ell}\|)] \qquad (5.10)$$

Now, we observe the following:

$$\sum_{i=\ell+2}^{q_k} |[v_{k+q_k}]_i| (\|E_{k+i-1} - E_{k+\ell}\|) =$$

$$[|v_{k+q_k}|_1, \ldots, |v_{k+q_k}|_{\ell+1}, |v_{k+q_k}|_{\ell+2}, \ldots, |v_{k+q_k}|_{q_k}] \begin{bmatrix} 0 \\ \vdots \\ \|E_{k+\ell+1} - E_{k+\ell}\| \\ \|E_{k+\ell+2} - E_{k+\ell}\| \\ \vdots \\ \|E_{k+q_k-1} - E_{k+\ell}\| \end{bmatrix}$$

$< \|v_{k+q}\| \cdot \delta_{k+q_k}$, where δ_{k+q_k} is the vector with components

$\|E_{k+i-1} - E_{k+\ell}\|$. (5.11)

Using (5.11) in (5.10) we have:

$$|[v_{k+q_k}]_{\ell+1}| < \left(\frac{1}{\eta}\right) |\alpha_1 \mid g^t_{k+q_k} \bar{d}^{k+\ell}| + \|v_{k+q_k}\| \delta_{k+q_k}| \qquad (5.12)$$

Finally, using part ii) of this lemma, and observing that (5.12) holds for all $0 < \ell < q_k$ we have:

$$\| v_{k+q_k} \| < \frac{\alpha_1}{\eta} \mid g^t_{k+q_k} \bar{d}^{k+\ell} \mid + \left(\frac{1}{\eta}\right) \| v_{k+q_k} \| \delta_{k+q_k}$$

or, equivalently:

$$\| v_{k+q_k} \| < \frac{\left(\frac{\alpha_1}{\eta}\right) |g_{k+q_k} \bar{d}^{k+\ell}|}{|1 - \left(\frac{1}{\eta}\right) \delta_{k+q_k}|}$$

By definition $\delta_{k+q_k} \to 0$ as $k \to \infty$ and therefore the denominator goes to 1. It follows that:

$$\| v_{k+q} \| = O(|g^t_{k+q_k} \bar{d}^{k+\ell}|)$$

and the desired result follows from part ii) of the lemma.

iv) The proof of this part follows directly from part iii) above, since:

$$\| \sum_{i=1}^{q_k} [v_{k+q_k}]_i \, y_{k+1} \| < \sum_{i=1}^{q_k} |[v_{k+q_k}]_i| \, \| y_{k+i} \| \qquad (5.13)$$

and observing that:

$$\| y_{k+i} \|^2 = \frac{(g_{k+i} - g_{k+i-1})^t \, y_{k+i}}{\| \lambda_{k+i-1} d^{k+i-1} \|} = \frac{\lambda_{k+i-1} (d^{k+i-1})^t G(\zeta^{k+i-1}) y_{k+i}}{\| \lambda_{k+i-1} d^{k+i-1} \|}$$

$$= (\bar{d}^{k+i-1})^t G(\zeta^{k+i-1}) y_{k+i} < \| \bar{d}^{k+i-1} \| \, \| G(\zeta^{k+i-1}) y_{k+i} \| < G(\zeta^{k+i-1}) \| \, \| y_{k+i} \|$$

which gives:

$$\|y_{k+i}\| < \|G(\zeta^{k+i-1})\| < \Delta \tag{5.14}$$

where Δ is a bound on the norm of $G(\zeta^{k+i-1})$ and $\zeta^{k+i-1} \in N_{\varepsilon_0}(x^*)$.

Using (5.14) in (5.13) we finally get:

$$\|\sum_{i=1}^{q_k} [v_{k+q_k}]_i y_{k+i}\| < \Delta \|v_{k+q_k}\|$$

and therefore:

$$\|\sum_{i=1}^{q_k} [v_{k+q_k}]_i y_{k+i}\| = O(\|v_{k+q_k}\|)$$

which completes the proof of this lemma.

We now have all the ingredients for superlinear convergence. It will be seen that the reduction in the error from the point x^* is of the order of the length of the dual vector associated with the conjugacy contraints. This vector is based on second order information and will go to zero as fast as $E(\cdot)$, the error in the approximation of the Hessian matrix at x^*.

Theorem 7. Let f be continuously differentiable and bounded below on the bounded set S. In addition, let assumption A be satisfied. Then we have:

i) $1 < q_k < n-r$, for k large enough

ii) $\lim_{k\to\infty} \frac{\|x^{k+q_k} - x^*\|}{\|x^k - x^*\|} = 0$

Proof. i) From theorem 5, there is k_0 such that for $k \geq k_0$, the set of binding constraints will not change and we have:

$$a_i^t x^k = b_i, \; i=1,\dots,r$$

Since $\{a_1,\dots,a_r\}$ are linearly independent, the rank of $\bar{A}$ is equal to r. This implies that at most (n-r) conjugate direction steps will be performed before the system

$$\bar{A}x^k = b$$
$$Y_k x^k = e_k$$

is inconsistent. In addition, since $\{a_1,\dots,a_r\}$ are othogonal to d_i, $i=k+1,\dots,k+q_k$, $\{a_1,\dots,a_r, d^{k+1},\dots,d^{k+q_k}\}$ form a set of linearly independent vectors in E^n and thus: $r+q_k \leq n \Rightarrow q_k \leq n-r$. Also, if $q_k = 0$ conjugate directions are constructed, so $q_k \geq 1$. It follows then that:

$$1 \leq q_k \leq n-r$$

ii) To prove this part we need to establish several intermediate results. First, we show that for k large enough we have

$$\|\nabla f(x^{k+q_k}) - \nabla f(x^*)\| = O(\|v_{k+q_k}\|)$$

From (5.6) and (5.7) in lemma 12 we have:

$$\|\nabla f(x^{k+q_k}) - \nabla f(x^*)\| < \left(\frac{2\Delta}{m}\right) \|\nabla f(x^{k+q_k}) + r_{k+q_k}\| \tag{5.15}$$

where $\quad r_{k+q_k} \in \{r \mid r = \sum_{i=1}^{r} t_i a_i\}$

Then, to get the desired result we observe the following from the Kuhn-Tucker conditions of the $(k+q_k)$th projection problem:

$$\alpha_{k+q_k} \nabla f(x^{k+q_k}) + \sum_{i=1}^{r} [u_{k+q_k}]_i a_i + \sum_{i=1}^{q_k} [v_{k+q_k}]_i y_{k+i} = 0 \; y_{k+i} = 0$$

or,

$$\nabla f(x^{k+q_k}) + r_{k+q_k} = -\sum_{i=1}^{q_k} \frac{[v_{k+q}]_i}{\alpha_{k+q_k}} y_{k+i}$$

$$\|\nabla f(x^{k+q_k}) + r_{k+q_k}\| = \left\|\sum_{i=1}^{q_k} \frac{[v_{k+q_k}]_i}{\alpha_{k+q_k}} y_{k+i}\right\|$$

Now, using (5.15) we have:

$$\|\nabla f(x^{k+q_k}) - \nabla f(x^*)\| \le \left(\frac{2\Delta}{m_1}\right)\|\nabla f(x^{k+q_k}) + r_{k+q_k}\| \le \left(\frac{2\Delta}{m_1\alpha_0}\right) \|\sum_{i=1}^{q_k} [v_{k+q_k}]_i\, y_{k+i}\|$$

and using part iv) of lemma 13 we have:

$$\|\nabla f(x^{k+q_k}) - \nabla f(x^*)\| = O(\|v_{k+q_k}\|) \tag{5.16}$$

Next we show that:

$$\frac{\|v_{k+q_k}\|}{\nabla f(x^k) - \nabla f(x^*)\|} \to 0 \quad \text{for k large enough.}$$

To do that, we use Taylor's theorem to get:

$$[\nabla f(x^k) - \nabla f(x^*)]^t(x^k - x^*) = (x^k - x^*)^t G(\zeta^k)(x^k - x^*)$$

$$\ge \frac{m_1}{2}\|x^k - x^*\|^2 \quad \text{for k large enough.}$$

Therefore:

$$\frac{m_1}{2}\|x^k - x^*\|^2 \le [\nabla f(x^k) - \nabla f(x^*)]^t(x^k - x^*) \le \|\nabla f(x^k) - \nabla f(x^*)\|\,\|x^k - x^*\|$$

$$\Rightarrow \quad \|\nabla f(x^k) - \nabla f(x^*)\| \ge \left(\frac{m_1}{2}\right)\|x^k - x^*\| \tag{5.17}$$

Now, using (5.17) and part iii) of lemma 13:

$$\frac{\|v_{k+q_k}\|}{\|\nabla f(x^k) - \nabla f(x^*)\|} < \frac{\|v_{k+q_k}\|}{(\frac{m_1}{2})\|x^k - x^*\|}$$

$$= \frac{O(\|x^k - x^*\| [\sum_{i=\ell+}^{q_k} (\|E_{k+i-1} - E_{k+\ell}\|) + \|E^{\varepsilon}_{k+\ell}\|])}{(\frac{m_1}{2}) \|x^k - x^*\|}$$

$$= O(\sum_{i=\ell+2}^{q_k} (\|E_{k+i-1} - E_{k+\ell}\|) + \|E_{k+\ell} - E^{\varepsilon}_{k+\ell}\|) \qquad (5.18)$$

The last expression goes to zero as k goes to infinity which produces the desired result.

Similarly to (5.17) and (5.7) we have:

$$\|\nabla f(x^{k+q_k}) - \nabla f(x^*)\| \geq (\frac{m_1}{2}) \|x^{k+q_k} - x^*\|$$

$$\|\nabla f(x^k) - \nabla f(x^*)\| \leq \Delta_1 \|x^k - x^*\|$$

which gives:

$$\frac{\|x^{k+q_k} - x^*\|}{\|x^k - x^*\|} < \left(\frac{2}{m_1}\right) \frac{\|\nabla f(x^{k+q_k} - \nabla f(x^*)\|}{\|x^k - x^*\|} < \frac{\left(\frac{2}{m_1}\right)\|\nabla f_{k+q_k} - \nabla f(x^*)\|}{\left(\frac{1}{\Delta_1}\right)\|\nabla f(x^k) - \nabla f(x^*)\|}$$

Now, using (5.16) and (5.18) we get:

$$\frac{\|x^{k+q_k} - x^*\|}{\|x^k - x^*\|} < \left(\frac{2\Delta_1}{m_1}\right) \frac{\|\nabla f^{k+q_k} - \nabla f(x^*)\|}{\|\nabla f(x^k) - \nabla f(x^*)\|} = \frac{O(\|v_{k+q_k}\|)}{\|\nabla f(x^k) - \nabla f(x^*)\|}$$

$$= O\left(\sum_{i=\ell+2}^{q_k} (\|E_{k+i-1} - E_{k+\ell}\| + \|E_{k+\ell} - E^{\varepsilon}_{k+\ell}\|\right) \qquad (5.19)$$

As k gets large enough E_{k+i-1}, $E_{k+\ell}$, $E^{\varepsilon}_{k+\ell} \to 0$ and therefore:

$$\lim_{k\to\infty} \frac{\|x^{k+q_k} - x^*\|}{\|x^k - x^*\|} = 0 .$$

which completes the proof of the theorem.

As a consequence of theorem 7 we can achieve a quadratic rate of convergence under a Lipschitz condition on $G(x^*)$.

Corollary. Under the assumptions of theorem 7 suppose there exists $L > 0$ such that for $\zeta \in N(x^*)$ we have:

$$\|G(\zeta) - G(x^*)\| < L\,\|\zeta - x^*\|$$

where $\zeta^k \in L(x^k, x^{k+1})$

Then the following will hold:

$$\lim_{k\to\infty} \frac{\|x^{k+q_k} - x^*\|}{\|x^k - x^*\|^2} = \text{constant}.$$

Proof. By definition, $E_k = G(\zeta^k) - G(x^*)$ and

$$\|E_k\| \leq \|G(\zeta^k) - G(x^*)\| \leq L\|\zeta^k - x^*\|$$

In addition:

$$\|\zeta^k - x^*\| = \|tx^k + (1 - t)x^{k+1} - x^*\|, \quad \text{for} \quad 0 \leq t \leq 1$$

$$= \|tx^k + (1 - t)x^{k+1} - tx^* + (1 - t)x^*\|$$

$$= \|t(x^k - x^*) + (1 - t)\ (x^{k+1} - x^*)\|$$

$$\leq t\|x^k - x^*\| + (1 - t)\ \|x^{k+1} - x^*\|$$

$$\leq \|x^k - x^*\| + \|x^{k+1} - x^*\|$$

Thus:

$$\|E_k\| \leq L(\|x^k - x^*\| + \|x^{k+1} - x^*\|) \tag{5.20}$$

Now, from theorem 7 using (5.19) we have:

$$\frac{\|x^{k+q_k} - x^*\|}{\|x^k - x^*\|} = O\left(\sum_{i=\ell+2}^{q_k} [\|E_{k+i-1} - E_{k+\ell}\|] + \|E_{k+\ell} - E^{\varepsilon}_{k+\ell}\|\right)$$

$$= O\left(\sum_{i=\ell+2}^{q_k} \|E_{k+i-1} - E_{k+\ell}\|\right) = O\left(\sum_{i=\ell+2}^{q_k} \|E_{k+i-1}\| + (n-r)\|E_{k+\ell}\|\right)$$

Using (5.20), we have:

$$\frac{\|x^{k+q_k} - x^*\|}{\|x^k - x^*\|^2} = O\left(\sum_{i=\ell+2}^{q_k} L\,\frac{\|x^{k+i-1} - x^*\|}{\|x^k - x^*\|} + \frac{\|x^{k+i} - x^*\|}{\|x^k - x^*\|}\right.$$

$$\left. + L\left(\frac{\|x^{k+\ell} - x^*\|}{\|x^k - x^*\|} + \frac{\|x^{k+\ell-1} - x^*\|}{\|x^k - x^*\|}\right)\right)$$

$$= O\left(\sum_{i=\ell+2}^{q_k} L\left(\left[\frac{m_2}{m_1}\right]^{\frac{1}{2}} + \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}}\right) + L\left(\left[\frac{m_2}{m_1}\right]^{\frac{1}{2}} + \left[\frac{m_2}{m_1}\right]^{\frac{1}{2}}\right)\right)$$

$$= O\left(2(n-r+1)\,L\left[\frac{m_2}{m_1}\right]^{\frac{1}{2}}\right) = \text{constant}$$

Therefore:

$$\|x^{k+q_k} - x^*\| = O(\|x^k - x^*\|^2)$$

and the proof is complete.

CHAPTER VI

COMPUTATIONAL RESULTS

1. Introduction

In this chapter we examine the performance of the basic algorithm developed in chapter three on numerical test problems. The main numerical features we will consider are reliability, efficiency and sensitivity. Reliability is concerned with convergence to a solution point and its accuracy. Efficiency is measured in terms of total number of iterations, function and gradient evaluations, and computing time. Sensitivity will be evaluated by using different values for certain key parameters for a test problem. To accomplish this task, thirty test problems, from two to one hundred variables in size, are selected from the literature of nonlinear programming.

The algorithm was coded in Fortran V and the computations were performed in single-precision on a CDC Cyber 70 Model 7428/CDC 6400.

In the next section we discuss relevant aspects of coding and implementation of the algorithm. In section three we present the numerical results and their comparison to other published works in the field. In Appendix A the thirty test problems are given in detail. However, in Table 6.1 below we give a summary of the main characteristics of each problem.

Table 6-1: A Summary of Test Problems						
PROBLEM	NUMBER OF VARIABLES	OBJECTIVE	CONSTRAINTS			SOURCE
			Equality	Inequality	Bounds	
1	2	Quadratic		2	2	Bazaraa & Shetty (1979)
2	2	Quartic		1		Hock (1981)
3	2	Cubic		2		Hock (1981)
4	2	Trig.	1			Hock (1981)
5	2	Quadratic		1	4	Hock (1981)
6	3	Nonlinear (exp-log.)			6	Himmelblau (1972)
7	3	Quadratic	1			Hock (1981)
8	3	Quadratic		1	3	Hock (1981)
9	3	Nonlinear		1	6	May (1979)
10	3	Nonlinear		2	6	Hock (1981)
11	3	Nonlinear (log.)	1		6	Hock (1981)
12	4	Quartic			8	Himmelblau (1972)
13	4	Quadratic		6	4	Hock (1981)
14	4	Nonlinear	1		8	Hock (1981)
15	4	Quadratic		3	4	Hock (1981)
16	4	Quadratic	2		4	Luenberger (1973)
17	5	Cubic		10	5	Himmelblau (1972)
18	5	Nonlinear			10	Hock (1981)

Table 6-1: (Cont.)						
PROBLEM	NUMBER OF VARIABLES	OBJECTIVE	CONSTRAINTS			SOURCE
			Equality	Inequality	Bounds	
19	5	Quadratic	2			Hock (1981)
20	5	Nonlinear	3		20	Hock (1981)
21	6	Nonlinear (exp.)	6		8	Hock (1981)
22	8	Nonlinear (exp.)		1	16	Bracken, McCormick (1968)
23	10	Nonlinear (log.)			20	Himmelblau (1972)
24	10	Nonlinear (log.)	3		10	Himmelblau (1972)
25	11	Cubic		10	22	Day (1979)
26	14	Nonlinear (exp.)	4	5	28	Kezouh (1983)
27	15	Quadratic		29	30	Hock (1981)
28	16	Quartic	8		32	Himmelblau (1972)
29	45	Nonlinear (log.)	16		45	Himmelblau (1972)
30	100	Nonlinear		12	100	Himmelblau (1972)

2. Coding and Implementation

The proposed method was coded in Fortran V. The code consisted of a main program and the following subroutines:

i) Function evaluation: Provides numerical values of analytically specificed objective functions.

ii) Gradient evaluation: At each iteration, this finite differencing routine provides an approximation to the gradient vector.

iii) Linear Complementary Pivoting: This routine provides the dual solution of the projection problem from which the direction vector is computed. It includes determining variables to become basic, minimum ratio test, and pivoting.

iv) Orthogonal Factorization: This routine computes the QR factorization of the matrix of conjugacy constraints.

v) Projection Updating: When a new conjugacy requirement is added to the projection problem, this routine updates the operator:

$$P_k = [I - Y_k^t(Y_k Y_k^t)^{-1} Y_k]$$

The updating is performed using the recursive formula:

$$P_{k+1} = P_k - \frac{P_k y_k y_k^t P_k}{y_k^t P_k y_k}$$

vi) Approximate Line Search: This routine computes the Armijo number ν_k at each iteration giving the step-size $\lambda_k = (\beta_k)^{\nu_k} w_k$.

vii) Mukai's Approximation: This routine is used close to a solution point to give an initial approximation a_k to the step-size at each iteration. This approximation is based on the current local quadratic approximation.

viii) Slack Variables: This routine is used to evaluate the constraints and determine the amount of slack $s_k = b - Ax_k$.

Parameters

For the unconstrained step:

$$x_u^k = x^k - \alpha_k \nabla f(x^k)$$

a value of α_k needs to be specified in the range $[\alpha_0, \alpha_1]$, $\alpha_0 > 0$. For all the problems solved, it was determined empirically that $\alpha_k \in [0.1, 3.0]$. The following values of α_k gave the best solution for each problem:

α_k	% Problems
0.25	73%
0.50	23%
1.00	3%

Other parameters used were the constrained step-size parameters: $\sigma_k = 1/3$, $\beta_k = 1/2$ for all $k > 0$. The initial approximation a_k was always set to the value 1.0 whenever Mukai's approximation was not used.

Restarting

A restarting iteration is initiated if $\|d^k\| < 10^{-4}$. The slightly lower accuracy used here is due to the fact that we want to avoid generating steps that are nearly parallel, which introduces a lot of ill-conditioning in the matrix Y_k.

Termination Criteria

The problem is terminated if $\|d^k\| < 10^{-5}$ and $J_k = 0$. This result in appriximately five significant digits of accuracy in the values of $f(x^k)$ and x^k.

Line Search Scheme

As described in section 3 of chapter 3, the line search is performed until the first nonnegative integer ν_k satisfies:

$$f[x^k + (\beta_k)^{\nu_k} w_k d^k] - f(x^k) < \sigma_k (\beta_k)^{\nu_k} w_k \frac{\|d^k\|^2}{2\alpha_k} \qquad (2.1)$$

which results in a step-size $\lambda_k = (\beta_k)^{\nu_k} w_k$. The search starts with $\nu_k = 0$, or equivalently, $\lambda_k = 1.0$. The value of $f(x^k + \lambda_k d^k)$ is computed and the search is continued until the inequality (2.1) is satisfied, at which time the search is terminated. Otherwise, λ_k is halved and the process is repeated.

The Initial Approximation

The following initial approximation, discussed in section 3.3 of Chapter 4 is used when x_k is close to a stationary point:

$$\text{Compute} \quad \tau_k = 2[f(x^k + \varepsilon d^k) - f(x^k) - \varepsilon \nabla f(x^k)^t d^k]$$

$$\text{and let} \quad a_k = \begin{cases} \dfrac{-\varepsilon^2 \nabla f(x^k)^t d}{\tau_k}, & \text{if } \tau_k > \varepsilon^2 \delta_k \|d^k\|^2 \\ 1, & \text{otherwise} \end{cases}$$

$$\text{then set } w_k = \min\{a_k, 1\}$$

In case $a_k = 1$, δ_k is halved for the purposes of the next iteration. It was determined empirically that the values for the parameter ε and δ_0 were acceptable in the following ranges:

$$\varepsilon \in [10^{-4}, 10^{-3}]; \ \delta_0 \in [0.5, 2]$$

Since it was shown that this initial approximation is only necessary when x^k is close to a stationary point, it was decided not to use it in the early iterations. However, since all the problems solved had known optimal objective function values, some experimentation resulted in the following rule: Initiate Mukai's approximation when $f(x^k)$ is within 10% of the optimal value f*. More specifically, when:

$$|f(x^k) - f^*| < \Delta = \begin{cases} 10\% \; f^* \\ 10\% \; |f(x^0) - f^*| \end{cases}$$

Gradient Approximation

To avoid the need for having to provide analytical derivatives of f(x), the gradient vector is approximated at each iteration by the forward-difference formula:

$$\nabla f(x^k) \simeq \frac{f(x^k + h) - f(x^k)}{h}$$

where h = 0.0001.

The Projection Operator

As discussed in Chapter 3, the proposed method requires the operator:

$$P_k = [I - Y_k^t(Y_k Y_k^t)^{-1} \; Y_k]$$

which projects any vector in E^n into the null space of Y_k, the matrix of conjugacy requirements.

For a general iteration, the updating in performed by the formula:

$$P_{k+1} = P_k - \frac{P_k y_k y_k^t P_k}{y_k^t P_k y_k} \qquad (2.2)$$

When this updating fails an orthogonal factorization of Y_k is performed which is given by:

$$Y_k = [R_k^t | 0] \left[\frac{Q_1^{(k)}}{Q_2^{(k)}} \right]$$

and then $P_k = Q_2^{(k)t} \cdot Q_2^{(k)}$

If this process fails, the procedure is restarted with P_k reinitialized to the identy matrix. In addition, when equality constraints are present in the original problem in the form:

$$Dx = d$$

the initial matrix P_0 is initialized to:

$$P_0 = [I - D^t(DD^t)^{-1}D]$$

via a QR factorization. This is because this factorization allows the detection of dependencies among the rows of D.

3. Numerical Results

In this section the numerical solutions for the 30 test problems are presented with some relevant information on each problem. These results are then summarized and compared to published results from other methods. Finally, a test problem is used to show the sensitivity of the procedure to certain key parameters.

3.1 Solutions

The problems will be identified by their number given in Table 6.1. In addition, the detailed formulation of each problem is given in Appendix A for reference. For ease of presentation and readability the solutions are presented in tabular form by class of problems grouped according to the number of variables. The following statistics are provided for each problem:

a) The value of the objective function $f(x^*)$.

b) The solution vector x^*.

c) The number of iterations required to solve the problem.

d) The total number of restarts required.

e) The total number of functional evaluations.

f) The total number of gradient evaluations.

g) The number of conjugate directions constructed. These are broken down into:

   i) The maximum number of conjugate directions constructed between restarts.

   ii) The minimum number of conjugate directions constructed between restarts.

h) Line search statistics. These include:

   i) The maximum number of trials, corresponding to the largest value of ν_k, required per iteration.

   ii) The minimum number of trials, corresponding to the smallest value of ν_k, required per iteration.

i) For the computation of the direction vector, Lemke's linear complementary pivoting routine is used. Two statistics are given:

   i) The maximum number of pivots required per iteration.

   ii) The minimum number of pivots required per iteration.

j) The number of binding constraints at optimality.

l) The execution time required in CPU seconds.

m) The value of the unconstrained step-size parameter α_k.

Tables (6-2) to (6-14) contain these statistics for each group of problems. Table (6-15) gives an average picture of the performance of the algorithms on each class of problems. These averages are computed based on the number of problems solved in each class, which is indicated in paratheses in column one.

Table 6-2: Solution Summary for 2-Variable Problems

STATISTICS \ PROBLEM #		1	2	3	4	5
$f(x^*)$		-7.1601	0.0003	2.6666	-0.4998	-99.9590
x_1^* x_2^*		1.1191 0.7763	0.9998 0.9999	1.0000 0.0000	-2.9998 -4.0001	2.0000 -0.0001
VALUE OF α_k USED		0.25	0.25	0.25	0.25	0.25
NUMBER OF ITERATIONS		3	10	2	4	5
NUMBER OF RESTARTS		3	5	2	4	5
FUNCTIONAL EVALUATIONS		4	25	3	6	6
GRADIENT EVALUATIONS		3	10	2	4	5
CONJUGATE DIRECTIONS	MAX. #	1	2	1	1	1
	MIN. #	1	2	1	1	1
LINE SEARCH	MAX. ν_k	0	6	0	1	0
	MIN. ν_k	0	0	0	0	0
PIVOTS	MAX. #	1	0	2	0	1
	MIN. #	1	0	2	1	2
BINDING CONSTRAINTS		1	0	2	1	2
EXECUTION TIME		0.071	0.23	0.028	0.086	0.14

Table 6-3: Solution Summary for 3-Variable Problems

STATISTICS \ PROBLEM #		6	7	8	9	10	11
$f(x^*)$		0.00001	0.0002	0.1115	-3300.0000	-3455.9999	-26272.5143
x_1^* x_2^* x_3^*		50.0000 25.0001 1.4999	0.5000 -0.4999 0.4998	1.3121 0.7870 0.4500	20.0000 11.0000 15.0000	24.0002 11.9999 11.9999	0.6178 0.3281 0.0539
VALUE OF α_k USED		0.25	0.25	0.25	0.25	0.25	0.25
NUMBER OF ITERATIONS		5	5	4	2	3	8
NUMBER OF RESTARTS		4	5	3	1	2	8
FUNCTIONAL EVALUATIONS		8	6	5	3	5	34
GRADIENT EVALUATIONS		5	5	4	2	3	8
CONJUGATE DIRECTIONS	MAX. #	1	1	2	2	3	1
	MIN. #	0	1	1	0	0	1
LINE SEARCH	MAX. ν_k	2	0	0	0	1	7
	MIN. ν_k	0	0	0	0	0	1
PIVOTS	MAX. #	2	0	1	3	1	2
	MIN. #	1	0	1	3	1	1
BINDING CONSTRAINTS		0	0	1	4	1	0
EXECUTION TIME		0.214	0.125	0.105	0.094	0.138	0.83

Table 6-4: Solution Summary for 4-Variable Problems						
PROBLEM # / STATISTICS		12	13	14	15	16
$f(x^*)$		0.0001	-15.0000	1.9299	-4.6803	1.4099
x^*_1		0.9988	0.0000	0.6666	0.2727	1.1229
x^*_2		0.9998	3.0000	0.3333	2.0606	0.6509
x^*_3		1.0001	-0.0000	0.3333	0.0000	1.8289
x^*_4		1.0002	4.0000	2.0000	0.5454	0.5680
VALUE OF α_k USED		0.25	0.25	0.25	0.25	0.50
NUMBER OF ITERATIONS		20	14	5	9	2
NUMBER OF RESTARTS		6	5	4	4	2
FUNCTIONAL EVALUATIONS		32	15	7	10	3
GRADIENT EVALUATIONS		20	14	5	9	2
CONJUGATE DIRECTIONS	MAX. #	4	4	3	3	1
	MIN. #	0	0	0	1	0
LINE SEARCH	MAX. ν_k	8	0	1	0	0
	MIN. ν_k	0	0	0	0	0
PIVOTS	MAX. #	0	1	1	4	0
	MIN. #	0	0	1	0	0
BINDING CONSTRAINTS		0	4	1	2	2
EXECUTION TIME		0.995	1.165	0.173	0.956	0.06

Table 6-5: Solution Summary for 5-Variable Problems

STATISTICS \ PROBLEM #		17	18	19	20
$f(x^*)$		-32.3486	1.0001	0.0000	4.0899
x^*_1		0.3000	1.0000	1.0000	-0.7777
x^*_2		0.3335	1.9999	0.9999	0.2599
x^*_3		0.4000	2.9999	0.9999	0.6299
x^*_4		0.4284	4.0000	1.0000	-0.1198
x^*_5		0.2239	4.9000	1.0000	0.2599
VALUE OF α_k USED		0.25	0.25	0.25	0.25
NUMBER OF ITERATIONS		8	7	7	7
NUMBER OF RESTARTS		5	3	7	5
FUNCTIONAL EVALUATIONS		18	8	8	8
GRADIENT EVALUATIONS		8	7	7	7
CONJUGATE DIRECTIONS	MAX. #	2	2	1	2
	MIN. #	1	0	0	1
LINE SEARCH	MAX. ν_k	4	0	0	0
	MIN. ν_k	0	0	0	0
PIVOTS	MAX. #	9	4	0	2
	MIN. #	4	1	0	0
BINDING CONSTRAINTS		4	5	2	0
EXECUTION TIME		1.78	0.62	0.063	0.081

Table 6-6: Solution Summary for Problem 21			Table 6-7: Solution Summary for Problem 22	
$f(x^*)$		6.3333	$f(x^*)$	1138.4162
x_1^*		0.0000	x_1^*	0.4128
x_2^*		1.3333	x_2^*	0.4033
x_3^*		1.6666	x_3^*	131.2613
x_4^*		1.0001	x_4^*	164.3134
x_5^*		0.6666	x_5^*	217.4222
x_6^*		0.3333	x_6^*	12.2801
			x_7^*	15.7717
			x_8^*	20.7468
VALUE OF α_k USED		0.25		0.25
NUMBER OF ITERATIONS		8		13
NUMBER OF RESTARTS		8		9
FUNCTIONAL EVALUATIONS		12		19
GRADIENT EVALUATIONS		8		13
CONJUGATE DIRECTIONS	MAX. #	1		4
	MIN. #	0		1
LINE SEARCH	MAX. ν_k	2		3
	MIN. ν_k	0		0
PIVOTS	MAX. #	2		4
	MIN. #	1		2
BINDING CONSTRAINTS		8		0
EXECUTION TIME		1.653		1.935

Table 6-8: Solution Summary for 10-Variable Problems

STATISTICS \ PROBLEM #		23	24
$f(x^*)$		-45.7783	-47.7609
x_1^*		9.3503	0.0409
x_2^*		9.3503	0.1479
x_3^*		9.3502	0.7830
x_4^*		9.3503	0.0009
x_5^*		9.3503	0.4848
x_6^*		9.3503	0.0009
x_7^*		9.3502	0.0269
x_8^*		9.3502	0.0178
x_9^*		9.3503	0.0369
x_{10}^*		9.3503	0.0970
VALUE OF α_k USED		0.50	0.50
NUMBER OF ITERATIONS		14	16
NUMBER OF RESTARTS		13	16
FUNCTIONAL EVALUATIONS		21	27
GRADIENT EVALUATIONS		14	16
CONJUGATE DIRECTIONS	MAX. #	2	1
	MIN. #	0	0
LINE SEARCH	MAX. ν_k	4	5
	MIN. ν_k	0	0
PIVOTS	MAX. #	3	6
	MIN. #	1	1
BINDING CONSTRAINTS		0	2
EXECUTION TIME		2.01	2.33

Table 6-9: Solution Summary for Problem 25		
$f(x^*)$		37.6459
x_1^*		0.0099
x_2^*		0.0984
x_3^*		0.0631
x_4^*		0.40054
x_5^*		0.3881
x_6^*		0.1810
x_7^*		0.6499
x_8^*		0.3330
x_9^*		0.3409
x_{10}^*		0.1635
x_{11}^*		0.6499
VALUE OF α_k USED		0.50
NUMBER OF ITERATIONS		15
NUMBER OF RESTARTS		7
FUNCTIONAL EVALUATIONS		37
GRADIENT EVALUATIONS		15
CONJUGATE DIRECTIONS	MAX. #	3
	MIN. #	2
LINE SEARCH	MAX. ν_k	4
	MIN. ν_k	0
PIVOTS	MAX. #	8
	MIN. #	4
BINDING CONSTRAINTS		5
EXECUTION TIME		7.79

Table 6-10: Solution Summary for Problem 26		
$f(x^*)$		-1114.5475
x_1^*		3.4021
x_2^*		-0.1366
x_3^*		-0.0548
x_4^*		0.0955
x_5^*		0.0948
x_6^*		0.5869
x_7^*		0.1416
x_8^*		-0.7286
x_9^*		0.1512
x_{10}^*		-0.7516
x_{11}^*		-0.7338
x_{12}^*		0.4864
x_{13}^*		-1.5411
x_{14}^*		0.5411
VALUE OF α_k USED		0.25
NUMBER OF ITERATIONS		24
NUMBER OF RESTARTS		23
FUNCTIONAL EVALUATIONS		102
GRADIENT EVALUATIONS		24
CONJUGATE DIRECTIONS	MAX. #	2
	MIN. #	1
LINE SEARCH	MAX. ν_k	6
	MIN. ν_k	4
PIVOTS	MAX. #	1
	MIN. #	1
BINDING CONSTRAINTS		4
EXECUTION TIME		4.32

Table 6-11: Solution Summary for Problem 27		
$f(x^*)$		664.8213
x_1^*		7.9998
x_2^*		49.0001
x_3^*		3.000
x_4^*		0.9999
x_5^*		55.9989
x_6^*		0.0009
x_7^*		0.9999
x_8^*		63.0000
x_9^*		5.9980
x_{10}^*		3.0000
x_{11}^*		70.0016
x_{12}^*		11.9973
x_{13}^*		12.0000
x_{14}^*		77.9998
x_{15}^*		18.0001
VALUE OF α_k USED		0.50
NUMBER OF ITERATIONS		17
NUMBER OF RESTARTS		9
FUNCTIONAL EVALUATIONS		23
GRADIENT EVALUATIONS		17
CONJUGATE DIRECTIONS	MAX. #	5
	MIN. #	1
LINE SEARCH	MAX. ν_k	2
	MIN. ν_k	0
PIVOTS	MAX. #	10
	MIN. #	4
BINDING CONSTRAINTS		15
EXECUTION TIME		3.47

Table 6-12: Solution Summary for Problem 28		
$f(x^*)$		-244.8997
x_1^*		0.0389
x_2^*		0.7919
x_3^*		0.2028
x_4^*		0.8443
x_5^*		1.1269
x_6^*		0.9347
x_7^*		1.6819
x_8^*		0.1553
x_9^*		1.5678
x_{10}^*		0.0000
x_{11}^*		-0.0000
x_{12}^*		-0.0000
x_{13}^*		0.6602
x_{14}^*		0.0000
x_{15}^*		0.6742
x_{16}		0.0000
VALUE OF α_k USED		0.500
NUMBER OF ITERATIONS		12
NUMBER OF RESTARTS		12
FUNCTIONAL EVALUATIONS		50
GRADIENT EVALUATIONS		12
CONJUGATE DIRECTIONS	MAX. #	1
	MIN. #	1
LINE SEARCH	MAX. ν_k	4
	MIN. ν_k	0
PIVOTS	MAX. #	13
	MIN. #	8
BINDING CONSTRAINTS		5
EXECUTION TIME		5.12

Table 6-13: Solution Summary for Problem 29			
$f(x^*)$		-1911.40524	

x_1 = -.3497E-13	x_{16} = .8710E-13	x_{31} = .2300E-02
x_2 = .2139E+00	x_{17} = .1550E-01	x_{32} = .4730E-13
x_3 = .3595E+01	x_{18} = .2259E-13	x_{33} = .1794E-13
x_4 = .2466E+00	x_{19} = -.3064E-13	x_{34} = -.3458E-13
x_5 = .6330E+00	x_{20} = .5843E-13	x_{35} = .2429E-13
x_6 = .4572E-01	x_{21} = .6206E-13	x_{36} = -.4441E-15
x_7 = .3231E-13	x_{22} = .4291E-13	x_{37} = .9781E-13
x_8 = .1182E-12	x_{23} = .9281E-01	x_{38} = .1943E-15
x_9 = .2903E-13	x_{24} = .6304E-01	x_{39} = -.2109E-14
x_{10} = .7077E+00	x_{25} = .4830E-01	x_{40} = .1665E-02
x_{11} = .2496E-01	x_{26} = .2116E+02	x_{41} = .7335E-02
x_{12} = -.3775E-13	x_{27} = .3575E-13	x_{42} = -.3439E-13
x_{13} = .2559E+02	x_{28} = -.2912E-13	x_{43} = .1021E-12
x_{14} = .2216E-01	x_{29} = .2107E-13	x_{44} = .2934E-01
x_{15} = .8266E-13	x_{30} = .2110E-01	x_{45} = -.1474E-13

VALUE OF α_k USED		0.500
NUMBER OF ITERATIONS		62
NUMBER OF RESTARTS		55
FUNCTIONAL EVALUATIONS		178
GRADIENT EVALUATIONS		62
CONJUGATE DIRECTIONS	MAX. #	2
	MIN. #	1
LINE SEARCH	MAX. ν_k	7
	MIN. ν_k	0
PIVOTS	MAX. #	28
	MIN. #	14
BINDING CONSTRAINTS		41
EXECUTION TIME		265.50

Table 6-14: Solution Summary for Problem 30

f(x*)	-1732.0333

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	x_1	24				32	37	28	22						5			52		x_{99}
2	2	8	2	18	11										29	9	21			
3		9			29	62									35		17	25	62	60
4											9	39		58		44				
5	47 x_5	5 x_{10}	36	12		6					50	42		51		1				x_{100}

VALUE OF α_k USED		1.00
NUMBER OF ITERATIONS		272
NUMBER OF RESTARTS		192
FUNCTIONAL EVALUATIONS		498
GRADIENT EVALUATIONS		272
CONJUGATE DIRECTIONS	MAX. #	8
	MIN. #	1
LINE SEARCH	MAX. ν_k	4
	MIN. ν_k	0
PIVOTS	MAX. #	52
	MIN. #	4
BINDING CONSTRAINTS		70
EXECUTION TIME		912.59

(a) The values of the decision variables are rounded to the closest integer, since x_{ij} represents the number of weapons of type i assigned to target j.

Table 6-15: Average Results per Number of Variables

Statistics PB. Per # Var.	Average # Iterations	Average # Restarts	Average Maximum Conjugate Direction	Average Maximum Line Search α_k	Average Maximum Pivots	Average # Function Evaluations	Average # Gradient Evaluations	Average Execution Time
2(5)	4.8	3.8	1.2	1.4	0.8	8.8	4.8	0.11
3(6)	4.5	3.8	1.7	1.7	1.5	10.2	4.5	0.25
4(5)	10	4.2	3.0	1.8	1.2	13.4	10	0.67
5(4)	7.3	5.0	1.8	1.0	3.0	10.5	7.3	0.64
6(1)	8.0	8.0	1.0	2.0	2.0	12.0	8.0	1.65
8(1)	13.0	9.0	4.0	3.0	4.0	19.0	13.0	1.93
10(2)	15.0	14.5	1.5	4.5	4.5	24.0	15.0	2.17
11(1)	15.0	7.0	3.0	4.0	8.0	37.0	15.0	7.79
14(1)	24.0	23.0	2.0	6.0	1.0	102.0	24.0	4.32
15(1)	17.0	9.0	5.0	2.0	10.0	23.0	17.0	3.47
16(1)	12.0	12.0	1.0	4.0	13.0	50.0	12.0	5.12
45(1)	62.0	55.0	2.0	7.0	28.0	178.0	62.0	265.50
100(1)	272.0	192.0	8.0	4.0	52.0	498.0	272.0	912.59

3.2 Sensitivity

Some experimentation showed that the method is not affected by differnet starting points in terms of achieving the same optimal solution. This is true as long as the initial point is feasible. However, the method show some sensitivity to the unconstrained step size α_k and different strategies of reinitializing the projection matrix P_k.

Concerning α_k, it is seen that from a computational point of view this parameters acts as a sealing factor in the expression:

$$z_p^k = x^k - \alpha_k P_k \nabla f(x^k)$$

In addition, the performance of the line search is affected through the computation of ν_k:

$$\nu_k = \text{Min}\left\{ \begin{matrix} \nu > 0 \\ \text{integer} \end{matrix} \;\middle|\; f[x^k + (\beta_k)^\nu w_k d^k] - f(x^k) < -\sigma_k (\beta_k)^\nu w_k \frac{\|d^k\|^2}{2\alpha_k} \right\}$$

This effect of α_k is shown in Table 6-16 for the test problem 17. Figure 6-1 gives a graphical representation of different pertinent statistics as a function of α_k. It can be seen that for problem 17 the value $\alpha_k = 1.0$ gives the best solution overall.

Concerning different reinitialization strategies for the matrix P_k, it is seen in Table 6-16 that the best solution for problem 17 is achieved when P_k is reinitialized to the identity matrix. This corresponds to a fresh set of conjugate directions. Experimentation with restarting the procedure without discarding previously acquired directional information proved inferior. Table 6-16 shows that in most

Table 6-16: Results Showing Sensitivity to α_k and Type of Restarting for PB. 17

RESTARTING		FRESH RESTARTING: P_{k+1} REINITIALIZED TO I							
α_k		0.10	0.25	0.50	0.75	1.00	1.50	2.00	2.50
$f(x^*)$		-32.3486	-32.3486	-32.3486	-32.3486	-32.3486	-32.3486	-32.3486	-32.34861
x_1^*		0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
x_2^*		0.3335	0.3335	0.3335	0.3335	0.3335	0.3335	0.3335	0.3335
x_3^*		0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
x_4^*		0.4275	0.4275	0.4275	0.4275	0.4275	0.4275	0.4275	0.4275
x_5^*		0.2244	0.2247	0.2247	0.2247	0.2247	0.2247	0.2247	0.2247
NUMBER OF ITERATIONS		15	8	12	10	7	12	8	8
NUMBER OF RESTARTS		13	5	8	8	5	9	5	5
FUNCTIONAL EVALUATIONS		31	18	28	29	17	38	19	19
GRADIENT EVALUATIONS		15	8	12	10	7	12	8	8
CONJUGATE DIRECTIONS	MAX. #	2	2	2	2	2	2	2	2
	MIN. #	1	1	1	1	1	1	1	1
LINE SEARCH	MAX. ν_k	2	4	5	5	5	6	7	7
	MIN. ν_k	0	0	0	0	0	0	0	0
PIVOTS	MAX. #	7	9	9	9	9	9	13	13
	MIN. #	3	4	6	6	6	6	6	6
BINDING CONSTRAINTS		4	4	4	4	4	4	4	4
EXECUTION TIME		3.63	1.78	2.86	2.57	1.72	3.05	1.95	1.97

Table 6-16: Continued							
RESTARTING		(a) $P_{k+1} = P_k$		(b) $P_{k+1} = P_k$			
α_k		0.25	1.00	0.20	0.25	0.75	0.50
$f(x^*)$		-37.6332	-35.2505	-32.3486	-32.3486	-36.6003	-32.3486
x_1^*		0.3000	0.2250	0.3000	0.3000	0.2865	0.3000
x_2^*		0.5300	0.0590	0.3335	0.3335	0.5595	0.3333
x_3^*		0.4398	0.6134	0.4000	0.4000	0.3985	0.4000
x_4^*		0.6097	0.4859	0.4291	0.4284	0.5040	0.4279
x_5^*		0.2067	0.5514	0.2240	0.2239	0.1866	0.2244
NUMBER OF ITERATIONS		21	21	15	8	21	11
NUMBER OF RESTARTS		0	1	8	5	5	3
FUNCTIONAL EVALUATIONS		72	78	36	18	48	23
GRADIENT EVALUATIONS		21	21	15	8	21	11
CONJUGATE	MAX. #	3	3	5	2	5	5
DIRECTIONS	MIN. #	3	3	1	1	2	2
LINE	MAX. ν_k	6	7	5	4	6	7
SEARCH	MIN. ν_k	0	0	0	0	0	0
PIVOTS	MAX. #	14	14	9	9	11	10
	MIN. #	8	8	4	4	6	5
BINDING CONSTRAINTS		-	-	4	4	-	4
EXECUTION TIME		3.65	4.10	2.63	1.83	3.20	2.05

(a) Reinitialization occurs whenever $P_{ij_t} \geq 10^8$ or $y_k^t P_k y_k \leq 1.0E-10$

(b) Reinitialization occurs only when $y_k^t P_k y_k \leq 1.0E-10$

cases the method did not converge to the optimal solution. Finally, we need to mention that the updating formula for P_{k+1}:

$$P_{k+1} = P_k - \frac{P_k y_k y_k^t P_k}{y_k^t P_k y_k}$$

can cause some ill-conditioning when the term $y_k^t P_k y_k$ in the denominator becomes too small. Practically, each time this happened, the QR factorization was tried and if not successful, a restart was initialized.

Finally, we note that no special provisions for detecting dependencies and ill-conditioning were implemented. This accounts for the high percentage of restarts on most of the problems.

3.3 Comparison with Other Methods

In the last three tables, (6-17) - (6-19), we present a summary of the performance of the new method compared to six other methods for which published results exist. It is to be noted that not all problems were solved by all methods, and some of the relevant statistics are missing. In addition, these solutions are obtained on different computer systems. Except for the second method (BG) all others are the result of commercial codes. We also note that some of the execution times appear to be standardize times even though they were reported to be actual CPU seconds. From these results we can draw the following conclusions concerning the proposed method:

i) Reliability: Computational results clearly show that the method is reliable in the sense that all test problems were solved to optimality.

ii) Efficiency: In terms of number of iterations, function and gradient evaluations, the method seems to compare very well with other methods. In many cases its performance is superior in this regard. However, execution times are higher than for other methods. This seems to be an indication of more work per iteration. This can be partially explained by the fact that the method uses the entire constraint set at each iteration to solve the projection problem. In addition, special structures, such as lower and upper bounds, are not taken advantage of. Finally, to a lesser degree, the "rough" nature of the code may be a factor.

Table 6-17: Comparative Results: Function and Gradient Evaluations								
Problem #	N, M, NEQ	GRG	BG	BR	MS	MSH (c)	M (a)	New Method
6	3, 6, 0	6, 12	-	-	-	-	-	8, 5
10	3, 7, 0	-	-	-	-	-	48	5, 3
12	4, 8, 0	255, 43	-	-	-	-	462	32, 20
17	5, 15, 0	63, 9	30, 11	12, 12	9, 9	-	61	18, 8
23	10, 10, 3	77, 17	61, 17	65, 37	-	-	-	27, 16
24	10, 20, 0	32, 5	-	-	-	-	-	21, 14
25	11, 32, 0	-	-	-	-	-	512	37, 15
28	16, 32, 8	162, 37	24, 9	17, 15	16, 16	44	187	50, 12
29	45, 45, 16	229, 50	1105, 182	-	452, 452	513	-	178, 62
30	100, 112, 0	239, 41	1455, 485	208, 169	296, 296	310	-	498, 272

GRG: Generalized Reduced Gradient (IBM 370/145)
BG: Bazaraa and Goode (CDC Cyber 70 Model 74-28/CDC 6400)
BR: Best and Ritter (IBM 360-75)
MS: Murtagh and Saunders (IBM 370/168)
MSH: Marsten and Shanno (CDC Cyber 175)
M: May (DEC - 1077KI-10)
(a): May's method is a non-derivative method. The numbers given are function evaluations only.

Table 6-18: Comparative Results: Number of Iterations

Problem #	N, M, NEQ	GRG	BG	BR	MS	MSH	M (b)	New Method
6	3, 6, 0	10	-	-	-	-	-	5
10	3, 7, 10	-	-	-	-	-	-	3
12	4, 8, 0	43	-	-	-	-	-	20
17	5, 15, 0	9	11	11	8	-	-	8
23	10, 10, 3	9	17	36	-	-	-	16
24	10, 20, 0	16	-	-	-	-	-	14
25	11, 32, 0	5	-	-	-	-	-	15
28	16, 32, 8	36	9	14	16	26	-	12
29	45, 45, 16	49	182	-	103	274	-	62
30	100, 112, 0	40	485	168	133	170	-	272

(b) Not provided in published study.

(c) The numbers given are combined function and gradient evaluations.

Table 6-19: Comparative Results: Execution Times (in CPU seconds)								
Problem #	N, M, NEQ	GRG	BG	BR	MS	MSH	M	New Method
6	3, 6, 0	1.90	-	-	-	-	-	0.214
10	3, 7, 0	-	-	-	-	-	0.14	0.138
12	4, 8, 0	1.32	-	-	-	-	0.58	0.995
17	5, 15, 0	1.53	0.20	0.23	0.63	-	0.43	1.78
23	10, 10, 3	3.81	0.59	1.19	-	-	-	2.01
24	10, 20, 0	1.72	-	-	-	-	-	2.33
25	11, 32, 0	-	-	-	-	-	9.18	7.79
28	16, 32, 8	38.55	2.78	1.19	1.50	0.283	4.48	5.12
29	45, 45, 16	227.26	111.57	-	2.9	1.807	-	265.50
30	100, 112, 0	570.37	614.27	290.67	48.3	2.548	-	912.59

CHAPTER VII

CONCLUSIONS AND EXTENSIONS

The primary purpose of this thesis was to develop a method to solve the general linearly constrained nonlinear programming problem using conjugate directions. The computational results show that the method seems to compare well with the best existing methods in the field, such as the Generalized Reduced Gradient Method.

First, a reliable procedure is developed to produce a descent direction. The procedure is based on solving a projection problem which is strictly convex and requires a finite number of steps. This projection problem is designed to project an unconstrained descent direction onto the feasible region in such a way as to produce a feasible direction which is conjugate to previously constructed conjugate directions. This conjugacy property is aimed at producing a second order convergence of the algorithm.

Second, the line search procedure is an Armijo-type inexact line search which is based on the properties of the projection problem and produces a step-size with a finite number of function evaluations automatically. An initial step-size approximation is introduced close to a solution point. This approximation has the property that the exact step-size along the conjugate directions is closely approximated and that the Armijo number will eventually always equal zero.

Computational results show that on the average only two to three functional evaluations are required per iteration. In addition, other contributions were made in this research. Principally, we have established the global convergence of the procedure through the use of the inexact line search scheme and the convergence of the entire sequence under additional assumptions. We have also established the superlinear convergence of the procedure owing to the use of conjugate directions.

The following extensions are worthwhile investigating:

1. Modification of the proposed method to directly handle lower and upper bounded variables.
2. Further investigation of strategies to solve the projection problem and updating the projection operator to avoid ill-conditioning.
3. Extending the computational results by solving larger problems and duplicating the runs made so that the sensitivity of the procedure to certain key parameters can be better ascertained. In particular, the effect of changes in α_k on a larger set of problems will be a desirable study.
4. Computational testing of an accelerated and near-binding extensions to the basic algorithm.
5. Numerically more stable implementation of the basic algorithm so that the local convergence behavior can be better evaluated.

APPENDIX A

Test Problems

Problem 1

Source: Bazaraa and Shetty (1979), p. 366

Number of Variables: 2

Starting Point: $x_o = (0,0)$, $f(x_o) = 0.00$

Objective Function: $f(x) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2$

Constraints:
$$x_1 + x_2 \leq 2$$
$$x_1 + 5x_2 \leq 5$$
$$x_1, x_2 \geq 0$$

Problem 2

Source: Hock and Schittkowski (1981), p. 24

Number of Variables: 2

Starting Point: $x_o = (-2,1)$, $f(x_o) = 909.00$

Objective Function: $f(x) = 100\,(x_2 - x_1^2)^2 + (1 - x_1)^2$

Constraints: $-x_2 \leq 1.5$

Problem 3

Source: Hock and Schittkowski (1981), p. 27

Number of Variables: 2

Starting Point: $x_o = (1.125, 0.125)$, $f(x_o) = 3.3235$

Objective Function: $f(x) = 1/3(x_1 + 1)^3 + x_2$

Constraints: $-x_1 \leq -1$

$-x_2 \leq 0$

Problem 4

Source: Hock and Schittkowski (1981), p. 32

Number of Variables: 2

Starting Point: $x_o = (0.0, 0.0)$, $f(x_o) = 0.00$

Objective Function: $f(x) = \sin(\pi x_1/12) \cos(\pi x_2/16)$

Constraints: $4x_1 - 3x_2 = 0$

Problem 5

Source: Hock and Schittkowski (1981), p. 44

Number of Variables: 2

Starting Point: $x_o = (-1.0, -1.0)$, $f(x_o) = -98.99$

Objective Function: $f(x) = 0.01x_1^2 + x_2^2 - 100$

Constraints:

$$
\begin{aligned}
-10x_1 + x_2 &\leq 10 \\
x_1 \qquad &\leq 50 \\
x_2 &\leq 50 \\
-x_1 \qquad &\leq -2 \\
-x_2 &\leq 50
\end{aligned}
$$

Problem 6

Source: Himmelblau (1972) Problem 21, p. 422

Number of Variables: 3

Starting Point: $x_o = (100.0, 12.5, 3.0)$

$f(x_o) = 32.835$

Objective Function: $f(x) = \sum_{i=1}^{99} \left| \exp\left(-\frac{(u_i - x_2)^{x_3}}{x_1}\right) - 0.01i \right|^2$

$$u_i = 25 + (-50 \ln 0.01i)^{1/1.5}$$

Constraints: $0.1 < x_1 < 100.0$

$0.0 < x_2 < 25.6$

$0.0 < x_3 < 5.0$

Problem 7

Source: Hock and Schittkowski (1981), p. 51

Number of Variables: 3

Starting Point: $x_o = (-4.0, 1.0, 1.0)$

$f(x_o) = 13.00$

Objective Function: $f(x) = (x_1 + x_2)^2 + (x_2 + x_3)^2$

Constraints: $x_1 + 2x_2 + 3x_3 - 1 = 0$

Problem 8

Source: Hock and Schittkowski (1981), p. 58

Number of Variables: 3

Starting Point: $x_o = (0.5, 0.5, 0.5)$

$f(x_o) = 2.25$

Objective Function: $f(x) = 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3 - 8x_1$

$$-6x_2 - 4x_3 + 9$$

Constraints: $x_1 + x_2 + 2x_3 \leq +3$

$x_1, x_2, x_3 \geq 0$

Problem 9

Source: May (1979), p. 479

Number of Variables: 3

Starting Point: $x_o = (10.0, 10.0, 10.0)$

$f(x_o) = -1000.00$

Objective Function: $f(x) = -x_1 x_2 x_3$

Constraints: $x_1 + 2x_2 + 2x_3 \leq 72$

$0 \leq x_1 \leq 20$

$0 \leq x_2 \leq 11$

$0 \leq x_3 \leq 42$

Note: This problem, known as the "Post Office Problem" has the distinctive feature that the Hessian diagonal entries are all zero.

Problem 10

Source: Hock and Schittkowski (1981), p. 60

Number of Variables: 3

Starting Point: $x_o = (10.0, 10.0, 10.0)$

$f(x_o) = -1000.00$

Objective Function: $f(x) = -x_1 x_2 x_3$

Constraints: $x_1 + 2x_2 + 2x_3 < 72$

$-x_1 - 2x_2 - 2x_3 < 0$

$0 < x_1 < 42$

$0 < x_2 < 42$

$0 < x_3 < 42$

Problem 11

Source: Hock and Schittkowski (1981), p. 84

Number of Variables: 3

Starting Point: $x_o = (0.70, 0.20, 0.10)$

$f(x_o) = -25698.30$

Objective Function:

$$f(x) = -32.174(255 \ln((x_1 + x_2 + x_3 + 0.03)/(0.9x_1 + x_2 + x_3 + 0.03)) + 280 \ln((x_2 + x_3 + 0.03)/(0.07x_2 + x_3 + 0.03)) + 290 \ln((x_3 + 0.03)/(0.13x_3 + 0.03)))$$

Constraints: $0 < x_1 < 1$

$0 < x_2 < 1$

$0 < x_3 < 1$

Problem 12

Source: Himmelblau (1972), Problem 8, p. 403

Number of Variables: 4

Starting Point: $x_o = (-3.0, -1.0, -3.0, -1.0)$

$f(x_o) = 19.192$

Objective Function:

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1\,[(x2 - 1)^2 + (x4 - 1)^2] + 19.8(x2 - 1)(x4 - 1)$$

Constraints: $-10 < x_1 < 10$

$-10 < x_2 < 10$

$-10 < x_3 < 10$

$-10 < x_4 < 10$

Note: This problem was designed to have a nonoptimal stationary point at $f(x) \approx 8.0$ that can cause premature convergence.

Problem 13

Source: Hock and Schittkowski (1981), p. 67

Number of Variables: 4

Starting Point: $x_o = (0.0,0.0,0.0,0.0)$

$f(x_o) = 0.00$

Objective Function: $f(x) = x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4$

Constraints:

$$
\begin{aligned}
x_1 + 2x_2 &< 8 \\
4x_1 + x_2 &< 12 \\
3x_1 + 4x_2 &< 12 \\
2x_3 + x_4 &< 8 \\
x_3 + 2x_4 &< 8 \\
x_3 + x_4 &< 5 \\
x_1,\ x_2,\ x_3,\ x_4 &> 0
\end{aligned}
$$

Problem 14

Source: Hock and Schittkowski (1981), p. 64

Number of Variables: 4

Starting Point: $x_o = (2.0,2.0,2.0,2.0)$

$f(x_o) = -6.00$

Objective Function: $f(x) = 2 - x_1x_2x_2$

Constraints: $x_1 + 2x_2 + 2x_3 + x_4 = 0$

$$0 < x_1 < 1$$

$$0 < x_2 < 1$$

$$0 < x_3 < 1$$

$$0 < x_4 < 2$$

Problem 15

Source: Hock and Schittkowski (1981), p. 96

Number of Variables: 4

Starting Point: $x_o = (0.5, 0.5, 0.5, 0.5)$

$f(x_o) = -1.25$

Objective Function: $f(x) = x_1^2 + 0.5x_2^2 + x_3^2 + 0.5x_4^2 - x_1x_3$

$$+ x_3x_4 - x_1 - 3x_2 + x_3 - x_4$$

Constraints: $x_1 + 2x_2 + x_3 + x_4 < 5$

$$3x_1 + x_2 + 2x_3 - x_4 < 4$$

$$-x_2 - 4x_3 < -1.5$$

$$x_1, x_2, x_3, x_4 > 0$$

Problem 16

Source: Luenberger (1973), p. 264

Number of Variables: 4

Starting Point: $x_o = (2.0, 2.0, 1.0, 0.0)$

$f(x_o) = 5.00$

Objective Function: $f(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2x_1 - 3x_4$

Constraints: $2x_1 + x_2 + x_3 + 4x_4 = 7$

$$x_1 + x_2 + 2x_3 + x_4 = 6$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Problem 17

Source: Himmelblau (1972), Problem 10, p. 404

Number of Variables: 5

Starting Point: $x_o = (0.0,0.0,0.0,0.0,1.0)$

$f(x_o) = 20.00$

Objective Function: $f(x) = \sum_{j=1}^{5} e_j x_j + \sum_{i=1}^{5} \sum_{j=1}^{5} c_{ij} x_i x_j + \sum_{j=1}^{5} d_j x_j^3$

Constraints: $-\sum_{j=1}^{5} a_{ij} x_j \leq -b_i, \quad i=1,\ldots,10$

$$x_j \geq 0, \quad j=1,\ldots,5$$

The data for e_i, $i=1,\ldots,5$; c_{ij}, $i=1,\ldots,5$, $j=1,\ldots,5$; a_{ij}, $i=1,\ldots,10$, $j=1,\ldots,5$; and b_i, $i=1,\ldots,10$ is given below in Table A-1.

Note: This problem is known as the Shell Development Company Problem. Its objective function is non-convex.

Problem 18

Source: Hock and Schittkowski (1981), p. 68

Number of Variables: 5

Starting Point: $x_o = (1.0,2.0,2.0,2.0,2.0)$

$f(x_o) = 1.87$

Objective Function: $f(x) = 2 - 1/120 \; x_1 x_2 x_3 x_4 x_5$

Table A-1

j	1	2	3	4	5
c_j	-15	-27	-36	-18	-12
c_{1j}	30	-20	-10	32	-10
c_{2j}	-20	39	-6	-31	32
c_{3j}	-10	-6	10	-6	-10
c_{4j}	32	-31	-6	39	-20
c_{5j}	-10	32	-10	-20	30
d_j	4	8	10	6	2
a_{1j}	-16	2	0	1	0
a_{2j}	0	-2	0	4	2
a_{3j}	-3.5	0	2	0	0
a_{4j}	0	-2	0	-4	-1
a_{5j}	0	-9	-2	1	-2.8
a_{6j}	2	0	-4	0	0
a_{7j}	-1	-1	-1	-1	-1
a_{8j}	-1	-2	-3	-2	-1
a_{9j}	1	2	3	4	5
a_{10j}	1	1	1	1	1

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
-40	-2	-.25	-4	-4	-1	-40	-60	5	1

Constraints: $0 < x_1 < 1$

$0 < x_2 < 2$

$0 < x_3 < 3$

$0 < x_4 < 4$

$0 < x_5 < 5$

Problem 19

Source: Hock and Schittkowski (1981), p. 71

Number of Variables: 5

Starting Point: $x_o = (3.0,5.0,-3.0,2.0,-2.0)$

$f(x_o) = 84.00$

Objective Function: $f(x) = (x_1 - 1)^2 + (x_2 - x_3)^2 + (x_4 - x_5)^2$

Constraints: $x_1 + x_2 + x_3 + x_4 + x_5 = 5$

$x_3 + 2x_4 + 2x_5 = -3$

Problem 20

Source: Hock and Schittkowski (1981), p. 76

Number of Variables: 5

Starting Point: $x_o = (2.0,2.0,2.0,2.0,2.0)$

$f(x_o) = 6.00$

Objective Function:

$$f(x) = (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2$$

Constraints:

$$
\begin{aligned}
x_1 + 3x_2 \qquad\qquad &= 0 \\
x_3 + x_4 - 2x_5 &= 0 \\
x_2 \qquad\qquad - x_5 &= 0 \\
-10 < x_1 &< 10 \\
-10 < x_2 &< 10 \\
-10 < x_3 &< 10 \\
-10 < x_4 &< 10 \\
-10 < x_5 &< 10
\end{aligned}
$$

Problem 21

Source: Hock and Schittkowski (1981), p. 78

Number of Variables: 6

Starting Point: $x_0 = (1.0, 2.0, 0.0, 0.0, 0.0, 2.0)$

$f(x) = 6.0$

Objective Function: $f(x) = x_1 + 2x_2 + 4x_5 + \exp(x_1 x_4)$

Constraints:

$$
\begin{aligned}
x_1 + 2x_2 \qquad + 5x_5 \qquad &= 6 \\
x_1 + x_2 + x_3 \qquad\qquad &= 3 \\
x_4 + x_5 + x_6 &= 2 \\
x_1 \qquad + x_4 \qquad\qquad &= 1 \\
x_2 \qquad + x_5 \qquad &= 2 \\
x_3 \qquad + x_6 &= 2 \\
0 < x_1 &< 1 \\
0 < x_4 &< 1 \\
x_2, x_3, x_5, x_6 &> 0
\end{aligned}
$$

Problem 22

Source: Bracken and McCormick (1968), p. 90

Number of Variables: 8

Starting Point: $x_0 = (0.1, 0.2, 100, 125, 175, 11.2, 13.2, 15.8)$

$f(x_0) = 1,297.67$

Objective Function: $f(x) = - \sum_{i=1}^{235} \ell n \left| \frac{(a_i(x) + b_i(x) + c_i(x)}{(2\pi)^{\frac{1}{2}}} \right|$

where:

$$a_i(x) = \frac{x_1}{x_6} \exp \left| - \frac{(y_i - x_3)^2}{2x_6^2} \right|; \qquad i=1,\ldots,235$$

$$b_i(x) = \frac{x_2}{x_7} \exp \left| - \frac{(y_i - x_4)^2}{2x_7^2} \right|; \qquad i=1,\ldots,235$$

Constraints: $x_1 + x_2 < 1$

$0.001 < x_1 < 0.499$

$$c_i(x) = \frac{(1 - x_2 - x_1)}{x_8} \exp \left| - \frac{(y_i - x_5)^2}{2x_8^2} \right|; \quad i=1,\ldots,235$$

$0.001 < x_2 < 0.499$

$100 < x_3 < 180$

$130 < x_4 < 210$

$170 < x_5 < 240$

$5 < x_6 < 25$

$5 < x_7 < 25$

$5 < x_8 < 25$

The data for y_i is given in Table A-2

Table A-2

i	y_i	i	y_i
1	95	168-175	175
2	105	176-181	180
3-6	110	182-187	185
7-10	115	188-194	190
11-25	120	195-198	195
26-40	125	199-201	200
41-55	130	202-204	205
56-68	135	205-212	210
69-89	140	213	215
90-101	145	214-219	220
102-118	150	220-224	230
119-122	155	225	235
123-142	160	226-232	240
143-150	165	233	245
151-167	170	234-235	250

Problem 23

Source: Himmelblau (1972), Problem 17, p. 416

Number of Variables: 10

Starting Point: $x_0 = (9,9,9,9,9,9,9,9,9,9)$

$f(x_0) = -43.134$

Objective Function: $$f(x) = \sum_{i=1}^{10} \left[(\ell n(x_i - 2))^2 + (\ell n(10 - x_i))^2\right] - \left(\prod_{i=1}^{10} x_i\right)^{0.2}$$

Constraints: $2.001 \leq x_i \leq 9.999,\ i=1,\ldots,10$

Note: The objective function of this problem is undefined outside the feasible region.

Problem 24

Source: Himmelblau (1972), Problem 4, p. 395

Number of Variables: 10

Starting Point: $x_0 = (0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1)$

$f(x_0) = -20.961$

Objective Function: $$f(x) = \sum_{i=1}^{10} x_i \left[c_i + \ell n \frac{x_i}{\sum_{j=1}^{10} x_j}\right]$$

where c_i, $i=1,\ldots,10$ are given in Table A-3

Constraints:
$$x_1 + 2x_2 + 2x_3 + x_6 + x_{10} = 2$$
$$x_4 + 2x_5 + x_6 + x_7 = 1$$
$$x_3 + x_7 + x_8 + 2x_9 + x_{10} = 1$$
$$x_i \geq 0;\ i=1,\ldots,10$$

Table A-3

j	c_j	j	c_j
1	-6.089	6	-14.986
2	-17.164	7	-24.100
3	-34.054	8	-10.708
4	-5.914	9	-26.662
5	-24.721	10	022.179

Note: For $x_i = 0$, the objective function is not defined. Also, both the gradient and Hessian are unbounded in the vicinity of $x_i = 0$.

Problem 25

Source: Day (1979), p. 480

Number of Variables: 11

Starting Point:

$$x_0 = (0.33,0.65,0.65,0.65,0.65,0.65,0.65,0.65,0.65,0.65)$$

Objective Function: $f(x) = \sum_{i=1}^{11} \alpha_i x_i(\beta_i + \gamma_i x_i + \delta_i x_i^2)$

where the data for α_i, β_i, γ_i, δ_i, i=1,...,11 is given in Table A.4:

Constraints: $\sum_{j=1}^{11} a_{ij} x_j < 1$; i=1,2,...,10

$$0.01 < x_1 < 0.33$$

$$0.01 < x_i < 0.65;\ i=2,3,\ldots,11$$

where the a_{ij} are given in Table A-5.

Table A-4 (Power of 10 in parentheses)

j	α_j	β_j	γ_j	δ_j
1	6.5776(-1)	-2.6408(-2)	-1.2099(-2)	8.5941(-5)
2	3.6017(0)	-4.3808(-1)	6.5400(-3)	-5.4872(-5)
3	1.9244(-1)	-2.5327(-1)	-1.3651(-2)	1.0190(-4)
4	1.5122(0)	-6.3925(-1)	1.1832(-2)	-9.1418(-5)
5	2.0889(0)	-2.3527(-1)	-1.6156(-3)	8.1508(-6)
6	1.8326(0)	-1.7797(-1)	7.8416(-4)	-1.3105(-5)
7	3.0427(0)	-3.6994(-1)	5.2798(-3)	-3.9607(-5)
8	3.9075(0)	-5.1033(-1)	6.4643(-3)	-4.5730(-5)
9	1.2003(0)	-1.2865(-1)	-1.5012(-3)	5.3325(-6)
10	8.9714(-2)	-1.5468(-1)	-1.9908(-1)	-5.1898(-6)
11	5.3508(0)	-5.9023(-1)	1.5166(-2)	-1.2015(-4)

Note: This problem is known as the "Water Quality Problem." The data come from a version of the Willamette River System in Oregon. The variable x_i is the maximum allowable ratio of BOD (biochemical oxygen demand) of the effluent outflow to the BOD of the wastewater inflow for treatment plant i. The cost curves come from least squares fitting of cost data resulting in a nonconvex objective function. The final values of the variables x_j are equal to one minus the efficiency of each

Table A-5

	1	2	3	4	5	6	7	8	9	10	11
1	1.26700										
2	1.07600	7.7380	.215800	.53600							
3	0.51880	3.9450	.212400	.66010	.49670	.75550					
4	0.07548	0.6012	.044760	.14750	.13590	.21300	.13550				
5	0.07144	0.5690	.042370	.13960	.12860	.20160	.12820	2.1540			
6	0.06884	0.5409	.037030	.12040	.10650	.16600	.08642	1.4150	.46550		
7	0.04014	0.3197	.023810	.07842	.07226	.11330	.07205	1.2100	.71450		
8	0.02213	0.1763	.013130	.04323	.03984	.06246	.03972	0.6674	.39390	.08500	
9	0.01339	0.1067	.007943	.02617	.02411	.03780	.02404	0.4039	.23840	.05144	1.1120
10	0.03061	0.2416	.017030	.05561	.04992	.07799	.04389	0.7264	.31880	.03890	0.5605

treatment plant j. The objective function reflects the total cost of building or expanding the treatment plants to accomodate certain specified BOD loads.

Problem 26

Source: Kezouh (1983): Unpublished

Number of Variables: 14

Starting Point: $x_0 = (5,0,0,0,0,0,0,0,5,-1,0,0,-1,0)$

$f(x_0) = 43,074$

Objective Function:

$$f(x) = -[417x_1 + 96x_2 + 104x_3 + 110x_4 + 107x_5 + 206x_6 + 129x_7 + 48x_8 + 61x_9x_{10}x_{13} + 28x_9x_{10}x_{14} + 7x_9x_{10} + 68x_9x_{11}x_{13} + 23x_9x_{11}x_{14} + 13x_9x_{11} + 58x_9x_{12}x_{13} + 40x_9x_{12}x_{14} + 12x_9x_{12} + 53x_9x_{13} + 38x_9x_{14}] + \exp(x_1 + x_2)\,[\exp(x_6 + x_9x_{10}x_{13}) + \exp(x_7 + x_9x_{10}x_{14}) + \exp(x_8 + x_9x_{10})] + \exp(x_1 + x_3)[\exp(x_6 + x_9x_{11}x_{13}) + \exp(x_7 + x_9x_{11}x_{14}) + \exp(x_8 + x_9x_{11})] + \exp(x_1 + x_4)[\exp(x_6 + x_9x_{12}x_{13}) + \exp(x_7 + x_9x_{12}x_{14}) + \exp(x_8 + x_9x_{12})] + \exp(x_1 + x_5)[\exp(x_6 + x_9x_{13}) + \exp(x_7 + x_9x_{14}) + \exp(x_8 + x_9)]$$

Constraints:

$$x_2 + x_3 + x_4 + x_5 = 0$$
$$x_6 + x_7 + x_8 = 0$$
$$x_{10} + x_{11} + x_{12} = -1$$
$$x_3 + x_{14} = -1$$
$$x_{10} - x_{11} \leq 0$$
$$x_{11} - x_{12} \leq 0$$
$$x_{12} \leq 1$$
$$x_{13} - x_{14} \leq 0$$

$x_{14} \leq 1$

$-20 \leq x_i \leq 20; \; i=2,3,\ldots,8$

$-3 \leq x_i \leq 1; \; i=10,11,.,13,14$

$-10 \leq x_9 \leq 10$

$0 \leq x_1 \leq 20$

Problem 27

Source: Hock and Schittkowski (1981), p. 126

Number of Variables: 15

Starting Point: $x_0 = (20,55,15,20,60,20,20,60,20,20,60,20,20,60,20)$

$f(x_o) = 664.82$

Objective Function:
$$f(x) = \sum_{k=0}^{4} [2.30\, x_{3k+1} + 0.0001\, x_{3k+1}^2 + 1.7\, x_{3k+2} + 0.0001\, x_{3k+2}^2 + 2.2\, x_{3k+3} + 0.00015\, x_{3k+3}^2]$$

Constraints:

$0 \leq x_4 - x_1 \leq 6$

$0 \leq x_5 - x_2 \leq 6$

$0 \leq x_6 - x_3 \leq 6$

$0 \leq x_7 - x_4 \leq 6$

$0 \leq x_8 - x_5 \leq 6$

$0 \leq x_9 - x_6 \leq 6$

$0 \leq x_{10} - x_7 \leq 6$

$0 \leq x_{11} - x_8 \leq 6$

$0 \leq x_{12} - x_{19} \leq 6$

$0 \leq x_{13} - x_{10} \leq 6$

$0 \leq x_{14} - x_{11} \leq 6$

$0 < x_{15} - x_{12} < 6$

$-x_1 - x_2 - x_3 < -60$

$-x_4 - x_5 - x_6 < -50$

$-x_7 - x_8 - x_9 < -70$

$-x_{10} - x_{11} - x_{12} < -85$

$-x_{13} - x_{14} - x_{15} < -100$

$8 < x_1 < 21$

$43 < x_2 < 57$

$3 < x_3 < 16$

$0 < x_4 < 90$

$0 < x_5 < 120$

$0 < x_6 < 60$

$0 < x_7 < 90$

$0 < x_8 < 120$

$0 < x_9 < 60$

$0 < x_{10} < 90$

$0 < x_{11} < 120$

$0 < x_{12} < 60$

$0 < x_{13} < 90$

$0 < x_{14} < 120$

$0 < x_{15} < 60$

Note: The objective function is a quadratic, strictly convex function. This is a problem related to electric power scheduling as discussed in Biggs (1976). It is a representation of the problem of scheduling three generators to meet the demand for power over a period of time. The variables denote the output from the different generators at specific points in time.

Problem 28

Source: Himmelblau (1972), problem 19, p.417

Number of Variables: 16

Starting Point:

$x_o = (0,0.885,0,0.289,1.308,0.828,2.270,0,1.691,0.261,0,0,0,0,0.276,0)$

Objective Function: $f(x) = \sum_{i=1}^{16} \sum_{j=1}^{16} a_{ij}(x_i^2 + x_i + 1)(x_j^2 + x_j + 1)$

Constraints: $\sum_{j=1}^{16} b_{ij}x_j = c_i; \; i=1,2,\ldots,8$

$0 < x_i < 5; \quad i=1,\ldots,16$

The data for a_{ij}, b_{ij}, and c_i is given in Table A-6.

Problem 29

Source: Himmelblau (1972), problem 6, p. 397

Number of Variables: 45

Starting Point: x_o =

[.6540	0.0000	3.7050	46.7500	0.0000	0.0000	0.0000	0.0000
0.0000	.8005	.0881	.0483	0.0000	0.0000	.2615	.0204
.0155	0.0000	0.0000	0.0000	.0022	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	.0211	.0023	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	.0091	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000]			

Objective Function: $f(x) = \sum_{k=1}^{n_k} \left[\sum_{j=1}^{n_k} x_{jk} \left(c_{jk} + \ln \frac{x_{jk}}{\sum_{j=1}^{n_k} x_{jk}} \right) \right]$

Constraints: $f(x) = \sum_{k=1}^{7} [\sum_{j=1}^{n_k} E_{ijk} x_{jk}) = b_i; i=1,...,16$

$-x_{jk} < 0 \; [(j=1,...,n_k), \; k=1,...,7]$

where the data for n_k, c_{jk}, and b_i is given in Table A-7, and the data for E_{ijk} is given in Table A-8.

Note: This problem is known as the "Chemical Equilibrium Problem"

Table A-7

i	b_i	j	k	c_{jk}	j	k	c_{jk}
1	0.6529581	1	1	0.0	6	3	0.0
2	0.281941	2	1	-7.69	7	3	2.2435
3	3.705233	3	1	-11.52	8	3	0.0
4	47.00022	4	1	-36.60	9	3	-39.39
5	47.02972	1	2	-10.94	10	3	-21.49
6	0.08005	2	2	0.0	11	3	-32.84
7	0.08813	3	2	0.0	12	3	6.12
8	0.04829	4	2	0.0	13	3	0.0
9	0.0155	5	2	0.0	14	3	0.0
10	0.0211275	6	2	0.0	15	3	-1.9028
11	0.0022725	7	2	0.0	16	3	-2.8889
12	0.0	8	2	2.5966	17	3	-3.3622
13	0.0	9	2	-39.39	18	3	-7.4854
14	0.0	10	2	-21.35	1	4	-15.639
15	0.0	11	2	-32.84	2	4	0.0
16	0.0	12	2	6.26	3	4	21.81
		13	2	0.0	1	5	-16.79
		1	3	10.45	2	5	0.0
		2	3	0.0	3	5	18.9779
		3	3	-0.50	1	6	0.0
		4	3	0.0	2	6	11.959
		5	3	0.0	1	7	0.0
					2	7	12.899

Table A-8

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x_{jk}																
x_{11}	1															
x_{21}		1														
x_{31}			1													
x_{41}				1	1											
x_{12}	1															
x_{22}		1														
x_{32}			1													
x_{42}				1								1				
x_{52}					1							-1				
x_{62}						1						-1				
x_{72}							1					1				
x_{82}								1				1				
x_{92}				1	1											
$x_{10.2}$	1			1								-1				
$x_{11.2}$	1		1	1												
$x_{12.2}$	1		-1	1								-2				
$x_{13.3}$								1				-1				
x_{13}	1															
x_{23}		1														
x_{33}			1													
x_{43}				1												
x_{53}					1											
x_{63}						1										
x_{73}							1									
x_{83}								1								
x_{93}				1	1											
$x_{10.3}$	1			1												
$x_{11.3}$	1		1	1												
$x_{12.3}$	1		-1	1												
$x_{13.3}$									1							
$x_{14.3}$											1		-4			
$x_{15.3}$	1										1		-3	-1		
$x_{16.3}$	2										1		-2	-2		
$x_{17.3}$	3										1		-1	-3		
$x_{18.3}$	4										1			-4		
x_{14}				1									1			
x_{24}													1			
x_{34}				-1									1		-4	
x_{15}				1										1		
x_{25}														1		-4
x_{35}				-1										1		
x_{16}															1	
x_{26}		1		-1											1	1
x_{17}																1
x_{27}		1		-1												

Problem 30

Source: Himmelblau (1972), problem 23, p. 423

Number of Variables: 100

Starting Point: x_o is given in Table A-9

$f(x_o) = -181.80$

Objective Function: $f(x) = \sum_{j=1}^{20} u_j \left[\sum_{i=1}^{5} a_{ij}^{x_{ij}} - 1 \right]$

Constraints:

$$-\left(\sum_{i=1}^{5} x_{ij} \right) \quad -b_j; \; j=1,6,10,14,15,16,20$$

$$\left(\sum_{j=1}^{20} x_{ij} \right) \quad c_i; \; i=1,\ldots,5$$

$$-x_{ij} \quad 0; \; i=1,\ldots,100$$

The data for a_{ij}, $_j$, b_j, and c_i, is given in Table A-10

Note: This problem is known as the "Weapons Assignment Problem."

Table A-9

i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	x_1	x_6								35				50	70	35			x_{91}	10
2						95				5										
3	30					5														
4																				
5	x_5	x_{10}																x_{90}	x_{95}	x_{100}

Table A-10

a_{ij} probability that weapon i will not damage target j

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	c_l: no of weapons available
1	1	.95	1	1	1	.85	.90	.85	.80	1	1	1	1	1	1	1	1	.95	1	1	200
2	.84	.83	.85	.84	.85	.81	.81	.82	.80	.86	1	.98	1	.88	.87	.88	.85	.84	.85	.85	100
3	.96	.95	.96	.96	.96	.90	.92	.91	.92	.95	.99	.98	.99	.98	.97	.98	.95	.92	.93	.92	300
4	1	1	1	1	1	1	1	1	1	.96	.91	.92	.91	.92	.98	.93	1	1	1	1	150
5	.92	.94	.92	.95	.95	.98	.98	1	1	.90	.95	.96	.91	.98	.99	.99	1	1	1	1	250

b_j: minimum no. of weapons to be assigned to target j

30					100				40				50	70	35				10

u_j: military value of target j

60	50	50	75	40	60	35	30	25	150	30	45	125	200	200	130	100	100	100	150

BIBLIOGRAPHY

Abadie, J., (1970), "Applications of the GRG Algorithm to Optimal Control Problems," in Integer and Nonlinear Programming, J. Abadie (Editor), Elsevier, New York.

Abadie, J., (1977), "The GRG Method for Nonlinear Programming," in Design and Implementation of Optimization Software, H. J. Greenberg (Editor), pp. 335-362.

Allwright, J. C., (1972), Paper 12, in D. J. Bell, Recent Mathematical Developments in Control, Academic Press.

Armijo, L., (1966), "Minimization of Functions Having Continuous Partial Derivatives," Pacific Journal of Mathematics, Vol. 16, pp. 1-3.

Avriel, M., and R. S. Dembo, (1979), (eds.), Mathematical Programming Study 11: Engineering Optimization, North Holland.

Bartels, R. H., Golub, G. H., and M. A. Saunders, (1970), "Numerical Techniques in Mathematical Programming," in Nonlinear Programming, O. L. Mangasarian and K. Ritter (eds.), Academic Press.

Bazaraa, M. S., and J. J. Goode, (1981), "An Algorithm for Linearly Constrained Nonlinear Programming Problems," Journal Math. Analysis and Applications, 81, pp. 8-19.

Bazaraa, M. S., and C. M. Shetty, (1979), Nonlinear Programming Theory and Algorithms, Wiley, New York.

Baxter, E. B., Bolton, J. F., Gagnon, C. R., and R. H. Hicks, (1975), "Plant Scheduling Program Optimizer," Report BCS G-0648, Boeing Computer Services, March.

Beale, E. M. L., (1967), "Numerical Methods," in Nonlinear Programming, J. Abadie, (ed.), North-Holland, Amsterdam.

Beale, E. M. L., (1969), "Nonlinear Optimization by Simplex-Like Methods," in Optimization, R. Fletcher (ed.), Academic Press, New York.

Beale, E. M. L., (1972), "A Derivation of Conjugate Gradients," in Numerical Methods for Nonlinear Optimization, F. A. Lootsma (ed.), Academic Press.

Beale, E. M. L., (1974), "A Conjugate Gradient Method of Approximation Programming," in Optimization Methods, R. W. Cottle and J. Krarup (eds.), English Universities Press, London.

Bertsekas, D. P., (1978), "Algorithms for Nonlinear Multicommodity Network Flow Problems," International Symposium on Systems Optimization and Analysis, Rocquencourt, December 11-13, A. Bensoussan and J. L. Lions, (eds.), pp. 210-224.

Bertsekas, D. P., (1979), "Algorithms for Nonlinear Multicommodity Network Flow Problems," in International Symposium on Systems Optimization and Analysis, A Bensoussan and J. Lions (eds.), Springer Verlag, pp. 210-224.

Best, M. J., (1975a), "A Feasible Conjugate Direction Method to Solve Linearly Constrained Minimization Problems," J.O.T.A., Vol. 16, pp. 25-38.

Best, M. J., (1975b), "A Method to Accelerate the Rate of Convergence of a Class of Optimization Algorithms," Mathematical Programming, Vol. 8, pp. 139-160.

Best, M. J., and K. Ritter, (1975), "An Accelerated Conjugate Direction Method to Solve Linearly Constrained Minimization Problems," Journal of Computer and System Sciences, Vol. 11, pp. 295-322.

Best, M. J., and K. Ritter, (1976), "A Class of Accelerated Conjugate Direction Methods for Linearly Constrained Minimization Problems," Math. of Computation, Vol. 30, No. 135, pp. 478-504.

Biggs, M. C., (1976a), "Constrained Minimization Using Recursive Quadratic Programming: Some Alternative Subproblem Formulations," in Towards Global Optimization, L. C. W. Dixon and G. P. Szego (eds.), North-Holland, Amsterdam.

Biggs, M. C., (1976b), "An Approach to Optimal Scheduling of an Electric Power System," in Optimization in Action, L. C. W. Dixon (ed.), Academic Press.

Boberg, T. C., Woods, E. G., McDonald, W. J., Stone, H. L., and O. Abib, (1974), "Application of Inverse Simulation to a Complex Multireservoir System," Journal of Petroleum Technology, Vol. 26, No. 7, July, pp. 801-808.

Botsaris, C. A., (1976), "A Newton-Type Curvilinear Search Method for Optimization," Journal Math. Analysis and Applications, 54, pp. 217-229.

Botsaris, C. A., (1978), "Differential Gradient Methods," Journal Math. Analysis and Applications, 63, pp. 177-198.

Botsaris, C. A., (1979), "An Efficient Curvilinear Method for the Minimization of a Nonlinear Function Subject to Linear Inequality Constraints," Journal Math. Analysis and Applications, 71, pp. 482-515.

Bouzaher, A., (1978), A Mathematical Model for Regional Crop Allocation, unpublished Masters' Thesis, Georgia Tech, Atlanta.

Bracken, J., and G. P. McCormick, (1968), Selected Applications of Nonlinear Programming, Wiley, New York.

Broyden, C. G., (1967), "Quasi-Newton Methods and Their Application to Function Minimization," Math. of Computation, Vol. 21, pp. 368-381.

Broyden, C. G., (1970), "The Convergence of a Class of Double-Rank Minimization Algorithms, 2. The New Algorithm," J. Inst. Maths. Applcs., Vol. 6, pp. 222-231.

Buckley, A. G., (1975), "An Alternative Implementation of Goldfarb's Minimization Algorithm," Mathematical Programming, Vol. 8, pp. 207-231.

Cheifetz, A., (1974), "An Oil Field Development Model Using Approximation Programming," 4th ORSA/TIMS National Conference, Boston, April.

Cohen, A. I., (1972), "Rate of Convergence of Several Conjugate Gradient Algorithms," SIAM Jr. on Num. Analysis, Vol. 9, pp. 248-259.

Collins, M., Cooper, L., Helgason, R., Kennington, J., and L. Leblanc, (1978), Management Science, Vol. 24, No. 1, March, pp. 747-760.

Colville, A. R., (1970), "A Comparative Study of Nonlinear Programming Codes," Proceedings of the Princeton Symposium on Mathematical Programming, H. W. Kuhn (ed.), Princeton University Press, Princeton, N.J.

Daniel, J. W., (1973), "Stability of the Solution of Definite Quadratic Programs," Mathematical Programming, Vol. 5, pp. 41-53.

Davidon, W., (1959), "Variable Metric Methods for Minimization," AEC Res. and Development Report, ANL-5900, Argone National Laboratory, Lemont, Illinois.

DeNeufrille, R. P., Schaake, J., and J. H. Stafford, (1971), "Systems Analysis of Water Distribution Networks," Proceedings of the ASCE, 97, (SAC), pp. 825-842.

Dennis, J. E., and J. J. Moore, (1977), "Quasi-Newton Methods: Motivation and Theory," SIAM Review, Vol. 19, pp. 46-89.

Dixon, L. C. W., (1972), "Quasi-Newton Algorithms Generate Identical Points," Mathematical Programming, Vol. 2, pp. 383-387.

Dixon, L. C. W., (1975a), "Quadratic Termination and Second Order Convergence," in Towards Global Optimization, L. C. W. Dixon and G. P. Szego (eds.), North-Holland, Amsterdam.

Dixon, L. C. W., (1975b), "Conjugate Gradient Algorithms: Quadratic Termination Without Line Searches," J. Inst. Maths. Applcs., Vol. 15, pp. 9-18.

Dixon, L. C. W., (1980), "On Conjugate Gradient Algorithms," in Nonlinear Optimization: Theory and Algorithms, L. C. W. Dixon, E. Spedicato, and G. P. Szego, (eds.), Birkhauser.

Donachie, R., (1973), "Digital Program for Water Network Analysis," Jo. Hydraulics Div., Proc. Amer. Soc. Civil Engineers, Vol. 180, pp. 393, 403.

Duffin, R. J., Peterson, E. L., and L. Zener, (1967), Geometric Programming: Theory and Applications, Wiley, New York.

Durrer, E. J., and G. E. Slater, (1977), "Optimization of Petroleum and Natural Gas Production: A Survey," Management Science, Vol. 24, No. 1, pp. 35-43.

Dux, C., Krohm, G., and J. L. Vankniken, (1977), "Modeling Long-Term Coal Production with the Argonne Coal Market Model," Report No. ANL/AA-13, Argonne National Laboratory, August.

Eaves, B. C., (1971), "The Linear Complementary Problem," Management Science, Vol. 17, pp. 612-634.

Fiacco, A. V., and G. P. McCormick, (1968), Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley, New York.

Fisher, J., (1981), "Proof of Convergence for a Modification of Goldfarb's Algorithm," Mathematics of Operations Research, Vol. 6, No. 2, p. 233.

Fletcher, R., (1970), "A new Approach to Variable Metric Algorithms," Computer Journal, Vol. 13, pp. 317-322.

Fletcher, R., (1972a), "An Algorithm for Solving Linearly Constrained Optimization Problems," Math. Programming, Vol. 2, pp. 133-165.

Fletcher, R., (1972b), "Conjugate Direction Methods," in Numerical Methods for Unconstrained Optimization, W. Murray, (ed.), Academic Press.

Fletcher, R., and M. J. D. Powell, (1963), "A Rapidly Convergent Descent Method for Minimization," Computer Journal, Vol. 6, pp. 163-168.

Fletcher, R., and C. M. Reeves, (1964), "Function Minimization by Conjugate Gradients," Computer Journal, Vol. 7, pp. 149-154.

Frank, M., and R. Wolfe, (1956), "An Algorithm for Quadratic Programming," NRLQ, Vol. 3, pp. 95-110.

Gallager, R., (1974), "A Minimum Delay Routing Algorithm Using Distributed Computation," IEEE Transactions on Communication, Vol. COM-25, pp. 73-85.

Ganzhela, I. F., (1970), "An Algorithm of Descent with Constraints," USSR Comp. Math. and Math. Phys., Vol. 10, pp. 191-206.

Geoffin, J. L., and J. M. Rousseau, (1982), (eds.), Mathematical Programming Study 20, Applications, North-Holland, Amsterdam.

Gill, P. E., Golub, G. H., Murray, W., and M. A. Saunders, (1974), "Methods of Modifying Matrix Factorizations," Math. of Comp., Vol. 28, pp. 505-535.

Gill, P. E., and W. Murray, (1974a), "Newton-Type Methods for Linearly Constrained Optimization," in Numerical Methods for Constrained Optimization, P. E. Gill and W. Murray (eds.), Academic Press.

Gill, P. E., and W. Murray, (1974b), "Quasi-Newton Methods for Linearly Constrained Optimization," in Numerical Methods for Constrained Optimization, P. E. Gill and W. Murray (eds.), Academic Press.

Glassey, C. R., (1978), "Price Sensitive Consumer Demands in Energy Modeling," Management Science, 24, pp. 877-886.

Goldfarb, D., (1969), "Extension of Davidon's Variable Metric Algorithm to Maximization Under Linear Inequality and Equality Constraints," SIAM J. Appl. Math., Vol. 17, pp. 739-764.

Goldfarb, D., (1970), "A Family of Variable Metric Methods Derived by Variational Means," Math. of Comp., Vol. 24, pp. 23-26.

Goldfarb, D., (1976), "Matrix Factorizations in Optimization of Nonlinear Functions Subject to Linear Constraints," Math. Programming, Vol. 10, pp. 1-31.

Goldstein, A. A., (1964), "Convex Programming in Hilbert Space," Bull. Amer. Math. Soc., Vol. 70, pp. 709-710.

Gruver, W. A., and N. H. Engersback, (1976), "A Variable Metric Algorithm for Constrained Minimization Based on Augmented Lagrangian," Int. J. for Numer. Math. in Eng., 10, pp. 1047-1056.

Hamilton, J. F., and K. M. Ragsdell, (1977), "On the Performance of a Combined Generalized Reduced Gradient and Davidon-Fletcher-Powell Algorithm," Engineering Optimization, 2, pp. 269-278.

Hanscom, M. A., Lafond, L., Lasdon, L. S., and Pronovost, G., (1976), "Modeling and Resolution of the Deterministic Mid-Term Energy Production Problem for the Hydro-Quebec System," Report IREQ - 1453, L'Institute de Recherches de l'Hydro-Quebec, Varennes, P. Q.

Happ, H. H., (1977), "Optimal Power Dispatch," IEEE Transactions, PAS 93, 820-830.

Heady, E., and U. K. Srivastava, (1975), Spatial Sector Programming Models in Agriculture, Iowa State University Press, Iowa City.

Heltne, D. R., and Littschwager, (1975), User's Guide for GRG73 and Technical Appendices to GRG73, December.

Hestenes, M. R., and E. Steifel, (1952), "Methods of Conjugate Gradients for Solving Linear Systems," Journal of Research of the National Bureau of Standards, Vol. 29, pp. 409-439.

Himmelblau, D. M., (1972), Applied Nonlinear Programming, McGraw Hill, New York.

Hock, W., and K. Schiffkowski, (1981), Test Examples for Nonlinear Programming Codes, Springer Verlag, New York.

Huang, H. Y., (1970), "A Unified Approach to Quadratically Convergent Algorithms for Function Minimization, JOTA, Vol. 5, pp. 405-423.

Jeppson, R. W., (1975), Analysis of Flow in Pipe Networks, Ann Arbor Science, Ann Arbor, Michigan.

Juseret, R., (1978), "Long Term Optimization of Electrical System Generation by Convex Programming," Survey of Mathematical Programming. Proceedings of the 9th International Mathematical Programming Symposium, Budapest, August 23-27, 1976, Vol. 3, A. Prekopa (ed.) pp. 229-242.

Kawamura, K., and R. A. Volz, (1973), "On the Rate of Convergence of the Conjugate Gradient Reset Method with Inaccurate Linear Minimizations," IEEE Transactions on Automatic Control, Vol. AC-18, pp. 360-366.

Kezouh, A., (1983), Private Communication.

Klessig, R., (1974), "A General Theory of Convergence for Constrained Optimization Algorithms that Use Antizigzagging Provisions," SIAM J. Control, Vol. 2, No. 4, pp. 598-607.

Kuhn, H. W., and A. W. Tucker, (1951), "Nonlinear Programming," Proceedings 2nd Berkely Symposium on Mathematical Statistics and Probability, J. Neyman (ed.), University of California Press, Berkeley, CA.

Lasdon, L. S., and A. D. Warren, (1978), "Generalized Reduced Gradient Software for Linearly and Nonlinearly Constrained Problems," in Design and Implementation of Optimization Software, H. J. Greenberg (ed.), pp. 363-396.

Lasdon, L. S., and A. D. Warren, (1980), "Survey of Nonlinear Programming Applications," Operations Research, Vol. 28, No. 5, Sept-Oct., pp. 1029-1073.

Lemke, C. E., (1961), "The Constrained Gradient Method for Linear Programming," SIAM Journal, Vol. 9, pp. 1-17.

Lemke, C. E., (1965), "Bimatrix Equilibrium Points and Mathematical Programming," Management Science, Vol. 11, pp. 681-689.

Lemke, C. E., (1968), "On Complementary Pivot Theory," in Mathematics of Decision Sciences, Part I, G. B. Dantzig and A. F. Vernott (eds.), American Mathematical Society.

Lenard, M. L. (1979), "A Computational Study of Active Set Strategies in Nonlinear Programming with Linear Constraints," Mathematical Programming, Vol. 16, pp. 81-97.

Levitin, E. S., and B. T. Polyak, (1966), "Constrained Minimization Methods," USSR Comp. Math. and Math. Phy., Vol. 6, pp. 1-50.

Luenberger, D. C., (1973), Introduction to Linear and Nonlinear Programming, Addison-Wesley.

Manne, A., Richels, R., and J. Weyant, (1979), "Energy Policy Modeling: A Survey," Operations Research, 27, pp. 1-36.

May, J. H., (1976), "Users' Manual QRMNEW," Working Paper 168, Graduate School of Business, University of Pittsburg, PA, August.

May, J. H., (1979), "Solving Nonlinear Programs Without Using Analytic Derivatives," Operations Research, 27, No. 3, May-June, pp. 457-484.

McCormick, G. P., (1970a), "The Variable Reduction Method for Nonlinear Programming," Management Science, Vol. 17, pp. 146-160.

McCormick, G. P., (1970b), "A Second Order Method for Lineraly Constrained Nonlinear Programming Problems," in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. R. Ritter (eds.), Academic Press.

McCormick, G. P., and K. Ritter, (1972a), "Methods of Conjugate Directions Versus Quasi-Newton Methods," Mathematical Programming, Vol. 3, pp. 101-116.

McCormick, G. P. and K. Ritter, (1972b), "Projection Method for Unconstrained Optimization," JOTA, Vol. 10, pp. 57-66.

Mukai, H., (1978), "A Scheme for Determining Step Sizes for Unconstrained Optimization Methods," IEEE Transactions on Automatic Control, Vol. AC-23, pp. 987-995.

Mukai, H., (1979), "Readily Implementable Conjugate Gradient Methods," Mathematical Programming, Vol. 17, pp. 298-319.

Murtagh, B. A., and R. W. H. Sargent, (1969), "A Constrained Minimization Method with Quadratic Convergence," in Optimization, R. Fletcher (ed.), Academic Press.

Murtagh, B. A., and M. A. Saunders, (1978), "Large-Scale Linearly Constrained Optimization," Mathematical Programming, Vol. 14, pp. 41-72.

Newell, J. S., and D. M. Himmelblau, (1975), "A New Method for Nonlinearly Constrained Optimization," AICHE Journal, Vol. 21, No. 3, pp. 479-486.

Nazareth, L., (1977), "A Conjugate Direction Algorithm without Line Searches," JOTA, Vol. 23, No. 3, pp. 373-387.

Ortega, J. M. and W. C. Rheinboldt, (1970), Iterative Solution of Nonlinear Equations in Several Variables, Academic Press.

Panne, C., (1974), "A Complementary Variant of Lemke's Method for the Linear Complementary Problem," Mathematical Programming, Vol. 7, pp. 283-310.

Pironneau, O., and E. Polak, (1973), "Rate of Convergence of a Class of Methods of Feasible Directions," SIAM J. Num. Anal., Vol. 10, No. 1, pp. 161-174.

Polak, E., and G. Ribiere, (1969), "Note sur la Convergence des Methodes Conjuguees," Rev. Fr. Rech. Oper., Vol. 16, pp. 35-43.

Polak, E., (1971), Computational Methods in Optimization, Academic Press.

Powell, M. J. D., (1966), "Minimization of Functions of Several Variables," in Numerical Analysis: An Introduction, J. Walsh (ed.), Academic Press.

Powell, M. J. D., (1971), "On the Convergence of the Variable Metric Algorithm," J. Inst. Maths. Applcs., Vol. 7, pp. 21-36.

Powell, M. J. D., (1972), "Quadratic Termination Properties of a Class of Double-Rank Minimization Algorithms," J. Inst. Maths. Applcs., Vol. 10, pp. 333-342.

Powell, M. J. D., (1974a), "Unconstrained Minimization and Extensions for Constraints," in Mathematical Programming in Theory and Practice, P. L. Hammer and G. Zoutendijk (eds.), pp. 31-79.

Powell, M. J. D., (1974b), "Introduction to Constrained Optimization," in Numerical Methods for Constrained Optimization, P. E. Gill and W. Murray (eds.), Academic Press.

Powell, M. J. D., (1977), "Restart Procedures for the Conjugate Gradient Method," Mathematical Programming, Vol. 12, pp. 241-254.

Ratliff, D. H., (1978), "Network Models for Production Scheduling Problems with Convex Cost and Batch Processing," AIIE Transactions, Vol. 10, No. 1, pp. 104-108.

Resta, G., (1978), "Convex Programming via Functional Linear Programming," in Towards Global Optimization, 2, L. C. W. Dixon and G. P. Szego (eds.), North-Holland, Amsterdam, pp. 343-362.

Ritter, K., (1971), "A Method of Conjugate Directions for Unconstrained Minimization," Methods of Operations Research, Vol. 13, pp. 293-320.

Ritter, K., (1975), "A Method of Conjugate Directions for Linearly Constrained Nonlinear Programming Problems," SIAM J. Num. Anal., Vol. 12, pp. 273-303.

Robinson, S. M., (1972), "A Quadratically Convergent Algorithm for General Programming Problems," Mathematical Programming, 3, pp. 145-156.

Rosen, J. B., (1960), "The Gradient Projection Method for Nonlinear Programming, Part I: Linear Constraints," SIAM Journal, Vol. 8, pp. 181-217.

Rosen, J. B., (1977), "Two-Phase Algorithm for Nonlinear Constraint Problems," Technical Report 77-8, Computer Science Dept., University of Minnesota, Minn., August.

Rosen, J. B., and J. Kreuser, (1972), "A Gradient Projection Algorithm for Nonlinear Constraints," Numerical Methods for Nonlinear Optimization, F. A. Lootsma (ed.), Academic Press.

Rosen, J. B., and S. Wagner, (1975), "The GPM Nonlinear Programming Subroutine Package; Description and User Instructions," Technical Report 75-9, Computer Science Dept., University of Minnesota, Minneapolis, May.

Rustem, B., (1981), Projection Methods in Constrained Optimization and Applications to Optimal Policy Decisions, Springer-Verlag, New York.

Sandgren, E., and K. M. Ragsdell, (1982), "On Some Experiments Which Delimit the Utility of Nonlinear Programming Methods for Engineering Design," in Mathematical Programming Study 16, pp. 118-136.

Sargent, R. W., (1974), "Reduced Gradient and Projection Methods for Nonlinear Programming," in Numerical Methods for Constrained Optimization, P. E. Gill and W. Murray (eds.), Academic Press.

Sargent, R. W., and B. A. Murtagh, (1973), "Projection Methods for Nonlinear Programming," Mathematical Programming, Vol. 4, pp. 245-268.

Schittkowski, K., (1980), Nonlinear Programming Codes, Springer-Verlag, Berlin Heidelberg, New York.

Schultz, H. K., (1973), " A Kuhn-Tucker Algorithm," SIAM Jr. Control, Vol. 11, No. 3, August, pp. 438-445.

Shanno, D. F., (1970), "Conditioning of Quasi-Newton Methods for Function Minimization," Mathematical Programming, Vol. 24, pp. 647-654.

Shanno, D. F., and R. E. Marsten, (1979), "Conjugate Gradient Methods for Linearly Constrained Nonlinear Programming," MIS Technical Report 79-13, The University of Arizona.

Topkis, D. M., and A. Veinolt, (1967), "On the Convergence of Feasible Direction Algorithms for Nonlinear Programming," SIAM J. Control, Vol. 5, pp. 268-279.

Van Der Hoek, G., (1980), Reduction Methods in Nonlinear Programming, Mathematical Center Tract 126, Amsterdam.

Warren, A. D., and L. S. Lasdon, (1979), "The Status of Nonlinear Programming Software," Operations Research, 27, pp. 431-456.

Wolfe, P., (1967), "Methods for Nonlinear Constraints," in Nonlinear Programming, J. Abadie (ed.), North-Holland.

Wolfe, P., (1969), "Convergence Conditions for Ascent Methods," SIAM Review, Vol. 11, pp. 226-235.

Wolfe, P., (1972), "On the Convergence of Gradient Methods Under Constraints," IBM J. Res. Development, July, pp. 407-411.

Yeh, W., (1974), "Optimal Identification of Parameters in an Inhomogeneous Medium with Quadratic Programming," SPE 5021, 49th Annual Fall Meeting, Houston, October.

Zangwill, W. I., (1969), Nonlinear Programming, Prentice Hall.

Zoutendijk, G., (1960), Methods of Feasible Directions, Elsevier Publishing Company, London, New York.

Zoutendijk, G., (1970), "Some Algorithms Based on the Principle of Feasible Directions," in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter (eds.), pp. 93-121.

Zoutendijk, G., (1974), "On Linearly Constrained Nonlinear Programming and Extensions," in Mathematical Programming in Theory and Practice, P. I. Hammer and G. Zoutendijk (eds.), North-Holland, pp. 81-119.

Zoutendijk, G., (1976), Mathematical Programming Methods, North-Holland.

VITA

Abdelaziz Bouzaher was born in Khenchela, Algeria on May 17, 1952. He attended the University of Algiers during the period September 1971 to June 1975, while working with different government companies. He earned a Bachelor Degree in Economics and travelled to the United States for graduate studies.

In June 1978 he obtained a Master of Science in Operations Research from the Georgia Institute of Technology. His Master's thesis: "A Mathematical Model for Regional Crop Allocation" won the Outstanding Masters' Thesis Award and the Sigma Xi Research Award. He joined the Ph.D. program in the same school in the Winter of 1979. He conducted research in nonlinear optimization and taught several undergraduate courses. In 1983 he earned the Ph.D. Degree and accepted a position of Lecturer in the School of Industrial and Systems Engineering, Georgia Institute of Technology.