Nonlinear Programming Using an Expanded Lagrangian Function: A Water Resources Management Case Study

by

Seyed Jafar Sadjadi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 1997

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Optimal planning and operation of large hydro-power systems, when realistically considered, usually result in non-linear, non-convex optimization problems of high dimension which can be difficult to solve using most optimization techniques. Our goal is to use a special form of potential function called the Expanded Lagrangian Function combined with the trust region algorithm to solve large-scale optimization problems arising in the applications of water resources management problems.

Our trust region algorithm uses a linear combination of an inexact Newton's direction and a steepest descent direction, to obtain a feasible descent direction. A bi-dimensional trust region scheme is used to obtain fast convergence. The inexact Newton's direction is obtained by solving a linear system of equations using a pre-conditioned conjugate gradient method which uses drop-tolerance pre-conditioner with RCM ordering.

The proposed method is tested on real data of 90 years of information for the Great Lakes water resources problem. The same application is solved with LANCELOT, using two different features of this software.

The results of the studies have shown that both algorithms converge to optimum objective values within a 3.0% difference from each other with LANCELOT providing worser objective values in most cases. Computer time required by both algorithms are comparable, with LANCELOT being somewhat slower.

The optimal storage levels and releases obtained from the proposed method when compared with past operations provide a significantly better operation.

Acknowledgment

Praise belongs to God who has been the source of inspiration, strength and confidence throughout my life and especially during the Ph.D Program.

I wish to thank my supervisor, Dr. Kumaraswamy Ponnambalam for the outstanding support and encouragement that he has provided me during my program at the University of Waterloo. Thanks are also due to my committee members, Dr. P.H. Calamai, D. Fuller, T. Vannelli and especially to my external examiner, Dr. M. Resende, for their time and constructive criticisms.

I am grateful to a number of people who have assisted me during my research including: Dr. J. M. Martínez for various useful discussions, Dr. P. Calamai, who generously devoted much time and effort in reading my thesis and discussing the numerical implications of my work, Mr. R. Orchard for reading my thesis, Dr. A. Samani for his great encouragement and help during my research, Dr. S. Fletcher and Mr. M. Bessa for providing the historical data of the Great Lakes problem.

The author acknowledges the scholarship provided by the Ministry of Culture and Higher Education of Iran. The author also appreciates the financial support provided by National Science and Engineering Resource Council (NSERC) of Canada through research grants to Dr. Ponnambalam.

Finally, my deepest appreciation belongs to my mother for her personal support, patience and tolerance during many years of my studies.

Dedicated to

The Glory of Almighty God

My mother and my family members

and

In memory of my late father, Mir Mosayeb

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Many systems engineering techniques are used in the planning and operation of various water resources systems. While simulation methods have been very popular, optimization models have become more attractive. During the past two decades, much progress has been made in the theory and practice of constrained nonlinear optimization. Although there are many standardized optimization procedures capable of solving a wide range of problems, there remains a class of problems characterized by high dimensionality, nonlinearity and non-convexity which can pose great solution difficulties to most existing optimization techniques. This is important because in many applications of water resources, problems with thousands of unknowns and nonlinear and non-convex constraints have to be solved. The joint operation of a system of reservoirs is a fundamental as well as a complex problem in water resources management. If the system reservoirs are to be operated to serve multiple purposes such as water supply, hydro-power generation, recreation, etc, the often conflicting objectives further complicate this problem. This research develops a new approach to solve large-scale nonlinear optimization

problems arising in water resources problems. A case study is presented where the proposed optimization technique is used for determining the optimal operational policy of the Great Lakes.

1.1 Problem Statement

Large-scale water resources systems usually serve several purposes such as water supply, hydro-power generation, recreation, water quality improvement, flood control, fish and wildlife maintenance, and navigation. These multiple objectives can often be in conflict with one another and optimal planning and operation studies attempt to compromise between the different purposes. Many large projects with significant investments, require quantifiable targets to be estimated based on non-monetary factors. Hydro-power generation is an exception in the sense that the benefits of additional power production can be easily quantified in monetary terms on the basis of the market price of energy or the cost of alternative supply.

Multi-reservoir operation planning (MROP) can be modeled as a nonlinear, constrained optimization problem. The objective is to find the set of reservoir release policies that minimizes the fluctuation of water levels and water flows. If the inflows to the system are assumed known or predictable with certainty, the resulting problem becomes deterministic. The focus of the present research is on deterministic optimization techniques which are generally simpler to solve than their stochastic counterparts.

A typical MROP problem is characterized by an objective function which is nonlinear and can be either convex or non-convex while there are both linear and nonlinear constraints. The nonlinear constraints are due to the fact that reservoir releases depend on nonlinear functions of reservoir storages. The constraints consist

of, in addition, linear equalities describing the dynamics of the state transformation, derived from consideration of mass balance at all storages in the system. In addition, the variables of the optimization (reservoir storages and releases) are bounded from above and below based on physical, design or operational limitations.

The non-convexity characteristics mentioned above constitute the primary difficulty in the successful mathematical solution of the above problem. For small systems, there are many methods available to solve such problems and find a global minimum. For example, discrete dynamic programming is the most reliable method for obtaining the global optimum solution.

The real difficulty arises when the system is large with many variables and constraints. The development of improved algorithms and/or adaptation of existing algorithms for solving such problems is therefore a subject of active research.

A stochastic formulation, broadly speaking, is a better representation of a practical MROP problem since inflow is a stochastic variable. Explicit treatment of stochasticity and risk, however, can be difficult and complex, particularly in the case of large systems [57]. Several methods have been developed, including implicit stochastic and adaptive planning methods [29].

Implicit stochastic optimization involves the use of a long inflow series in a deterministic optimization model and the analysis of the results using a regression model to develop generalized control polices. Adaptive planning or the feed forward control method is based on the repeated application of a deterministic model using the best estimate of short term forecasts and periodic updating of the system variables as real-time information becomes available in the progress of operation.

1.2 Study Objectives

The main objective of this research is to identify, review, design and compare suitable optimization models to use in operation planning of large-scale multi-reservoir systems. The specific objectives are as follows:

1. To survey the use of the optimization procedures for the operational planning of the multi-reservoir systems.

   This covers the applications of the different fields of Linear Programming (LP), Dynamic Programming (DP) and Nonlinear Programming (NP).

2. To develop, implement and experiment with an Expanded Lagrangian Function combined with a modified trust region method to provide the **Proposed Method (PM)**. This includes:

   - Studying some special cases of constrained optimization problems and summarizing the Expanded Lagrangian Function for these cases.
   - Proving the global convergence of the modified trust region algorithm.
   - Testing the PM on some benchmark application of hydro-power water resources problem with real data and comparing the output result with one of the current software packages.

3. To formulate and solve MROP problems as a large-scale nonlinear optimization problem, and to implement the PM on these formulated problems, specifically for the application of the Great Lakes water regulation problem. This requires:

   - Taking advantage of the sparsity pattern of the structure of the application of MROP in order to solve the linear systems of equations efficiently.

- Solving the existing application with similar alternatives in order to compare the output performance of the PM with a currently available software package.
- Deriving an operating policy for the releases for a long period of time.

1.3 Organization of the Dissertation

This dissertation is organized as follows:

Chapter 2 Nonlinear Programming Problem (NLP): Several implementations of optimization techniques for solving water resources management problems are considered. We briefly review the advances on methods based on Linear Programming (LP), Dynamic Programming (DP), and Nonlinear Programming (NP). The advantages and disadvantages of these methods are discussed.

Chapter 3 NP Problem Formulation: A class of algorithms based on the eventual solution of the first order necessary conditions for constrained optimization is explained. An expanded form of equations whose optimum solutions is a Karush-Kuhn-Tucker (KKT) point is introduced. The expanded form of equations is simplified for constrained optimization with bound constraints and a special case of convex quadratic optimization is also presented. This chapter also presents a literature review on trust region algorithms for nonlinear equations and introduces a new modification of a bi-dimensional trust region algorithm to solve unconstrained optimization problems and least square problems.

Chapter 4 The Implementation of the Proposed Method: The methodology used to solve constrained optimization methods is explained. There are two main phases for the algorithm. In the first phase, line search methods along with the BFGS method is used to obtain a feasible solution which is close to the optimal

solution. In the second phase, the Expanded Lagrangian Method is used to follow the penalty path efficiently. Some small test problems including one application of a water resources problem of medium size are solved with this algorithm and the results are compared with some other methods which use the Expanded Lagrangian Method.

Chapter 5 Case Studies: This chapter introduces the Great Lakes problem as a large-scale application to test the proposed method. Numerical results are presented along with the performance of the software package **LANCELOT**. We show that the proposed method is capable of solving the Great Lakes mathematical formulation problem with highly nonlinear and non-convex constraints in a reasonable amount of time. This development allows us to derive the optimum releases and storages of each lake for a long period of time in order to minimize the fluctuation of water during the operation of the system. A rolling horizon policy is employed in order to find the optimal releases and storages for each lake over a period of 90 years. Also, a portfolio optimization formulation from the field of economics is presented to test the proposed method for a simple case and also to test the properties of the iterative method used to solve the resulting large linear systems.

Chapter 6 Summary, Conclusions and Future Research: Chapter 6 presents a summary and conclusion for this research and provides suggestions for future research directions relating to the work developed in this dissertation.

Chapter 2

Optimization Methods in MROP

The optimum operation of multi-reservoir systems is a subject of great practical and economic interest in the field of water resources problems. With advances in computer technology, systems engineering techniques are rapidly being used in the planning, operation and management of modern-day large-scale water systems. To some extent, this is reflected by the abundance of research publications on this subject in engineering and related literatures in the past few decades. A complete review of all previous work on this subject would be a gigantic task. The scope of the present review will therefore be confined to the coverage of deterministic optimization procedures in the fields of linear programming and nonlinear programming applied to the planning and real-time operation of multi-reservoir water systems for hydro-power and closely related purposes. In addition, only methods with potential for broad application will be discussed.

2.1 Linear Programming

Linear Programming (LP) is perhaps the most widely used optimization technique amongst engineers. Its popularity, to some extent, is due to the existence of standard programming packages capable of solving large problems in a reasonable amount of time and at a reasonable cost. LP can handle a large number of variables and constraints but requires that the relationships between these variables be linear, both in the objective function and the constraints. Dantzig [13] first developed the Simplex algorithm for the solution of LP problems. Experimental studies show that the number of iterations of the Simplex method increases proportionally with the number of constraints and very slowly with the number of variables [68]. However, it is quite possible to make an example to show that the Simplex method can require exponential time to converge to an optimal solution [33]. The possibility of exponential complexity for the Simplex method created the need for much work to look for a **polynomial time algorithm**. Karmarkar [32] is believed to be the first to provide a practical polynomial time algorithm which comes from the class of algorithms called the **interior-point methods (IPM)**.

Purely linear programming problems in water resources are rare. Most problems are nonlinear in nature. Some applications of water resources problems have simple formulation in nature so that it is quite possible to use linear programming directly. Ellis and Revelle [18] present a deterministic, separable, linear algorithm for maximizing aggregate hydro-power production. In their iterative method, they used standard linear programming software to test several applications involving a hypothetical single-purpose hydro-power reservoir and 20-year monthly flow record from the Gunpowder River in Maryland. The separable linearized forms were solved quickly using Mathematical Programming System Extended (MPSX) software on

a variety of IBM hardware.

Srinivasan and Simonovic [70] present a new reliability model for planning the operation of a multi-reservoir system for hydro-power generation and flood control considering the stochastic nature of inflows.

In their proposed method, they maximize the benefits of hydro-power generation and minimize the economic losses due to reservoirs not meeting required reliabilities for hydro-power supply and flood control. This algorithm also uses a linearization technique to approximate the nonlinear energy function and determines the optimal reservoir release policy along with optimal reliabilities satisfying the hydro-power demand and providing the required flood control storage.

Most water resources problems, however, are inherently nonlinear in structure and therefore LP cannot be directly applied. This is especially true of hydro-power system where energy generation is a non-separable function. In order for standard LP to be implemented for this type of problem, a suitable linearization scheme must be introduced. Depending on the nature of such non-linearization, there are at least two methods for problem linearization.

The first approach, known as piecewise linearization (PL) converts the original nonlinear function to a series of linear functions with the definition of additional variables. Sun *et al.* [72] use the PL method on their implementation. Using a generalized network formulation, they incorporate the nonlinear evaporation loss function of a reservoir in a water-supply-optimization model. Then, they piecewise-linearize the nonlinear function and represent the resulting line segments by a series of arcs. An appropriate arc-flow multiplier is used to preserve the order of arc-flow fulfillment. They solve this model by an embedded generalized network algorithm (EMNET), which is designed to solve a linear programming problem with an em-

bedded generalized network substructure.

The second method involves repeated applications of LP to solve a series of approximate problems in which the original objective function is linearized. A widely adopted linearization scheme is based on the first-order Taylor's series expansion of a nonlinear function about a given initial solution. This method is known as **Successive Linear Programming (SLP)** and is more general as it can cope with the introduction of additional variables.

Rezniceck and Simonovic [63] introduce an algorithm for hydro-power optimization named EMSLP (energy management by successive linear programming). In the first stage, the EMSLP algorithm solves the underlying LP problem by successively updating the estimated storage values. In the second stage, it attempts to improve the objective function by decreasing the storage variability. The EMSLP model is considered to be a successful experience with the use of SLP in practice.

Tao and Lennox [73] apply SLP to the operation of the High Aswan Dam on the Nile River Basin. They report that the feasible solutions of a reservoir system lie on the hyper-plane determined by continuity equations. The global optimum is most likely given by a non-extreme point on that hyper-plane if the performance index is nonlinear and monotonic over the region defined by the bounds of storages and releases.

SLP usually works reasonably well for problems with lower degrees of nonlinearity on the objective function and the constraints. The convergence proof of SLP is given in [54] for the case of linear constraints. The convergence theorem states that if the objective function is continuous and differentiable, the sequence of solutions obtained by SLP will converge to a stationary point for the original problem, provided that a proper step-bound reduction scheme is chosen. The step-

bound limitation relates to the maximum allowable change in the decision variables between two successive iterations. In the next section we will describe **Dynamic Programming (DP)**, a method that is as widely used in water resources management problems as LP.

2.2 Dynamic Programming

Dynamic programming (DP) is one of the most widely used optimization techniques in reservoir operation studies. Early works started with a form of stochastic dynamic programming based on calculus of variations. The major breakthrough in dynamic programming was due to Bellman [3], who developed what is now called discrete dynamic programming. He defines dynamic programming as "the theory of multistage decision processes ". An advantage of dynamic programming over other methods such as linear programming is that a problem is solved one stage at a time. The computational burden therefore increases linearly with the number of stages. Another advantage of DP is that there is no restriction of any kind on the type and form of the objective value function and the solutions obtained are in general globally optimum.

Standard DP can be formulated either in the continuous or discrete form. The discrete form of DP, however, is more popular for most applications in water resources systems for the reason of simplicity. In discrete dynamic programming, the computation burden is dependent on the number and discretization of the state variables. For a system with n state variables and m levels of discretization in each state variables, there are m^n combinations that have to be explicitly evaluated at each stage of analysis. The Computational burden of DP therefore increases exponentially with the number of state variables. This characteristic is called the

curse of dimensionality and is the greatest weakness of DP. A lot of research has focused on ways of reducing the computational burden and/or circumventing the dimensionality problem.

A traditional but simplistic procedure for reducing the computational effort of DP is the coarse grid method. With this procedure, the problem is first solved using a coarse discretization of the state variable. Based on the resulting solution, revised bounds on the state variables are refined and the grid size is then reduced. The iterative procedure is repeated until the grid size has been reduced to the desired precision and when no further improvement in the objective function value is possible. This procedure, however, cannot guarantee a global optimum solution and it also does not resolve the dimensionality problem.

A technique that truly overcomes the dimensionality problem is the Dynamic Programming Successive Approximation Method (DPSA). Trott and Yeh [75] were the first to use this method in reservoir operation study. In this method, an initial guess of the state trajectory is first made. By allowing only one of the state variables to vary while keeping all the remaining variables fixed, the resulting problem is solved as a one dimensional DP problem. After an optimum solution is obtained, another variable is chosen to vary and the procedure is repeated until convergence. Trott and Yeh [75] solve a six-reservoir problem with this approach and claimed convergence within a reasonable number of iterations. In addition to not guaranteeing a globally optimum solution, the major weakness of DPSA is its inability to cope with common constraints such as minimum flow requirements downstream of two parallel reservoirs or total system generation demands.

Howson and Sacho [31] developed another variant of DP known as the progressive optimality algorithm (POA) for solving multi-state DP problems. The work was later expanded by Turgeon [76] and applied to the short term hydro-power

scheduling of a system of four reservoirs in series. The principle of progressive optimality is based on the premise that a particular trajectory is optimal, if and only if every decision set is optimal with respect to its initial and terminal states. The procedure is iterative in nature but does not require discretization of the state space. For a particular stage, say t, the operation of all reservoirs during time t is optimized by temporarily fixing the values of state variables at time $t+1$ and $t-1$. The resulting single stage optimization problem involving all reservoirs can be solved using iterative linear programming or nonlinear programming. The stage by stage optimization is implemented for all stages resulting in an improved/updated state trajectory. The entire procedure is then repeated until it effectively overcomes certain criteria. Marino and Loaiciga [45] also adopt POA to the optimal operation of Northern Central Valley, California which has a total of nine reservoirs.

The above literature review on the use of dynamic programming has been restricted to deterministic applications. It is necessary to mention that there are many applications of DP to stochastic problems. In fact, discrete dynamic programming is particularly suited to stochastic applications when compared to all other optimization procedures provided that the systems do not comprise too many state variables. Perera [55] uses stochastic dynamic programming (SDP) to determine the operating rules in terms of reservoir targets for urban water supply reservoir systems. The SDP explicitly accounts for the stochastic nature of a stream-flow to water supply system by considering a theoretical probability distribution for stream-flow. Perera provides a theoretical convergence for the global minimum of his method and applies it to the Melbourne urban water supply system in Australia to produce the reservoir targets. He concludes that the SDP provides an objective method of deriving satisfactory and acceptable operating values.

Recently, there has been interest in the use of a neural network procedure based

method to find the optimal reservoir operating policies. Raman and Chandramouli [61], for example, applied different methods of DP, SDP and standard operating policy (SOP) and the neural network procedure based method on the dynamic programming algorithm for the Aliyar Darn in Tamil Nadu, India and report promising results.

2.3 Nonlinear Programming

Nonlinear programming (NLP) is the most generalized deterministic mathematical programming technique. In this section, we study the recent advances on the implementation and the models of NLP as they apply to water resources problems.

Many nonlinear programming formulations for the optimization of the multi-reservoir operation of the hydro-power system exist. Tejada-Guibert *et al.* [74] develop a model for the optimization of the multi-month operation of the hydro-power system of the California Central Valley Project called CVPOP. CVPOP includes the dependence of energy values within each month on the capacity factor of the generating unit. This disallows the simplification of assuming constant monthly or yearly values, as is common in other models. The model also includes contractual energy and capacity constraints which are nonlinear because of the power-plant variables. Results indicate that large problems stemming from complex configurations of water resource systems and from diverse physical, economic and operational conditions, often of an unequivocally nonlinear nature, may be solved with nonlinear programming techniques using currently available commercial systems.

The implementation of Sequential Quadratic Programming (SQP) which provides a super-linear convergence for many applications is practically used for most small and medium size problems. Although SQP usually converges faster than

SLP for most of the problems in small and medium sizes there have been only a few people who have used SQP for the applications of water resources problems. Diaz and Fontane [15] use SQP to find the optimal allocation of power-plant releases during peak demand periods. This carries an economic advantage in the operation of hydro-power systems interconnected to large electrical networks. They exploit the concave characteristic of the nonlinear objective function and show a rapid convergence to the global optimum. They test their methodology on an existing multi-reservoir hydro-power system in Argentina and report very encouraging results. They also compare their results with SLP and report a faster rate of convergence for SQP.

Fletcher [23] uses SQP for a highly nonlinear and non-convex problem. The methodology introduces a new method for the first and the second moments of the storage state distribution in terms of the moments of inflow distribution. In comparison with other methods which do not consider the minimum and maximum bound storages and releases in their stochastic formulation, this method is able to handle bound constraints. The new formulation is highly nonlinear in terms of the objective function and constraints. Fletcher implements his algorithm to the Great Lakes Problem in Canada using the SQP method and provides the mean and variance of storages and releases of the application.

One of the highly successful software packages called GAMS/MINOS employs a *projected Lagrangian* [6] on a sequence of *linearly constrained subproblems* to solve problem with nonlinear constraints and objective function. During the past decade, GAMS/MINOS has successfully been used for different applications of water resources problems.

Peseshk *et al.* [56] presents a nonlinear optimization model to minimize pumping costs for both a well field and a main water supply distribution system. They

consider individual losses, pump efficiencies, and hydraulic losses in the pipe network. The model is established based on the fact that when demand served is less than the total capacity, there is a potential for reducing costs in the selection of pumps to meet the demand. The NLP model is solved using the general nonlinear optimization program MINOS. Peseshk *et al.* claim that for a given demand, the optimization procedure provides the best combination of pumps to meet the demand.

Ostfeld and Shamir [52] develop a model for the optimal operation of a multi-quality water supply system, under steady-state conditions. The system contains sources of different qualities, treatment facilities, pipes, and pumping stations. The objective is to minimize total cost, while delivering to all consumers the required quantities at acceptable qualities and pressures. They use a special approximation of the equation for water quality in pipes which enables the model to select the flow direction in pipes as part of the optimization. The steady-state example that they use consists of six consumers from three sources, two of them with treatment plants, and has three pumping stations and 10 pipes. The problem is solved using GAMS/MINOS for a base run and four additional runs aimed at studying the effects of modifications in key data. Ostfeld and Shamir report that the optimal solutions of the five cases demonstrate response to changes in economic and operational conditions.

In practice MINOS works reasonably well for applications with lots of linear constraints and few highly nonlinear constraints. During the past decade there has been many practitioners in water resources areas who are using MINOS with GAMS as an interface. (See for examples [53, 44, 71, 35]).

A water distribution system can be also represented by a **Network** where the nodes are consumption points, intermediated points, and supply or reservoir regu-

lation points. The links are pipes that conduct water between adjacent nodes. The flow through a pipe link is dependent on the pipe diameter and length, the difference in energy between adjacent nodes, and material friction characteristics. The cost of the water distribution system is related to the pipe dimensions and supply capacities. Santana and Soares [66] develop a method using a "branch and bound"type algorithm to solve the problem of least cost water distribution systems design. For each branch, they obtain the solution of the relaxed optimization by a generalized reduced gradient technique in the space of the pipe diameter variables. For each set of diameters they solve the Hazen-Williams nonlinear flow balance equations using a Newton-Raphson algorithm in the space of the node head variables. Santana and Soares test their method on a simple five-node artificial water distribution network design problem with a single load condition and report encouraging results.

During the past few years, there has been tremendous interest in the recent development of the software based on **Trust Region** Algorithms. Large and Nonlinear Constrained Extended Lagrangian Optimization Techniques **(LANCELOT)** has been one of the successful software packages designed for the purpose of general nonlinear programming problems. LANCELOT is a standard Fortran 77 package for solving large-scale nonlinear constrained optimization problems. By contrast to the highly successful package MINOS, which is particularly appropriate for large problems where the number of nonlinear degrees of freedom is modest, the emphasis in LANCELOT is on problems which are significantly nonlinear, in the sense that they involve a large number of nonlinear degrees of freedom [11]. However, the fact that LANCELOT is effective for large-scale applications does not mean that it is not appropriate for solving problems with only a few nonlinear degrees of freedom.

Since the early development of LANCELOT, there has been a great interest in the collection of different test problems. The Constrained and Unconstrained

Testing Environment (**CUTE**) is a suite of Fortran subroutines, scripts and test problems for linear and nonlinear optimization problems. This library contains over 800 different test problems for the purpose of comparison with existing software packages. Among the 800 test examples, there are a few problems related to water resources regulation. For example **DALLASL.SIF** is a large Dallas water distribution problem with 667 linear equality constraints and 906 bound constraints which can be found in the CUTE library. However, we are interested in very large-scale problems with nonlinear equality constraints and nonlinear objective function as the case study.

2.4 Summary

It is evident from the above literature review that a wide variety of techniques have been developed and applied to the optimization studies of multi-reservoir water resources systems. More algorithms can be expected to unveil themselves in the future since this is a field of active research and development. Three different research areas mainly based on LP, DP and NLP have been reviewed. We have explained that SLP is used very successfully in the applications of water resources problems. The fact that there has been many applications of water resources problems with nonlinear objective functions and constraints has motivated much interest in the use of NLP for these applications. However, there has been limited documented research on recent developments on the theory and software packages of NLP.

Chapter 3

NP Problem Formulation

3.1 Introduction

In 1988, Poore and Al-Hassan [58] introduced new algorithms for solving constrained optimization problems by using several path-following algorithms. These algorithms combined smooth penalty functions (the quadratic penalty for equality constraints, and the quadratic loss and logarithmic barrier functions for inequality constraints) and their modern counterpart, the augmented Lagrangian function. They solved these with continuation methods. In the first phase, they minimized an unconstrained or linearly constrained penalty function, or augmented Lagrangian to find a point which is reasonably close to the feasible region and then, in the second phase, they used predictor-corrector continuation methods to follow the path.

This chapter presents the derivation of the Expanded Lagrangian Function for general nonlinear programming. The quadratic penalty function is considered for the equality constraints and the logarithmic barrier function is used for the inequality constraints. A simple form of constraint qualification is given which represent a

perturbation of the Fritz John first-order necessary conditions. The existence and the regularity of the penalty path is also studied. A specific form of the optimization problem is presented and the expanded Lagrangian equations are summarized in an elegant form.

Consider the following general nonlinear programming problem,

$$\min\{f(x)|h(x)=0, g(x)\geq 0\} \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^1, h : \mathbb{R}^n \to \mathbb{R}^q$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ are assumed to be twice continuously differentiable.

Definition 3.1 *The Lagrangian function associated with* (3.1) *is defined to be:*

$$L(x,\lambda,\mu,\mu_0) \;:=\; \mu_o f - h^\top\lambda - g^\top\mu \tag{3.2}$$

where the components of $\lambda \in \mathbb{R}^q$, $\mu \in \mathbb{R}^p$, *and* $\mu_0 \in \mathbb{R}$ *are called the Lagrange multipliers.*

Define the mixed quadratic penalty-logarithmic barrier function to be:

$$P(x,r) = f(x) + \frac{1}{2r}h^\top(x)h(x) - r\sum_{i=1}^{p}\ln(g_i(x)). \tag{3.3}$$

where r is the penalty parameter. Here, the equality and inequality constraints are incorporated into the penalized objective function $P(x,r)$ through the use of smooth penalty functions. We describe the penalty path as the solution set of $\min P(x,r)$ as r changes and it must be a solution of $\nabla P = 0$ by the first-order necessary conditions. The gradient of the penalty function ∇P can be formally identified with that of the Lagrangian ∇L by defining the Lagrange multipliers appropriately. Using the definition of these multipliers as additional equations, one

can obtain an expanded Lagrangian system (ELS) of nonlinear equations which, with an additional modification, becomes a perturbation of the Fritz John first-order necessary conditions. This modification prevents unbounded multipliers. Two different systems, one based on the quadratic penalty-logarithmic barrier function and the other based on the quadratic penalty-quadratic loss function, have been derived and analyzed by Poore [58]. This chapter explains the one based on the quadratic penalty-logarithmic barrier function.

With regard to ill-conditioning, Poore and Al-Hassan [58] have shown that only three smooth penalty functions yield well-conditioned expanded Lagrangian systems. The canonical examples of these three classes are the quadratic penalty function for equality constraints and the logarithmic barrier function (an interior method) and quadratic loss function (an exterior method) for inequality constraints. The remaining smooth penalty functions introduce artificial singularities and ill-conditioning into the ELS and thus are not used in this dissertation.

The main features of this class of algorithms is that we first use an unconstrained optimization technique to get on the penalty path at a value of r, say r^0, where the problem is reasonably well conditioned. Then the expanded Lagrangian system is solved to reach optimality at $r = 0$.

3.2 The Optimality Conditions Defined by Nonlinear Equations

In order to have a better understanding of the theories that will be explained in this dissertation, we need to have an overview of some basic material. We review some of the necessary concepts such as the Lagrange function and the Karush-Kuhn-Tucker

first and second order necessary and optimality conditions.

The following theorem, due to Fiacco [19], states the Fritz John first order necessary conditions of problem (3.1).

Theorem 3.1 *Let f, g, and h in* (3.1) *be continuously differentiable on an open set containing the feasible region of* (3.1). *Suppose that x^* is a (feasible) local minimizer of* (3.1). *Then there exist scalars λ_i $(i = 1, \ldots, q)$, μ_0 , and μ_j $(j = 1, \ldots, p)$, not all zero, such that:*

$$\begin{cases} \mu_0 \geq 0, \mu \geq 0, g(x^*) & \geq & 0 \\ \nabla L(x^*, \lambda, \mu, \mu_0) & = & 0 \\ h(x^*) & = & 0 \\ Mg(x^*) & = & 0 \end{cases} \tag{3.4}$$

where $M = diag(\mu)$, $\mu = (\mu_1, \ldots, \mu_p)$ and $L = \mu_0 f - h^\top \lambda - g^\top \mu$

Note that if $\mu_0 = 0$, then the contribution of the objective function is eliminated in these optimality criteria. For this reason we need to add some additional assumptions to ensure that $\mu_0 > 0$ and without loss of generality, μ_0 can be taken to be 1. In other words, we need to confirm that some constraint qualification holds for these equations.

There are several constraint qualifications [19], and the one used in the following theorem is called the linear independence constraint qualification (LICQ). This leads to the results called the Karush-Kuhn-Tucker first order necessary conditions. The following theorems (3.2) to (3.6) are due to Poore and Al-Hassan [58].

Theorem 3.2 *Suppose that the functions f, h and g are continuously differentiable on an open set containing the feasible region. Let x^* be a feasible point of the*

nonlinear programming problem (3.1) *and define* $I(x^*) = \{i | 1 \leq i \leq p : g_i(x^*) = 0\}$ *to be the set of active constraints. Assume that the following linear independence constraint qualification holds at* x^* : *The set* $S := \{\nabla h_j(x^*)_{j=1}^{j=p}, \{\nabla g_i\}\{i \in I(x*)\}\}$ *is a set of* $p + |I(x^*)|$ *linearly independent vectors where* $|A|$ *denotes the cardinality of the set* A. *Then the necessary conditions that* x^* *be a local minimizer to* (3.1) *are given by equations* (3.4) *with* $\mu_0 = 1$.

The second order sufficient conditions require the following definitions of the index sets $\overline{\mathcal{A}}$ *and* $\mathcal{A}$, the tangent space $\overline{T}$, and a tangent cone $\overline{C}$:

$$\overline{\mathcal{A}} = \{j : 1 \leq j \leq q, g_j(x^0) = 0\} \qquad \mathcal{A} = \{j \in \overline{\mathcal{A}} : \mu_j > 0\}$$

$$\overline{T} = \{y \in \mathbb{R}^n : y^\top \nabla h_j(x^0) = 0 \ (j = 1, \ldots, q), y^\top \nabla g_j(x^0) = 0 \ (j \in \overline{\mathcal{A}})\}.$$

$$\overline{C} = \{y \in \mathbb{R}^n : y^\top \nabla h_j(x^0) = 0 \ (j = 1, \ldots, q), y^\top \nabla g_j(x^0) = 0 \ (j \in \mathcal{A}),$$

$$y^\top \nabla g_j(x^0) \geq 0 \ (j \in \overline{\mathcal{A}} - \mathcal{A})\}$$

Theorem 3.3 *Let* f, h, *and* g *in the nonlinear programming problem* (3.1) *be* C^2 *in a neighborhood of the feasible region and let* x^* *satisfy the first order necessary conditions of Fritz John. Suppose further that*

- $\overline{\mathcal{A}} = \mathcal{A}$.
- *the linear independence constraint qualification in theorem 3.2 is satisfied.*
- *the Hessian of the Lagrangian* $\nabla^2 L$ *is positive definite on the tangent space* $\overline{T}$.

Then x^* *is a local minimum to the nonlinear programming problem 3.1.*

This theorem closely parallels the theory to be developed in the next section. An example of a theorem with weaker hypotheses is:

Theorem 3.4 *Let $f, h,$ and g be C^2 in a neighborhood of the feasible region and suppose that Fritz John first order necessary conditions are satisfied at a feasible point x^*. Then x^* is a local minimizer of* (3.1) *provided the Hessian of the Lagrangian is positive definite on the tangent cone $\overline{C}$.*

3.3 The Expanded Lagrangian System

This section presents the Expanded Lagrangian System (ELS) for the general nonlinear programming problem. The mixed quadratic penalty-logarithmic barrier function is expanded to a set of nonlinear equations which is the perturbation of the Fritz John first order necessary conditions. The existence and regularity of the penalty path have already been explained by Poore and Tiahrt [60] and are not repeated here. Throughout this section f, g, and h are assumed to be at least C^1. The mixed quadratic penalty-logarithmic barrier function is

$$P(x,r) = f(x) + \frac{1}{2r}h^{\mathsf{T}}(x)h(x) - r\sum_{i=1}^{p}\ln(g_i(x)). \tag{3.5}$$

Assume $\{x : g(x) > 0\}$ to be nonempty. The first-order necessary condition for a minimum of P is that $\nabla P = 0$, which is a parameterized system of nonlinear equations. However, as we explained before, this system suffers from numerical problems when $r \to 0^+$, the Jacobian of ∇P, the Hessian of P, becomes increasingly ill conditioned as $r \to 0^+$. (The ℓ_2 condition number $k_2(\nabla^2 P) = O(\frac{1}{r})$ *as* $r \to 0^+$.) To remove the ill-conditioning, the equations $\nabla P = 0$ are expanded as follows : Assuming $g(x) > 0$ and $r > 0$ then (x, r) solves $\nabla P = \nabla f + \nabla h^{\mathsf{T}}\left[\frac{h}{r}\right] - \sum_{i=1}^{p}\nabla g_i\left[\frac{r}{g_i}\right] = 0$

if and only if (x, λ, μ, r) solves the expanded system

$$\begin{cases} \nabla f(x) - \nabla h(x)^\top \lambda - \nabla g(x)^\top \mu &= 0 \\ h(x) + r\lambda &= 0 \\ Mg(x) - re &= 0, \end{cases}$$

where $M = diag(\mu_1, \ldots, \mu_p)$ *and* $e = (1, 1, \ldots, 1)^\top \in \mathbb{R}^p$. (The last two equations are derived from the definitions $\lambda = -h/r$ and $\mu_i = r/g_i(x)$.) Fiacco and McCormick [19] were the first to use these equations to investigate the behavior of the penalty path near $r = 0$. However, these equations had never been used actively in the literature for nonlinear programming until the work of Poore and Al-Hassan [58] who used continuation methodology to solve them. The system (3.3) may still suffer from numerical deficiency because a multiplier may tend to infinity when either a constraint cannot be satisfied or a constraint qualification fails [58]. Also the use of shifts in the barrier function can sometimes cause a multiplier μ_i to tend to infinity at a positive value of r.

Theorem 3.5 *Let* $f : \mathbb{R}^n \to \mathbb{R}^1, h : \mathbb{R}^n \to \mathbb{R}^q$, *and* $g : \mathbb{R}^n \to \mathbb{R}^p$ *be* C^1 *functions, and suppose* β_0 *is a given nonzero real number. Then when* $r \neq 0$ *and* $g(x) > 0$, *component wise,* $(x; r)$ *solves*

$$\nabla P = \nabla f + \nabla h^\top \left[\frac{h}{r}\right] - \sum_{i=1}^{p} \nabla g_i \left[\frac{r}{g_i}\right] = 0 \tag{3.6}$$

if and only if for the given $\beta_0 \neq 0$ *there exists a solution* $(x, \lambda, \mu, \mu_0; \bar{r}) \in \mathbb{R}^{n+p+q+2}$ *with* $\mu_0 > 0$ *of*

$$F(x, \lambda, \mu, \mu_0; \bar{r}) = \begin{cases} \nabla L(x, \lambda, \mu, \mu_0) &= 0 \\ h(x) + \bar{r}\lambda &= 0 \\ Mg(x) - \mu_0^2 \bar{r} e &= 0 \\ \mu_0^2 + |\mu|_2^2 + |\lambda|_2^2 - \beta_0^2 &= 0 \end{cases} \tag{3.7}$$

where $L = \mu_0 f - h^\top \lambda - g^\top \mu$, $M = diag(\mu_1, \ldots, \mu_q)$, $\mu = (\mu_1, \ldots, \mu_p)^\top$, $\lambda = (\lambda_1, \ldots, \lambda_q)$, *and* $\overline{r} = r/\mu_0$.

Proof:Suppose equation (3.6) *is valid. Multiply through by* μ_0 *and choose* $\mu_0 > 0$ *so that* $\mu_0^2 = \beta_0^2[1 + |h/r|_2^2 + |r/g|_2^2]^{-1}$. *Then define* $\overline{r} = r/\mu_0, \lambda = -h/\overline{r}$, $\mu_i = (\mu_0^2 \overline{r})/g_i$, *so that* $(x, \lambda, \mu, \mu_0; \overline{r})$ *now solves the expanded system* (3.7). *When* $\mu_0 > 0$ *and* $(x, \lambda, \mu, \mu_0; \overline{r})$ *solves* (3.7), *the reduction to* (3.6) *also follows directly from these definitions once the normalization has been dropped.* □

It is clear that the system of (3.7) can be changed to (3.3) only by setting $\mu_0 = 1$ and dropping the normalization. Note that if x^0 is a solution of $\min P(x, r_0)$, $\lambda_0 = -h(x^0)/r_0$, $\mu_i^0 = r_0/g_i(x^0)$, and $\beta_0 = [1 + |\mu^0|_2^2 + |\lambda^0|_2^2]^{1/2}$, then $(x, \lambda, \mu, \mu_0; \overline{r})$ $= (x^0, \lambda^0, \mu^0, 1; r_0)$ is a solution of $F = 0$, and assuming no singularities are encountered, one can follow the penalty path $(x(r) = \min P(x, r))$ to $r = 0$ by following the solution of $F = 0$ to $\overline{r} = 0$. Given this formulation the necessary and sufficient conditions for the system (3.7) to be regular at $\overline{r} = 0$ are given.

Theorem 3.6 *Let the system (3.7) be denoted by* $F(z; \overline{r}) = 0$ *with* $z = (x, \lambda, \mu, \mu_0)$. *Let* $(z^0; 0)$ *be a solution of* $F = 0$ *and assume* f, h *and* g *are twice continuously differentiable in a neighborhood of* x^0. *Define two index sets* $\overline{\mathcal{A}}$ *and* $\mathcal{A}$ *and a corresponding tangent space* $\overline{T}$ *by*

$$\overline{\mathcal{A}} = \{j : 1 \leq j \leq q \ , \ g_j(x^0) = 0\} \qquad \mathcal{A} = \{j \in \overline{\mathcal{A}} : \mu_j > 0\}$$

$$\overline{T} = \{y \in \mathbb{R}^n : y^\top \nabla h_j(x^0) = 0 \ (j = 1, \ldots, q) \ , y^\top \nabla g_j(x^0) = 0 \ (j \in \overline{\mathcal{A}})\}.$$

A necessary and sufficient condition that the Jacobian $D_z F(z^0; 0)$ *be nonsingular is that each of the the following three conditions hold:*

- $\overline{\mathcal{A}} = \mathcal{A}$; *(Stirct Complementarity)*

- $S := \{\{\nabla g_i(x^0), i \in \overline{\mathcal{A}}\} \cup \{\nabla h_j(x^0)_{j=1}^p\}\}$ *is a linearly independent collection of* $p + |\overline{\mathcal{A}}|$ *vectors where* $|\overline{\mathcal{A}}|$ *denotes the cardinality of* $\overline{\mathcal{A}}$*;*

- *The Hessian of the Lagrangian* $\nabla^2 L$ *is nonsingular on the tangent space* $\overline{T}$ *at* z^0.

  If $D_z F(z^0; 0)$ *is nonsingular, there exist neighborhoods* β_1 *of* $\bar{r} = 0$ *and* β_2 *of* $(z^0; 0)$ *and a function* $\phi \in C^1(\beta_1)$ *such that* $F(\phi(\bar{r}), \bar{r}) = 0$ *for all* $\bar{r} \in \beta_1$ $\phi(0) = z^0$*. This solution is locally unique in the sense that if* $(z; \bar{r}) \in \beta_2$ *and* $F(z; \bar{r}) = 0$*,then* z *belongs to the manifold defined by* ϕ*,i.e.,* $z = \phi(\bar{r})$.

Proof: The necessary and sufficient conditions for the non-singularity of the Jacobian $D_z F(z^0; 0)$ *have been established in the work of Poore and Tiahrt [60] in the context of the parametric programming problem and will not be reported here. The remaining part of the theorem follows from the implicit function theorem [10].*

Several remarks are in order: If x^0 is a Fritz John or Karush-Kuhn-Tucker point, then the first condition described above is called strict-complementarity ($g_i(x^0) = 0$ implies μ_i^0 is *nonzero*) while the second condition is the linear independence constraint qualification. Furthermore, if the first and second conditions are satisfied and the third condition is strengthened to the Hessian of the Lagrangian being positive definite on the tangent space $\overline{T}$, then we have a second-order sufficient condition for x^0 to be a local minimum provided $\mu^0 \geq 0$ and $g \geq 0$. Finally, it is noted that the theorem is valid regardless of the point type. It may be a local minimum, a saddle point, local maximum, feasible or non-feasible critical point.

3.4 Special Cases

3.4.1 Application with Equality and Bound Constraints

In this section and the next we consider some special cases with nonlinear and linear equality and inequality constraints respectively. We eliminate the bound constraints from the expanded Lagrangian System and simplify the equations. This idea has already been used in the context of linear programming and we have adapted this for nonlinear programming [5]. Let us consider the minimization of the general objective function subject to some general equality and box constraints.

$$\begin{aligned} \min \quad & f(x), && (3.8)\\ s.t. \quad & h(x) = 0, \\ & l \leq x \leq u && (3.9) \end{aligned}$$

As explained, the above system can be expanded to the following system,

$$\begin{cases} \nabla f - \nabla h^{\top}\lambda - \nabla g_1^{\top}\mu_1 - \nabla g_2^{\top}\mu_2 &= 0 \\ h + r\lambda &= 0 \\ \underline{M} g_1 - re &= 0 \\ \overline{M} g_2 - re &= 0, \end{cases} \tag{3.10}$$

where $g_1 = x - l$, $g_2 = -x + u$, $\underline{\mu}$ and $\overline{\mu}$ are Lagrange multiplier corresponding to lower and upper bound constraints respectively.

$$\underline{M} = \begin{bmatrix} \underline{\mu}_1 & & & \\ & \underline{\mu}_2 & & \\ & & \ddots & \\ & & & \underline{\mu}_n \end{bmatrix}$$

and

$$\overline{M} = \begin{bmatrix} \overline{\mu}_1 & & & \\ & \overline{\mu}_2 & & \\ & & \ddots & \\ & & & \overline{\mu}_n \end{bmatrix}$$

The Newton step of the above equations has the following form,

$$\begin{bmatrix} \nabla^2 L & \nabla h^\top & -I & I \\ \nabla h & rI & 0 & 0 \\ \underline{M} & 0 & \underline{N} & 0 \\ -\overline{M} & 0 & 0 & \overline{N} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta\underline{\mu} \\ \Delta\overline{\mu} \end{bmatrix} = - \begin{bmatrix} \nabla L \\ \nabla h + rI \\ \underline{M} g_1 - re \\ \overline{M} g_2 - re \end{bmatrix} \tag{3.11}$$

with

$$\underline{N} = diag(-l_1 + x_1, ..., -l_n + x_n)$$

and

$$\overline{N} = diag(u_1 - x_1, ..., u_n - x_n)$$

The last two equations of (3.11) can be summarized as

$$\underline{M}\Delta x + \underline{N}\Delta\underline{\mu} = -\underline{M} g_1 + re \tag{3.12}$$

and

$$-\overline{M}\Delta x + \overline{N}\Delta\overline{\mu} = -\overline{M} g_2 + re \tag{3.13}$$

Thus

$$\Delta\underline{\mu} = \underline{N}^{-1}(-\underline{M}\Delta x - \underline{M}g_1 + re) \tag{3.14}$$

and

$$\Delta\overline{\mu} = \overline{N}^{-1}(\overline{M}\Delta x - \overline{M}g_2 + re) \tag{3.15}$$

From the first equation of (3.11), we have

$$\nabla^2 L\Delta x + \nabla h^\top \Delta\lambda - \Delta\underline{\mu} + \Delta\overline{\mu} = -\nabla L \tag{3.16}$$

By Substituting (3.15) and (3.14) into (3.16) we get

$$(\nabla^2 L + \underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M})\Delta x + \nabla h^\top \Delta\lambda = -\nabla L + \overline{N}^{-1}(\overline{M}g_2 - re) - \underline{N}^{-1}(\underline{M}g_1 - re) \tag{3.17}$$

so the system of equations (3.11) is summarized to

$$\begin{bmatrix} \nabla^2 L + \underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M} & \nabla h^\top \\ \nabla h & rI \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{3.18}$$

where $b_1 = -\nabla L + \overline{N}^{-1}(\overline{M}g_2 - re) - \underline{N}^{-1}(\underline{M}g_1 - re)$ and $b_2 = -h - rI$; $\underline{N}$ and $\overline{N}$ are diagonal matrices and the inverse of these matrices are readily available. On the other hand, this systems can still keep the sparsity pattern of the equations. This modification enables us to handle bound constraints without any need to increase the number of equations in the expanded Lagrangian system. The application to the water resources problem that we study later involves such bound constraints. The next section explains how to summarize equation (3.10) when there are only linear equations and bound constraints with a quadratic objective function. The quadratic optimization problem we solve later in section (5.10) has this form.

3.5 Application with Strictly Convex Quadratic Objective Function and Linear Constraints

The convex quadratic problem with linear constraints has already been studied by different people. (See for example [50, 77]). The problem that we are concerned with in this section has the following form,

$$\begin{aligned} \min \quad & c^\top x + x^\top C x, && (3.19)\\ s.t. \quad & Ax = b, \\ & l \leq x \leq u && (3.20) \end{aligned}$$

where $x, c, b, l, u \in \mathbb{R}^n$, $C \in \mathbb{R}^{n \times n}$ and $A : \mathbb{R}^n \to \mathbb{R}^q$. We also assume that $C > 0$. As we can see, the objective function is quadratic so we can add the linear constraints as a quadratic penalty term to the objective function. This modification allows us to simplify the Expanded form of the equations very significantly as follows:

$$\begin{aligned} \min \quad & c^\top x + x^\top C x + \tfrac{(Ax-b)^\top (Ax-b)}{2r}, && (3.21)\\ s.t. \quad & l \leq x \leq u \end{aligned}$$

The expanded form of (3.21) has the following form,

$$\begin{cases} \nabla L = 0 \\ \underline{M} g_1 - re = 0 \\ \overline{M} g_2 - re = 0 \end{cases}$$

where $\nabla L = c + Cx + \frac{A^T Ax}{r} - \frac{A^T b}{r} - \underline{\mu} + \overline{\mu}$ and $\underline{\mu}$ and $\overline{\mu}$ are Lagrange multipliers corresponding to bound constraints.

$$\begin{bmatrix} \nabla^2 L & -I & I \\ \underline{M} & \underline{N} & 0 \\ -\overline{M} & 0 & \overline{N} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \underline{\mu} \\ \Delta \overline{\mu} \end{bmatrix} = - \begin{bmatrix} \nabla L \\ \underline{M} g_1 - re \\ \overline{M} g_2 - re \end{bmatrix} \tag{3.22}$$

The last two equations of (3.22) are,

$$\Delta \underline{\mu} = \underline{N}^{-1}(re - \underline{MN} - \underline{M}\Delta x) \tag{3.23}$$

$$\Delta \overline{\mu} = \overline{N}^{-1}(re - \overline{MN} - \overline{M}\Delta x) \tag{3.24}$$

Having used (3.23) and (3.24) in the first equation of (3.22) , we have

$$(C + \frac{A^T A}{r} + \underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M})\Delta x = -\nabla L - \underline{N}^{-1}(\underline{MN} - re) + \overline{N}^{-1}(\overline{MN} - re) \tag{3.25}$$

Theorem 3.7 *If $C > 0$, $0 < l < x < u$, $\underline{\mu} \geq 0$ and $\overline{\mu} \geq 0$, Then*

$$C + \frac{A^T A}{r} + \underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M}$$

is positive definite.

Proof 3.1 *For any nonzero $d \in \mathbb{R}$ we have, $d^T(C + \frac{A^T A}{r} + \underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M})d = d^T C d + d^T \frac{A^T A}{r} d + d^T(\underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M})d$ where $d^T C d > 0$, since $C > 0$ and, for $r > 0$, $d^T \frac{A^T A}{r} d \geq 0$ and finally since*

$$\underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M} = \begin{bmatrix} \frac{\underline{\mu}_1}{x_1 - l_1} + \frac{\overline{\mu}_1}{u_1 - x_1} & & \\ & \ddots & \\ & & \frac{\underline{\mu}_n}{x_n - l_n} + \frac{\overline{\mu}_n}{u_n - x_n} \end{bmatrix} \tag{3.26}$$

where $\frac{\underline{\mu}_i}{x_i - l_i} + \frac{\overline{\mu}_i}{u_i - x_i} \geq 0$, for $i = 1, \ldots, n$, we have $d^T(\underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M})d \geq 0$ which establishes the proof.

Algorithm (3.1) shows a simple Pseudo-code for the application discussed in this section. The term $\frac{A^T A}{r}$ in the objective function may remove the sparsity pattern of the system of equations. But at each minor iteration, only diagonal elements of the matrix $C + \frac{A^T A}{r} + \underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M}$ are changed. This is an important feature as one can use the Choleskey factorization at the beginning of each minor iteration to solve the equations and then use this factorization for the next minor iterations as a preconditioner and apply the conjugate gradient method. As we will show in our results, at each step of the algorithm, only a few conjugate gradient iterations are needed in order to solve the system of equations.

Algorithm 3.1

Initialize x, r *and the iteration counter* k.

$k = 0$, $r = r_o$, $x_o = (u + l)/2$, $\underline{M} = \underline{M^o}$, $\overline{M} = \overline{M^o}$, $\underline{N} = \underline{N^o}$ *and* $\overline{N} = \overline{N^o}$.

Do while $r < toler$

while not converged do

$k \leftarrow k + 1$;

if $k = 1$

$L \leftarrow chol(C + \frac{A^T A}{r} + \underline{N}^{-1}\underline{M} + \overline{N}^{-1}\overline{M})$.

Solve $LL^T \Delta x = -\nabla L - \underline{N}^{-1}(\underline{MN} - re) + \overline{N}^{-1}(\overline{MN} - re)$.

else

Apply L *as pre-conditioner, find* Δx *using Preconditioned Conjugate Gradient.*

end

Update $\Delta\underline{\mu} = \underline{N}^{-1}(re - \underline{MN} - \underline{M}\Delta x)$.

and $\Delta\overline{\mu} = \overline{N}^{-1}(re - \overline{MN} - \overline{M}\Delta x)$.

Compute the step-length α

Update $x \leftarrow x + \alpha\Delta x$.

Update $\underline{\mu} \leftarrow \underline{\mu} + \alpha\Delta\underline{\mu}$.

Update $\overline{\mu} \leftarrow \overline{\mu} + \alpha\Delta\overline{\mu}$.

end do

$r \leftarrow$ *reduce* r *and* $k \leftarrow 0$

end do

3.6 Trust Region Methods for Nonlinear Equations

Our main concern in this section is to use the concept of Trust Region Algorithms to solve the nonlinear equations arising in the Expanded Lagrangian Systems (3.7). We are interested in having a trust region algorithm with robust convergence properties and the ability to deal specifically with Sparse Nonlinear Least Square Problems. Despite the fact that there are many applications in large scale, there have been few algorithms and codes designed for this purpose. Fletcher [20, 22] was the first person who provided software to solve the nonlinear equations based on a modification of Levenberg's algorithm. Duff *et al.* [17] use linear programming combined with trust region idea to solve the nonlinear equations. It is based on minimizing the l_1-norm of the linearized vector within an l_∞-norm trust region, thereby permitting linear programming techniques to be easily applied. At each step d_k is a solution of the problem

$$\begin{aligned} &\min \| f_k + J_k d \|_1 \\ &s.t. \quad \| d \|_\infty \leq \Delta_k. \end{aligned} \tag{3.27}$$

This is equivalent to a linear program which can be expressed as

$$\min \sum_i (\zeta_i + \xi_i)$$

$$s.t. \quad J_k d + \zeta - \xi = -f_k, \tag{3.28}$$

$$\tag{3.29}$$

$$\left.\begin{aligned} \zeta \geq 0, \xi \geq 0 \\ -\Delta_k \leq d_i \leq \Delta_k \end{aligned}\right\} i = 1, 2, \ldots, n. \tag{3.30}$$

Note that the constraint matrix is

$$(J_k \quad I \quad -I) \tag{3.31}$$

so that a typical basis has a mixture of columns from J_k, I and $-I$ and its triangular factors will be sparser than those of J_k if a good pivotal sequence is used. This linear program may be solved by the simplex method or any other interior point algorithm. Duff *et al* [17] show that their method converges to an optimal solution for some test problems even when the Levenberg-Marquardt code NS03 [62] and the sparse dog-leg code of Munksgaard and Reid [51] fail to converge.

There are also other algorithms which keep the properties of rapid convergence in the low-residual cases and behave reasonably well in the general cases. For example, Martínez's algorithm [47, 48] represents a compromise between global convergence in the general case and rapid local convergence in the "easy"case. Its main features, described in algorithm (3.2), are the following:

1) At each iteration, the linear Least Squares problem

$$J(x_k)d_k \approx -F(x_k)$$

is incompletely solved using a Preconditioned Conjugate Gradient method. We may call this procedure an "Inexact Gauss-Newton "strategy.

2) The point x_{k+1} is obtained using a search procedure in the plane spanned by d_k and $\nabla(\frac{1}{2} \parallel F(x) \parallel^2)|_{x^k}$.

Algorithm 3.2 *Let* $F : D \subset \Re^n \to \Re^m$, $m \geq n$, $F \in C^1(D)$, D *an open set.* $x^o \in D$ *is an arbitrary initial point,* (η_k) *a sequence of strictly positive numbers such that* $\lim \eta_k = 0$, θ_1, $\theta_2 \in (0,1)$, $\theta_3 \in (0, \frac{1}{2}), \upsilon > 1$, *and* $0 < \underline{M} < \bar{M} < \infty$.

Also, let x_k be the k-th approximation to the solution obtained with this algorithm. Then denote

$$F_k = F(x_k),\ J_k = J(x_k),\ g_k = J_k^T F_k = \nabla(\frac{1}{2} \parallel F(x) \parallel^2) \mid_{x_k}$$

To obtain the new step x_{k+1}, perform in the case $g_k \neq 0$ the following steps:

Step 1: *Compute J_k and g_k. If $g_k = 0$, stop.*

Step 2: *Obtain $w_k \in \Re^n$ such that*

$$\parallel J_k^T J_k w_k + g_k \parallel \leq \eta_k \parallel g_k \parallel \tag{3.32}$$

Step 3: *Obtain $v_k \in \Re^n$ as the solution of the following bi-dimensional problem:*

$$\textit{Minimize} \quad \parallel J_k v + F_k \parallel$$

$$s.t. \quad v = \lambda_1 g_k + \lambda_2 w_k, \quad \parallel v \parallel \leq \parallel w_k \parallel .$$

Step 4: *Set $d_k^1 = -g_k$. Test the following two conditions for v_k:*

$$\langle v_k, g_k \rangle \leq -\theta_1 \parallel v_k \parallel \parallel g_k \parallel \tag{3.33}$$

and

$$\underline{M} \parallel g_k \parallel \leq \parallel v_k \parallel \leq \bar{M} \parallel g_k \parallel . \tag{3.34}$$

If equations (3.33) and (3.34) are satisfied, set $d_k^2 = v_k$, otherwise $d_k^2 = d_k^1$

Step 5: *Set $t = \parallel d_k^2 \parallel$. Perform steps (5a) to (5d).*

(5a) *Obtain d as the solution of the problem:*

$$\textit{Minimize} \quad \parallel J_k d + F_k \parallel$$

$$s.t. \quad d = \lambda_1 d_k^1 + \lambda_2 d_k^2, \quad \parallel d \parallel \leq t \tag{3.35}$$

(5b) *If*

$$\frac{1}{2} \| F(x_k + d) \|^2 \leq \frac{1}{2} \| F(x_k) \|^2 + \theta_2 \langle g_k, d \rangle \tag{3.36}$$

go to step 5d.

(5c) *Let* $\hat{t}$ *be such that*

$$\theta_3 t \leq \hat{t} \leq (1 - \theta_3)\, t$$

Replace t *by* $\hat{t}$*, Go to step 5a.*

(5d) $d_k = d, \quad x_{k+1} = x_k + d_k.$

To conclude this section we shall prove that this algorithm is well defined. The following proof is restated from [47].

We prove that when t is small enough, condition (3.36) is satisfied. Let us call $d(t)$ the solution of the minimization problem (3.35). Because, $d(t)$ is the solution of a two variable problem restricted to the bi-dimensional ball of radius t, therefore, $\lim_{t\to 0} \frac{d(t)}{\|d(t)\|}$ is the steepest descent direction of the function restricted to the plane spanned by d_k^1 and d_k^2. On the other hand, the steepest descent direction of the (unrestricted) function belongs to this plane ($d_k^1 = -g_k$). Therefore

$$\lim_{t\to 0} \frac{d(t)}{\| d(t) \|} = -\frac{g_k}{\| g_k \|}$$

Now if we define:

$$\varphi(d) = \frac{1}{2} \frac{(\| F(x_k + d) \|^2 - \| F(x_k) \|^2)}{\| d \|}$$

then we have by the Mean Value Theorem:

$$\varphi(d) = \frac{\langle g(x_k + \zeta d), d \rangle}{\| d \|}, \qquad 0 \leq \zeta \leq 1.$$

Thus $\varphi(d)$ tends to $-\parallel g_k \parallel$, when t tends to 0.

On the other hand, the expression $\frac{\langle g_k, d\rangle}{\|d\|}$, also tends to $-\parallel g \parallel$. Thus,

$$\lim \quad \frac{1}{2} \frac{(\parallel F(x_k + d) \parallel^2 - \parallel F(x_k) \parallel^2)}{\langle g_k, d\rangle} = 1,$$

and so

$$\frac{1}{2} \frac{(\parallel F(x_k + d) \parallel^2 - \parallel F(x_k) \parallel^2)}{\langle g_k, d\rangle} \geq \theta_2$$

if t is small enough. But since $\langle g_k, d\rangle < 0$ the inequality (3.36) holds for small t.

3.7 Global Convergence Results

For completeness, in this part some global convergence results are presented. Most of these are stated without proof as these results are directly from [47].

The results presented here also hold for any scalar function $f \in C^1(D)$, with $\nabla f(x) = g(x)$.

Let, $f(x) = \frac{1}{2} \parallel F(x) \parallel^2$.

Define

$$C(v, w) = \{x \in \mathbb{R}^n \mid x = \gamma_1 v + \gamma_2 w, \quad \gamma_1, \gamma_2 \geq 0\}$$

(the convex cone determined by v and w).

The algorithm (3.2) is a particular case of a more general algorithm, which is as follows:

Algorithm 3.3 *Given $x^o \in D$ an arbitrary initial point, consider the sequence defined recursively as follows:*

If $g_k \equiv g(x^k) \equiv 0$, *stop. Otherwise:*

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

where

$$d_k \in C(d_k^1, d_k^2), \quad \| d_k \| \leq \| d_k^2 \|, \tag{3.37}$$

$$\langle d_k^i, g_k \rangle \leq -\theta_1 \| d_k^i \| \| g_k \|, \quad i = 1, 2. \tag{3.38}$$

$$\underline{M} \| g_k \| \leq \| d_k^i \| \leq \bar{M} \| g_k \|, \quad i = 1, 2. \tag{3.39}$$

$$f(x_k + d_k) \leq f(x_k) + \theta_2 \langle g_k, d_k \rangle, \tag{3.40}$$

and, finally, one of the two following possibilities hold:

$$d_k = d_k^2 \tag{3.41}$$

or:

There exists

$$\hat{d}_k \in C(d_k^1, d_k^2), \tag{3.42}$$

s.t.

$$\| \hat{d}_k \| \leq \upsilon \| d_k \|, \tag{3.43}$$

and

$$f(x_k + \hat{d}_k) > f(x_k) + \theta_2 \langle g_k, \hat{d}_k \rangle. \tag{3.44}$$

Theorem 3.8 (Martínez, 87) *If* $x^* \in D$ *is a limit point of the sequence generated by the above general algorithm, then* $g(x^*) = 0$.

Proof 3.2 *If* x^* *is a limit point of the sequence* (x_k), *then there exists a subsequence* $(x_k)_{k \in K_1 \subset \mathbb{N}}$ *such that*

$$\lim_{k \in K_1} x^k = x^*$$

Define $B = \{x_k,\ k \in K_1\}$. *B is a bounded subset of* D. *Therefore, as* $f \in C^1(D)$,

$$\| g(x) \| \leq C_1 \qquad \text{for all } x \in B.$$

Now, if $g(x^*) \neq 0$, *then there exists* K_2 *an infinite subset of* K_1, *such that:*

$$\| g(x_k) \| \geq C_2 > 0 \quad \text{for all } k \in K_2.$$

Therefore,

$$\underline{M} C_2 \ \leq \ \| d_k^i \| \ \leq \ \bar{M} C_1 \quad \text{for all } k \in K_2,\ i = 1, 2.$$

Hence, there exists $K_3 \subset K_2$ *such that*

$$\lim_{k \in K_3} \ d_k^i = d_i, \quad i = 1, 2.$$

and

$$\underline{M} C_2 \ \leq \ \| d^i \| \ \leq \ \bar{M} C_1, \quad i = 1, 2.$$

Taking limits on both sides of 3.38 and 3.39 for $k \in K_3$, *gives:*

$$\langle d_i, g(x^*) \rangle \leq -\theta_1 \| d_i \| \ \| g(x^*) \|$$

and

$$\underline{M} \| g(x^*) \| \ \leq \ \| d_i \| \ \leq \ \bar{M} \| g(x^*) \|, \quad i = 1, 2.$$

Now consider two possibilities:

(a) *There exists* $\alpha > 0$ *such that* $\| d_k \| \geq \alpha \| g_k \|$ *for all* $k \in K_3$.

(b) *The opposite to (a).*

(a) *In case (a),*

$$\alpha C_2 \leq \alpha \| g(x_k) \| \ \leq \ \| d_k \| \ \leq \ \| d_k^2 \| \ \leq \ \bar{M} \| g(x_k) \| \ \leq \ \bar{M} C_1$$

for $k \in K_3$.

Therefore, there exists $K_4 \subset K_3$, *such that*

$$\lim_{k \in K_4} \quad d_k = d \neq 0.$$

Now, by equation (3.37), d_k *is in the positive cone determined by* d_k^1 *and* d_k^2. *A straight forward calculation shows that the inequality (3.38) holds for all the members of this cone. So,*

$$\langle d_k, g(x_k) \rangle \;\leq\; -\theta_1 \parallel d_k \parallel \;\; \parallel g(x_k) \parallel \;\; = \;\; -\gamma < 0$$

Taking limits on both sides of this inequality for $k \subset K_4$, *results in:*

$$\begin{aligned} f(x^* + d) &= \lim_{k \in K_4} f(x_k + d_k) \leq \lim_{k \in K_4} f(x_k) + \theta_2 \langle d_k, g(x_k) \rangle \\ &= f(x^*) + \theta_2 \langle d, g(x^*) \rangle \leq f(x^*) - \theta_1 \theta_2 \parallel d \parallel \;\; \parallel g(x^*) \parallel \\ &= f(x^*) - \theta_2 \gamma. \end{aligned}$$

Thus, there exists $k_o \in \mathbb{N}$ *such that, if* $k \geq k_o$, $k \in K_4$, *then*

$$f(x_{k+1}) = f(x_{k+d_k}) \leq f(x^*) - \theta_2 \frac{\gamma}{2}$$

which is a contradiction.

(b) *In case (b),* $d_k \neq d_k^2$ *and* $\lim \; d_k = 0$ *for all* $k \in K_5$, *an infinite subset of* K_3. *Therefore,*

$$\lim_{k \in K_5} \quad \hat{d}_k = 0,$$

and

$$f(x_k + \hat{d}_k) > f(x_k) + \theta_2 \langle \hat{d}_k, g_k \rangle.$$

Thus for $k \in K_5$,

$$\langle \hat{d}_k, g(x_k + \zeta_k \hat{d}_k) \rangle > \theta_2 \langle \hat{d}_k, g_k \rangle, \qquad 0 \leq \zeta_k \leq 1$$

and so

$$\langle \frac{\hat{d}_k}{\| \hat{d}_k \|}, g(x_k + \zeta_k \hat{d}_k) \rangle > \theta_2 \langle \frac{\hat{d}_k}{\| \hat{d}_k \|}, g_k \rangle. \tag{3.45}$$

Now let K_6 be an infinite subset of K_5 such that $\lim_{k \in K_6} \frac{\hat{d}_k}{\|\hat{d}_k\|} = 0$.

Taking limits on both sides of equation (3.45) for $k \in K_6$, gives:

$$\langle g(x^*), v \rangle \geq \theta_2 \langle g(x^*), v \rangle.$$

But $\hat{d}_k$ belongs to $C(d_k^1, d_k^2)$, so $\langle \hat{d}_k / \| \hat{d}_k \|, g(x_k) \rangle \leq -\theta_1 \| g(x_k) \|$, thus $\langle v, g(x^*) \rangle < 0$. This implies that $\theta_2 > 1$, which is a contradiction.

Thus the proof is complete. •

Corollary 3.1 *Let $\epsilon > 0$. If $\{x : f(x) \leq f(x^o)\}$ is compact, then there exists $k \in \mathbb{N}$ such that $\| g(x_k) \| \leq \epsilon$.*

The 'topological' properties of the above general algorithm may be completed with the following two results, which are stated here without proof.

Lemma 3.1 *Let x^* be a strict local minimum of f in D, $\epsilon > 0$. Then, there exists a neighborhood V of x^* such that $x_k \in B(x^*, \epsilon)$ for all $k \geq k_o$, provided $x^{k_o} \in V$.*

Theorem 3.9 (Martínez, 87) *If x^* is a strict local minimum of f in D, then there exists $\epsilon > 0$ such that $\lim x_k = x^*$, whenever $x^o \in B(x^*, \epsilon)$.*

Martínez [47] showed that his algorithm shares the same properties as Gauss-Newton and related methods in the case of full-rank and low-residual, however, his algorithm needs to solve the subproblem several times. This makes the algorithm in some cases inefficient. Martínez and Santos [48] suggest a new Curvilinear search

direction which prevents the algorithm from solving the subproblem more than once to provide the new direction d_k.

The proposed method has new approach similar to the work of Martínez and Santos. Like in the previous algorithm, after the calculation of an approximated Gauss-Newton direction d, the next iterate on a two-dimensional subspace which includes d is determined. We simplify the process of searching the new point by defining the plane using a scaled gradient direction. The global convergence properties are presented at the end.

3.8 A New Algorithm with Curvilinear Search Direction

In this section we present the proposed algorithm which does not need to solve the subproblem in step 5 of algorithm (3.2) several times.

Algorithm 3.4 *Let* $F : \Omega \subset \mathbb{R}^n \to \mathbb{R}^m, m \geq n, F \in C^1(\Omega), \Omega$ *an open set. Let* $x^0 \in \Omega$ *be an arbitrary initial point,* $\eta \in [0,1), \theta_1, \theta_2 \in [0,1), \theta_3 \in [0, \frac{1}{2}), \overline{M} > 0, \underline{M} \in [\theta_1 \overline{M}, \overline{M}]$.

Let x_k *be the k-th approximation to the solution. We denote* $F_k = F(x_k)$, $J_k = J(x_k)$, $g_k = J_k^T F_k = \nabla(\frac{1}{2} \parallel F(x) \parallel^2) \mid_{x_k}$, $D_k = diag(\sigma_k^1, \ldots, \sigma_k^n), where$

$$\sigma_k^i = \begin{cases} (x_k^i)^2 & if \quad (x_k^i)^2 \in [\underline{M}, \overline{M}], \\ \underline{M} & if \quad (x_k^i)^2 < \underline{M}, \\ \overline{M} & if \quad (x_k^i)^2 > \underline{M}. \end{cases}$$

Step 1: *Compute* J_k *and* g_k. *If* $g_k = 0$, *stop.*

Step 2: *Obtain* $w_k \in \mathbb{R}^n$ *such that*

$$\parallel J_k^T J_k w_k + g_k \parallel \leq \eta_k \parallel g_k \parallel \tag{3.46}$$

Step 3: *Obtain* $v_k \in \mathbb{R}^n$ *as the solution of the following bi-dimensional problem:*

$$Minimize \quad \parallel J_k v + F_k \parallel$$

$$s.t. \quad v = \lambda_1 g_k + \lambda_2 w_k, \quad \parallel v \parallel \leq \parallel w_k \parallel .$$

Step 4: *Set* $d_k^1 = -g_k$. *Test the following two conditions for* v_k*:*

$$\langle v_k, g_k \rangle \leq -\theta_1 \| v_k \| \| g_k \| \tag{3.47}$$

and

$$\underline{M} \| g_k \| \leq \| v_k \| \leq \bar{M} \| g_k \| . \tag{3.48}$$

If (3.47) and (3.48) are satisfied, set $d_k^2 = v_k$ *otherwise set* $d_k^2 = d_k^1$

Step 5: *Set* $t = 1$ *perform steps (5.a) to (5.d)*

(5.a) *Set*

$$d = d(t) = t^2 d_k^2 + \frac{g_k^\top d_k^2}{g_k^\top d_k^1} t(1 - t^2) d_k^1 \tag{3.49}$$

(5.b) *If*

$$\frac{1}{2} \| F(x_k + d) \|^2 \leq \frac{1}{2} \| F(x_k) \|^2 + \theta_2 \langle g_k, d \rangle \tag{3.50}$$

go to step (5.d).

(5.c) *Let* $\hat{t}$ *be such that*

$$\theta_3 \| d(t) \| \leq \| d(\hat{t}\,) \| \leq (1 - \theta_3) \| d(t) \| \tag{3.51}$$

Replace t *by* $\hat{t}$*, Go to step 5a.*

(5.d) $d_k = d$, $x_{k+1} = x_k + d_k$.

The algorithm (3.4) has similar steps as the one introduced by Martínez [47]. In step (5.a), we are using a new curvilinear search algorithm which is slightly different from [48].

Theorem 3.10 *The algorithm 3.4 is well-defined.*

Proof 3.3 *Our proof is similar to Santos* et al. *[48] and corrects the errors in the original paper. We prove that, if $g_k \neq 0$, we can reach step (5.d) in a finite number of iterations. In step 2 of the algorithm, we solve a system of linear equations. Assuming that the Jacobian of the expanded Lagrangian function has full rank, the system of linear equations always has a unique solution. In step 3, we solve a two-dimensional subproblem and v is in the positive cone determined by g_k and w_k. Step 4 does not create any problems. Finally, we verify step 5. Let us write*

$$d = d(t) = t^2 d_2 + at(1 - t^2)d_1 \tag{3.52}$$

where $a = \frac{g_k^\top d_k^2}{g_k^\top d_k^1}$. By definition of d_1^k and (3.47), we know that $a > 0$. By (3.51), we only need to prove that (3.50) is satisfied when t is small enough.

In fact, by the Mean Value Theorem,

$$\frac{1}{2} \parallel F(x^k + d(t)) \parallel^2 - \frac{1}{2} \parallel F(x^k) \parallel^2 = g(x^k + \xi(t)d(t))^\top d(t) \tag{3.53}$$

where $g(x)$ denotes $\nabla(\frac{1}{2} \parallel F(x) \parallel^2)$ and $0 \leq \xi(t) \leq 1$.

On the other hand, $d(t)$ is a positive combination of d_1 and d_2 in (3.49) and $g_k^\top d_k^1 < 0$ and $g_k^\top d_k^2 < 0$. Therefore $g_k^\top d(t) < 0$ for $t \in [0, 1]$.

So, by (3.53)

$$lim_{t \to 0} \frac{\frac{1}{2} \parallel F(x_k + d(t)) \parallel^2 - \frac{1}{2} \parallel F(x_k) \parallel^2}{g_k^\top d(t)} = \frac{g(x_k + \xi(t)d(t))^\top d(t)}{g_k^\top d(t)} = \frac{g(x_k)^\top d_k^1}{g(x_k)^\top d_k^1} \tag{3.54}$$

Taking limits on both sides of (3.54), we have

$$\frac{\frac{1}{2} \parallel F(x_k + d_1) \parallel^2 - \frac{1}{2} \parallel F(x_k) \parallel^2}{g_k^\top d_k^1} = 1 \tag{3.55}$$

Therefore, given $\theta_2 \in (0, 1)$, there exists $\hat{t} > 0$ such that

$$\frac{\frac{1}{2} \parallel F(x_k + d(\hat{t})) \parallel^2 - \frac{1}{2} \parallel F(x_k) \parallel^2}{g_k^\top d(\hat{t})} \geq \theta_2 \tag{3.56}$$

for $t \in (0, \hat{t})$. Thus, using $g_k^T d(\hat{t}) < 0$, we obtain (3.50).
This completes the proof.

Theorem 3.11 *Assume that (x_k) is generated by algorithm (3.4) Then:*

(a) *If there exists $c > 0$ such that $\| g_k \| \leq c$ for all $k = 0, 1, 2, \ldots$ and $x^* \in \Omega$ is a limit point of (x_k), then $J(x^*)^T F(x^*) = 0$.*

(b) *Let $\epsilon > 0$. If $x \in \Omega : \| F(x) \|^2 \leq \| F(x^0) \|^2$ is compact, then there exists $k \in \mathbb{N}$ such that $\| J(x_k)^T F(x_k) \| \leq \epsilon$.*

(c) *Let x^* be a strict local minimizer of f in $\Omega, \epsilon > 0$. Then, there exists $\epsilon_1 > 0$ such that $\| x_k - x^* \| \leq \epsilon_1$.*

(d) *If x^* is a strict local minimizer of $\| F(x) \|^2$ and an isolated stationary point in Ω, then there exists $\epsilon > 0$ such that $\lim x_k = x^*$, whenever $\| x^0 - x^* \| \leq \epsilon$.*

Proof 3.4 *Algorithm (3.4) is a particular case of a slight extension of Algorithm (3.3). In fact, Santos* et al. *[48] proved the theorem by changing the inequality $\| d_k \| \leq \| d_k^2 \|$ in (3.51) to $\| d_k \| \leq K \| d_k^2 \|$ for some constant K. It is clear that the equations (3.47) and (3.49) and the definition of d_k^1 implies that $d_k \in C(d_k^1, d_k^2)$. We can also verify that d_k^1 satisfies:*

$$g_k^T d_k^1 \leq -\theta_1 \| d_k^1 \| \quad \| g_k \| \tag{3.57}$$

In fact,

$$\frac{|g_k^T d_k^1|}{\| g_k \| \quad \| d_k^1 \|} = \frac{|g_k^T D_k g_k|}{\| g_k \| \quad \| D_k g_k \|} \geq \frac{\underline{M} \| g_k \|^2}{\overline{M} \| g_k \| \quad \| g_k \|} = \underline{M}/\overline{M} \geq \theta_1. \tag{3.58}$$

So, (3.57) is proved.

Now, by (3.47),(3.48) and the choice of d_k^2 we have:

$$g_k^T d_k^2 \leq -\theta_1 \| d_k^2 \| \quad \| g_k \| . \tag{3.59}$$

Hence, the axiom (3.38) of (3.3) is satisfied.

By the definition of d_k^1, *we have*

$$\underline{M} \parallel g_k \parallel \ \leq \ \parallel d_k^1 \parallel \ \leq \ \overline{M} \parallel g_k \parallel . \tag{3.60}$$

Hence, by (3.47),(3.48), the axiom (3.39) of (3.3) is also satisfied. Also, by (3.50), the axiom (3.40) of (3.3) holds. Finally, we prove the inequality

$$\parallel d_k \parallel \ \leq K \parallel d_k^2 \parallel \tag{3.61}$$

From the expression (3.52) for $d(t)$ *we have,*

$$d'(t) = 2td_k^2 + a(1-3t^2)d_k^1$$

Therefore, if $\gamma(t) = \parallel d(t) \parallel^2$, *we have, for* $t \in [0,1]$,

$$\begin{aligned}
\gamma'(t) &= 2d'(t)^\top d(t) = 2(2td_k^2 + a(1-3t^2)d_k^1)^\top (t^2 d_k^2 + at(1-t^2)d_k^1) \\
&= 2[2t^3 {d_k^2}^\top d_k^2 + 2at^2(1-t^2){d_k^1}^\top d_k^2 + at^2(1-3t^2){d_k^1}^\top d_k^2 + a^2t(1-3t^2)(1-t^2){d_k^1}^\top d_k^1] \\
&= 4t^3 \parallel d_k^2 \parallel^2 + [4at^2(1-t^2) + 2at^2(1-3t^2)]{d_k^1}^\top d_k^2 + 2a^2t(1-3t^2)(1-t^2) \parallel d_k^1 \parallel^2 \\
&\leq 4 \parallel d_k^2 \parallel^2 + 6a{d_k^1}^\top d_k^2 + 2a^2 \parallel d_k^1 \parallel^2 \\
&\leq 4\overline{M}^2 \parallel g_k \parallel^2 \ + \frac{6\|g_k\| \ \|d_k^2\| \ \|d_k^1\| \ \|d_k^1\|}{\theta_1 \|g_k\| \ \|d_k^1\|} + \frac{2\|g_k\| \ \|d_k^2\| \ \|d_k^1\|^2}{\theta_1 \|g_k\| \ \|d_k^1\|} \\
&\leq 4\overline{M}^2 \parallel g_k \parallel^2 + \frac{6\overline{M}^2 \|g_k\|^2}{\theta_1} + \frac{2\overline{M}^2 \|g_k\|^2}{\theta_1} \\
&\leq (4\overline{M}^2 + \frac{6\overline{M}^2}{\theta_1} + \frac{2\overline{M}^2}{\theta_1}) \parallel g_k \parallel^2 = C_1 \parallel g_k \parallel^2 .
\end{aligned} \tag{3.62}$$

Therefore, for $t \in [0,1]$,

$$\begin{aligned}
\parallel d(t) \parallel^2 &= \gamma(t) \leq \gamma(1) + max_{t\in[0,1]}|\gamma'(t)| \leq \parallel d_k^2 \parallel^2 + C_1 \parallel g_k \parallel^2 \leq \parallel d_k^2 \parallel^2 + \frac{C_1 \|d_k^2\|^2}{\underline{M}^2} \\
&= (1 + \frac{C_1}{\underline{M}^2}) \parallel d_k^2 \parallel^2
\end{aligned} \tag{3.63}$$

Thus, (3.61) is satisfied with $K = \sqrt{1 + \frac{C_1}{\underline{M}^2}}$ *and the proof is complete.*

Now, by (3.47), (3.48) and the choice of d_k^2 we also have:

$$g_k^\top d_k^2 \leq -\theta_1 \| d_k^2 \| \| g_k \| .$$

So, the axiom of (3.38) of (3.3) is satisfied.

3.9 Motivation

In this section, we explain the motivation of using curvilinear search direction similar to one introduced in [48]. We know that $d(t)$ lies in the positive cone generated by d_k^1 and d_k^2 for all $t \in [0,1]$. On the other hand, we want the direction to have a negative gradient when the step is infinitesimal. We assume the search direction is tangent to d^1 for small step t, (i.e. $d_k'(0) = ad_k^1$). It is also assumed that $d_k(1) = d_k^2, d_k(0) = 0, a > 0$. Let h be the orthogonal projection of d_k^2 on the orthogonal complement of the line generated by d_k^1, related to the norm

$$\| \cdot \|_{D_k^{-1}} \ (\| z \|_{D_k^{-1}}^2 = z^\top D_k^{-1} z \text{ for all } z \in R^n).$$

Therefore,

$$h = d_k^2 - \frac{d_k^{2\top} D_k^{-1} d_k^1}{d_k^{1\top} D_k^{-1} d_k^1} d_k^1.$$

But $d_k^1 = -D_k g_k$, hence,

$$h = d_k^2 - \frac{g_k^\top d_k^2}{g_k^\top d_k^1} d_k^1.$$

Each point z in the plane spanned by d_k^1, h may be expressed as $z = y_1 d_k^1 + y_2 h$. d_k^2 corresponds to $y_1 = \frac{g_k^\top d_k^2}{g_k^\top d_k^1}, y_2 = 1$.

A simple curve considered by Martínez [48] has the following form,

$$P = \{z = y_1 d_k^1 + y_2 h \mid y_2 = (\frac{g_k^\top d_k^1}{g_k^\top d_k^2} y_1)^2\},$$

and the curve used in algorithm 3.4 has the following form in the coordinate (y_1, y_2)

$$P = \{z = y_1 d_k^1 + y_2 h \mid y_1 = \frac{g_k^\top d_k^2}{g_k^\top d_k^1}(-y_2^{\frac{3}{2}} + y_2 + \sqrt{y_2})\}.$$

3.9.1 The Implementation

In this section, we explain some of the details of our implementation for the trust region Algorithm 3.4. The bi-dimensional subproblem

$$\text{Minimize} \quad \| J_k \, d + F_k \|$$

$$\text{s.t} \quad \pi = \lambda_1 \pi_1 + \lambda_2 \pi_2, \quad \| d \| \leq \Delta$$

arising in step 3 of the algorithm (3.4) is solved as follows, where Δ defines the trust region radius.

Consider the usual case when π_1 and π_2 are linearly independent. Let $\{e_1, e_2\}$ be an orthonormal basis of the subspace spanned by $\{\pi_1, \pi_2\}$, $E = (e_1, e_2)$. The above problem can be written as:

$$\begin{aligned} &\text{Minimize} \quad \| J_k \, \pi + F_k \| \\ &\text{s.t.} \quad \pi = E\lambda, \quad \| \pi \| \leq \Delta \end{aligned} \tag{3.64}$$

since

$$E\lambda = [e_1, e_2] \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

Equivalently, the above problem can be written as

$$\begin{aligned} &\text{Minimize} && \| J_k\, E\lambda + F_k \| \\ &\text{s.t.} && \| E\lambda \| \;\leq\; \Delta \quad (\equiv\; \|\lambda\| \;\leq\; \Delta) \end{aligned}$$

(since E is orthonormal)

Minimizing the above system is also the same as minimizing

$$\begin{aligned} \|J_k E\lambda + F_k\|^2 &\equiv (J_k E\lambda + F_k)^\top (J_k E\lambda + F_k) \\ &\equiv \lambda^\top E^\top J_k^\top J_k E\lambda \;+\; 2(\lambda^\top E^\top J_k^\top F_k) \;+\; F_k^\top F_k \\ &\equiv \lambda^\top B\lambda \;+\; 2\lambda^\top C \;+\; D \\ \text{Subject to} \qquad & \|\lambda\| \leq \Delta. \end{aligned}$$

where:

$B = (E^\top J_k^\top J_k E)$	is a 2×2 symmetric positive definite matrix since, E is a $n \times 2$ matrix and $J^\top J$ is a $n \times n$ matrix, and J is a full rank matrix.
$C = (E^\top J_k^\top F_k)$	is a vector of length 2. Since, $J^\top$ is a $n \times n$ matrix and F_k is a n vector.
$D = F_k^\top F_k$	is a scalar constant.

Hence, the minimization problem (3.64) can be considered as a simple quadratic problem with a quadratic constraint:

$$\begin{aligned} &\text{Minimize} && \lambda^\top B\lambda \;+\; 2\lambda^\top C \;+\; D \\ &\text{s.t.} && \|\lambda\| \;\leq\; \Delta \end{aligned} \tag{3.65}$$

This problem can be solved in a number of ways. Appendix B describes the method that we used. There are , however, some better ways to solve this sub-problem. Williamson [78] for example, studied this case to find all local and global minima where B is undefined. By global minimizer, we mean a point in the feasible set where the objective function takes on the absolutely lowest value. Clearly, all global minimizers are also local minimizers. This approach also breaks down the analysis into several different cases based primarily on the eigen-decomposition of the (2×2) Hessian B. It is shown that in the degenerate case, all of the global solutions to problem (3.65) can be determined analytically. This fact is strongly dependent on the restriction to two dimensions and is the primary reason for the assertion that finding all of the local solutions to problem (3.65) is computationally inexpensive. In our case, the matrix B is positive definite and hence the local minimum is also the global minimum.

3.10 Summary

In this chapter, we have studied a method solving constrained optimization problems based on first order necessary conditions. We have explained that it is possible to remove the singularity from the penalty based algorithms by expanding the equations. We have shown that these expanded equations can be used to solve general nonlinear optimization problems. The special case of bound constraints has been treated so that we would not need to increase the size of the problem. A new bi-dimensional trust region with a curvilinear line search has been introduced and convergence properties have been studied. Details of the implementation for the solution of bi-dimensional trust region have been shown at the end.

Chapter 4

The Implementation of the Proposed Method

4.1 Introduction

This chapter is completely devoted to a detailed discussion of the works that have been completed and discussion of the difficulties arising for solving the systems of nonlinear equations in Expanded Lagrangian Systems. We test the proposed method on some small test problems and compare the number of the function evaluation of the proposed method with the other methods which use an Expanded Lagrangian Systems. The second example involves a real benchmark problem from water resources. This application involves a nonlinear objective function but only linear constraints, however, we will consider a large-scale problem with non-linear objective function and nonlinear constraints in chapter 6.

4.2 Proposed Method (PM)

Having described the necessary material, such as Expanded Lagrangian Function and the proposed trust region algorithm, we are now able to synthesize together and introduce the proposed method (PM). From now on, we assume that we have a feasible starting point which is at least inside the feasible region for inequality constraints. However, there are different alternatives to find a feasible point either inside or at least close to the feasible region. This is crucial because of the term $\ln(g_j(x))$ in the penalized objective function. One strategy is to use an unconstrained optimization technique to minimize the loss function

$$[(g(x) - 2r_0e)^-]^\top[(g(x) - 2r_0e)^-] \tag{4.1}$$

where $(g(x) - 2r_0e)^- = \{(g_1(x) - 2r_0)^-, \ldots, (g_p(x) - 2r_0)^-\}$ and $(g_i(x) - 2r_0)^- = \min\{(g_i(x) - 2r_0), 0\}, (i = 1, \ldots, p)$ to produce a point close to or in the interior of the feasible region for the inequality constraints. If the approximate solution of this problem is not feasible or is feasible but very close to the boundary of the feasible region, then a barrier function shift can be introduced in the following way: Choose $\epsilon > 0$ and define δ_i by

$$\delta_i = \begin{cases} 0 & if \quad g_i(\overline{x}) > \epsilon \\ \epsilon & if \quad 0 < g_i(\overline{x}) \leq \epsilon \\ \epsilon - g_i(\overline{x}) & if \quad g_i(\overline{x}) \leq 0 \end{cases} \tag{4.2}$$

where $\overline{x}$ is the point obtained from the minimization of the loss function. Now we have $g_i(\overline{x}) + \delta_i \geq \epsilon > 0$, $(i = 1, \ldots, p)$, and we use the shifted penalty function

$$P(x, r) = f(x) + h(x)^\top h(x)/(2r) - r\sum_{i=1}^{p}(\ln(g_i(x) + r\delta_i/r_0) \tag{4.3}$$

which is well-defined at $r = r_0$ and $x = x^0$. Having obtained an approximate answer x^0 from the unconstrained optimization technique, we add a homotopy to

the equation $\nabla L = 0$ in order to account for a moderate value of ∇P. Thus given x^0, we switch to the following expanded system:

$$\begin{cases} \nabla L - (r/r_0)\nabla P(x^0; r_0) &= 0 \\ h + r\lambda &= 0 \\ M(g + r\delta/r_0) - re &= 0, \end{cases} \tag{4.4}$$

where $L = f - h^\top \lambda - g^\top \mu$, $M = diag(\mu)$, $\mu = (\mu_1, \ldots, \mu_p)^\top$, and $\lambda^0 = h(x^0)/r_0$ and $\mu^0 = r_0/g(x^0)$. We define the quantity $\beta_0 = (1 + |\lambda|_2^2 + |\mu|_2^2)^{1/2}$ at each step of the iteration, if this quantity becomes large, say of the order 10^3 or larger, we switch to the following system:

$$F(z, \bar{r}) = \begin{cases} \nabla L - (\mu_0^2 \bar{r}/r_0)\nabla P(x^0, r_0) &= 0 \\ h + \bar{r}\lambda &= 0 \\ M(g + \mu_0 \bar{r}\delta/r_0) - \bar{r}\mu_0^2 e &= 0 \\ \mu_0^2 + |\mu|_2^2 + |\lambda|_2^2 - \beta_0^2 &= 0 \end{cases} \tag{4.5}$$

where $L = \mu_0 f - h^\top \lambda - g^\top \mu$, $M = diag(\mu)$, $\mu = (\mu_1, \ldots, \mu_p)^\top$, $\bar{r} = r/\mu_0$. At $(x, \lambda, \mu; r) = (x^0, \lambda^0, \mu^0; r_0)$, where $(x^0, \lambda^0, \mu^0; r_0)$ is the solution obtained from the first switch to the system (4.4), we have a solution of the system (4.5) given by $(x, \lambda, \mu, \mu_0; \bar{r}) = (x^0, \lambda^0, \mu^0, 1; r_0)$, and where β_0 is defined and fixed by $\beta_0 = (1 + |\lambda^0|_2^2 + |\mu^0|_2^2)^{1/2}$

4.3 Trust Region Method

The set of parameterized nonlinear equations (4.5) explained can be solved in a number of ways. A Continuation method is used by Poore and Al-Hassan [58] to keep track of the path as r tends to the prescribed limit. Lundberg [42] used

the same approach to solve some highly nonlinear test problems and his results indicate that continuation methodology performs reasonably well comparing with some current methods. However, for the test problems from the book of Hock and Schittkowski [30](results described next) it seems that the continuation method does not seem to be competitive and this motivated us to use a trust region based algorithm as the method is well known to be a reasonable alternative to solving nonlinear equations.

4.4 Test Results

We compare the performance of three algorithms PENCON, LOSCON and TR-ELF that solve the expanded Lagrangian systems with some other relevant methods in Table (4.2). The implementation of PENCON uses the quadratic penalty-log barrier function. Since an initial point satisfying strict inequality in the inequality constraint $g \geq 0$ is required for the log barrier function, Poore and Al-Hassan [60, 42] first use the loss function $(g^-)^\top g^-$to generate a point $\hat{x}$ at which $g(\hat{x}) > 0$ or is at least close to the feasible region $\{x : g(x) \geq 0\}$ and then introduce a shift δ so that $g(\hat{x}) + \delta > 0$. Then a quasi-Newton method with a BFGS update is used to minimize the penalty function $P(x, r) = f(x) + h^\top(x)h(x)/(2r) - r\sum \ln(g_i(x) + r\delta_i/r^0)$ at some value of the penalty parameter, say r_0, in which the problem is reasonably well conditioned. A quadratic-cubic line search and an Armijo stopping criterion [14], modified to maintain feasibility ($g(x) + \delta > 0$), have been used to globalize the quasi-Newton method. Once the minimization problem is solved, continuation techniques are used to track the solution to optimality at $r = 0$. The initial value of $r_0 = .1$ has been used in the numerical experiments reported in Table (4.2) under the heading PENCON where scaling has not been used. Additional information

can be found in [58].

The implementation that uses the quadratic penalty-loss function is called LOSCON. Since the loss function is an exterior penalty function, an initial feasible point for the inequality constraints is not required. At the end of the unconstrained optimization phase, expanded systems (4.4) or (4.5) is used and only the active inequality constraints as equality constraints are treated thereafter. The implementation of TR-ELF uses quadratic penalty-loss function to find a feasible solution close to the boundary or inside the feasible region and then it switches to expanded Lagrangian function (4.5) and uses the trust region algorithm (3.4) to solve the system of nonlinear equations arising in the Expanded Lagrangian System. The two dimensional subproblem arises in step three of algorithm (3.4) is solved based on the method in appendix (B). Several standard test problems from the book by Hock and Schittkowski [30] are solved using a variety of codes in addition to PENCON, LOSCON and TR-ELF. These codes, along with the corresponding authors and methods, are given in Table (4.1).

A comparative summary of the number of function evaluations for various codes and problem sets is presented in Table (4.2). The function evaluation counts for the codes other than PENCON and LOSCON are taken from [41]. Consistent with those function evaluation counts, they count the evaluation of a p-dimensional vector as p function evaluations; however, they do not count upper and lower bounds on variables since they are handled directly in the code and the gradient evaluations of linear constraints are counted only once. In particular, the n-dimensional gradient of a scalar function is counted as n function evaluations. The approximation of the Hessian of the Lagrangian in the continuation phase is based on finite differences [14].

The bottom row of Table (4.2) shows a ranking obtained by assigning one to

Table 4.1: The Summary of Codes Compared

Code #	Code Name	Author	Method
I	TR-ELF	Sadjadi & Ponnambalam	Expanded-Trust Region
II	PENCON	Al-Hassan, Poore	Penalty-continuation
III	LOSCON	Lundberg *et al.*	Loss function-continuation
IV	VF02AD	Powell	Quadratic approximation
V	OPRQP	Bartholomew-Biggs	Quadratic approximation
VI	GRGA	Abadie	Generalized reduced gradient
VII	VF01A	Fletcher	Multiplier
IIX	FUNMIN	Kraft	Multiplier
IX	FMIN	Kraft, Lootsma	Penalty

Table 4.2: The Summary of Function Evaluation

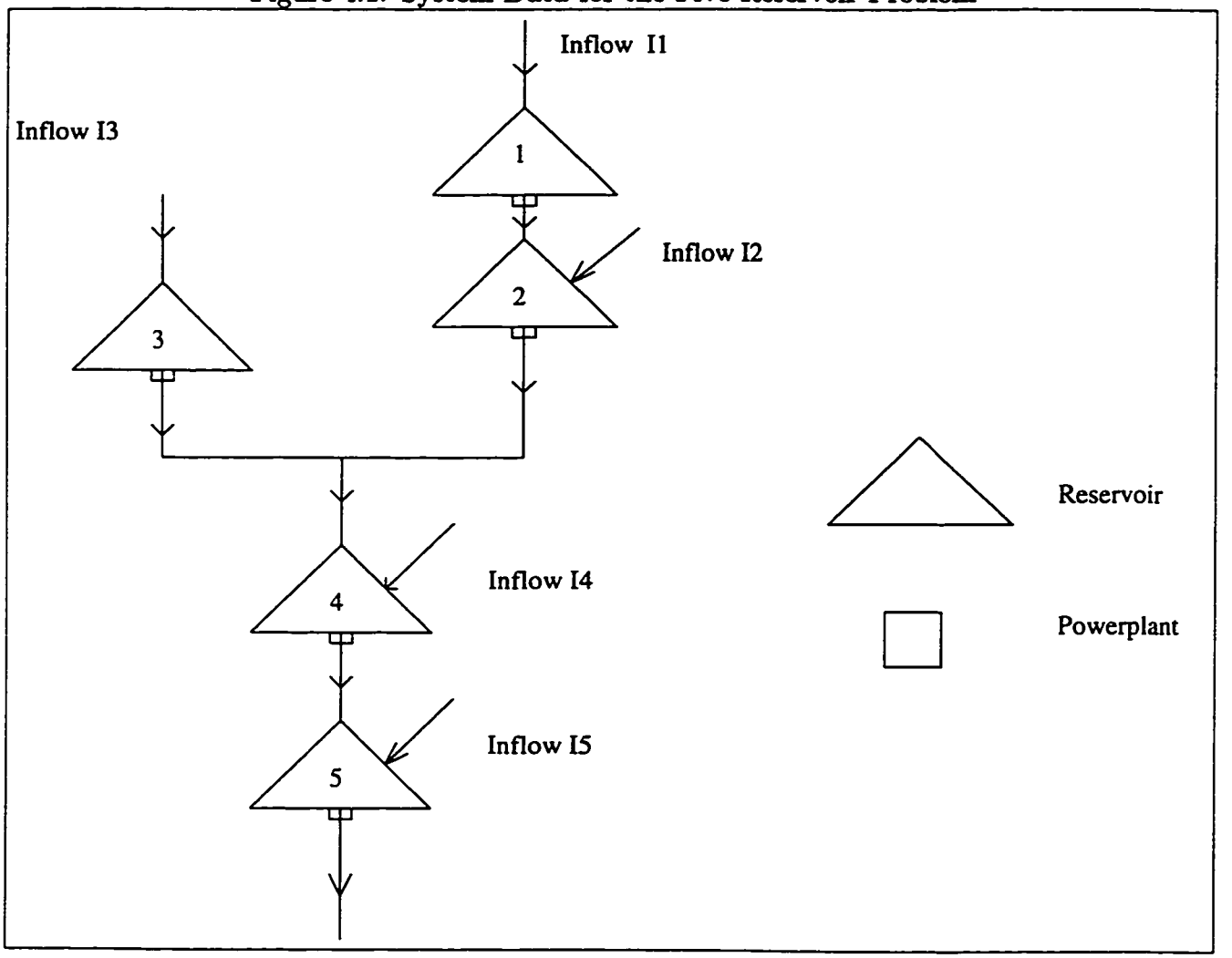
Prob.	I	II	III	IV	V	VI	VII	IIX	IX
H5	65	4	23	24	23	86	48	46	234
H10	49	124	97	50	128	678	282	556	689
H12	125	210	99	72	184	277	442	578	362
H14	137	156	140	44	149	108	226	774	887
H26	411	-	541	152	210	444	1560	2276	622
H27	1016	-	292	200	346	1714	435	960	556
H29	192	375	310	104	206	646	421	482	414
H34	1017	427	321	75	291	346	619	***	2483
H39	164	-	805	195	411	2557	1110	1638	2295
H40	514	-	526	120	300	2099	800	1724	2912
H43	219	509	590	240	220	1580	448	1032	1949
H50	830	161	-	144	160	190	248	432	720
H55	173	211	-	14	84	***	***	2737	4767
H65	211	148	209	44	3942	249	308	2520	374
H78	249	189	873	72	204	578	416	912	1304
H100	510	1563	-	200	400	854	944	6500	3362
Ave.	3.56	4.17	3.69	1.31	3.125	6.27	5.875	7	7.31

Ave.: The Average Rank I.The Proposed Method *** Indicates a Failure

the lowest function evaluation count, two to next, etc., for each problem, and then averaging the ranks over all problems. In case of a failure, the failed codes are all given the ranks plus one of the successful code having the largest function evaluation count. Table (4.2) summarizes the number of function evaluation counts for inequality and equality constrained problems. As we can verify, TR-ELF performs the best among the first three codes which are penalty based algorithms. The code VF02AD of Powell comes first in comparison with all different codes and the reason is quite clear. These problems are relatively simple and many are very close to quadratic programming problems. It is also clear that these test problems are all small and they cannot demonstrate the performance of the proposed method (PM) for general large-scale nonlinear programming. In order to be consistent with the theme of this research, the numerical test problem should be fairly representative of large-scale hydro-power system in which the problem is characterized by high dimensionality, nonlinearity and non-convexity.

The numerical problem used by Hiew [29] has five reservoirs with nonlinear and non-convex objective function that maximizes energy production. This problem appears to be most suitable as a benchmark or test problem. It has a simplistic but representative layout (comprising reservoirs in series and parallel) and a nonlinear, non-convex objective function. The number of reservoirs is not too large but is adequate for the purposes of algorithm evaluations. Figure (4.1) demonstrates the layout of hydro-power generation problem and the connections between all five reservoirs.

Figure 4.1: System Data for the Five-Reservoir Problem



4.5 General Description of The Five-Reservoir Problem

The five-reservoir problem has a tree-like layout configuration. Physical characteristics of this system are summarized in Table (4.3) and the inflows to this system are given in Table (4.4). The objective function is the maximization of total energy generation over an operation time span of 12 months at 15-day discretization. The mathematical definition of the objective function is given by :

$$\max \sum_{t=1}^{t=23} \sum_{i=1}^{i=5} \alpha_i O_i^t (1 - e^{-\beta_i S_i^{t+1}}) \tag{4.6}$$

where S_i^t and O_i^t are the storages and the outflows of reservoir i at time period t and α and β are numerical constants.

Table 4.3: System Data for the Five-Reservoir Problem

Description	Reservoir Number				
	1	2	3	4	5
Upper bound on storage	50.0	50.0	80.0	100.0	120.0
Lower bound on storage	5.0	5.0	8.0	10.0	15.0
Upper bound on release	15.0	17.0	20.0	25.0	30.0
Lower bound on release	0.0	0.0	0.0	0.0	0.0
Constant α	1.5	1.5	1.0	1.1	1.2
Constant β ($\times 100$)	.025	.02	.02	.05	.01

The above objective function is the exact formula used by Hiew [29] and it is not clear why 23 periods (as against the expected value of 24) were used with the first summation. The steady-state equations of the five reservoirs are as follows:

Table 4.4: Inflow Data for the Five-Reservoir Problem

Month	Reservoir Number				
	1	2	3	4	5
1	0.0	0.0	5.0	5.0	0.0
2	2.0	2.0	10.0	5.0	5.0
3	2.0	2.0	10.0	5.0	5.0
4	2.0	2.0	10.0	10.0	5.0
5	2.0	2.0	20.0	15.0	10.0
6	2.0	2.0	25.0	20.0	10.0
7	5.0	5.0	5.0	25.0	10.0
8	7.0	7.0	5.0	25.0	15.0
9	7.0	7.0	5.0	20.0	15.0
10	10.0	10.0	5.0	10.0	15.0
11	10.0	10.0	5.0	10.0	15.0
12	10.0	10.0	5.0	15.0	15.0
13	7.0	7.0	5.0	5.0	20.0
14	7.0	7.0	0.0	5.0	20.0
15	6.0	6.0	5.0	5.0	10.0
16	6.0	6.0	5.0	5.0	10.0
17	6.0	6.0	10.0	5.0	10.0
18	4.0	4.0	10.0	5.0	5.0
19	4.0	4.0	5.0	1.0	5.0
20	4.0	4.0	5.0	2.0	2.0
21	2.0	2.0	5.0	5.0	1.0
22	2.0	2.0	5.0	5.0	1.0
23	1.0	1.0	5.0	0.0	1.0

$$\left.\begin{aligned} S_1^{t+1} &= S_1^t + I_1^t - O_1^t \\ S_2^{t+1} &= S_2^t + I_2^t + O_1^t - O_2^t \\ S_3^{t+1} &= S_3^t + I_3^t - O_3^t \\ S_4^{t+1} &= S_4^t + I_4^t + O_2^t + O_3^t - O_4^t \\ S_5^{t+1} &= S_5^t + I_5^t + O_4^t - O_5^t \end{aligned}\right\} t = 1, \ldots, 23. \tag{4.7}$$

Where I_1^t to I_1^t are inflows of reservoir one to reservoir five respectively.

The storages and outflows are bounded within a certain bounds as follows:

Minimum Release and Storage Requirement

The low flow requirement on reservoir outflows is accommodated by imposing lower bounds on the release and storage rates.

$$\left.\begin{aligned} O_i^t &\geq O_{i,Min}^t \\ S_i^t &\geq S_{i,Min}^t \end{aligned}\right\} t = 1, 2, \ldots, T \ \ and \ \ i = 1, 2, \ldots, 5.$$

Maximum Release and Storage Requirement

Local control stations have been selected for each reservoir in order to achieve flood protection in the reaches immediately downstream of the reservoirs during flood control operations.

$$\left.\begin{aligned} O_i^t &\leq O_{i,Max}^t \\ S_i^t &\leq S_{i,Max}^t \end{aligned}\right\} t = 1, 2, \ldots, T \ \ and \ \ i = 1, 2, \ldots, 5.$$

The five-reservoir hydro-power optimization problem as defined is typically nonlinear and non-convex. For such a problem, convergence to a global optimum cannot be guaranteed. A recommended practice is to repeat the solution of the same problem from different initial solutions and selecting the best of the results.

This application has been solved by PM and MINOS and the output results of both methods are summarized in Table (4.5). The numerical results state that the PM is able to solve the above application with better objective function value than MINOS. However, this application cannot represent a general nonlinear constrained optimization problem as there is no nonlinear constraints in this application. The next chapter explains a case study of water resources management problem in which the mathematical model consists of nonlinear objective function and nonlinear constraints.

Table 4.5: The Results of Optimization

Evaluation Criteria	The Proposed Method	MINOS
Objective Value	2265.935	2155.876

The Objective Function is Maximization.

4.6 Summary

In this chapter, we have presented a new approach to solve the constraint optimization problems. We have tested the performance of PM with some small test problems. The preliminary results indicate that PM can be considered as an alternative among the other methods. Next chapter, we test the performance of PM with some applications of water resources problems and compare the results with LANCELOT project.

Chapter 5

Case Studies

This chapter presents two case studies of engineering problems. A water resource problem and an investment application will be studied. In both cases the proposed method (PM) is applied. The first application is inherently very large-scale in real world and because of general nonlinear and non-convex constraints in this application we compare the output results with LANCELOT which is designed for large-scale nonlinear optimization problems. The second case study is used to illustrate the efficiency of the linear solver used in our proposed method.

5.1 Applications of Water Resources Problem

5.1.1 Introduction

The Great Lakes make up the largest fresh water lake system in the world. Over 32 million Americans and 12 million Canadians live within the boundaries of the Great Lakes-St.Lawrence River drainage basin. Fluctuations in the water levels of

these lakes affects many of these people, both directly and indirectly. High water levels are of concern to those who live along the Great Lakes shoreline. These can combine with other factors, such as storms, to cause serious flood and erosion damage. Alternatively, low water levels affect shipping, recreational boaters, and hydroelectric power generation.
We plan to consider a mathematical model of the Great Lakes Water Level Regulation Problem. The objective of the work is to use the proposed method to minimize both the deviation of water levels from given targets and outflows from target flows for a given time horizon.

5.2 Background

The Great Lakes consist of Lakes Superior, Michigan, Huron, Erie and Ontario. Figure (5.1) illustrates the connections between the lakes.

They essentially form a chain of reservoirs with each one draining into the next. Lake Superior is the uppermost lake and the largest. It drains into Lake Huron by way of the St. Mary's River. Lake Michigan, located completely in the United States, also drains to Lake Huron. The water levels of both Lake Michigan and Lake Huron are necessarily at the same elevation since both lakes are connected by the wide and deep Straits of Mackinac.

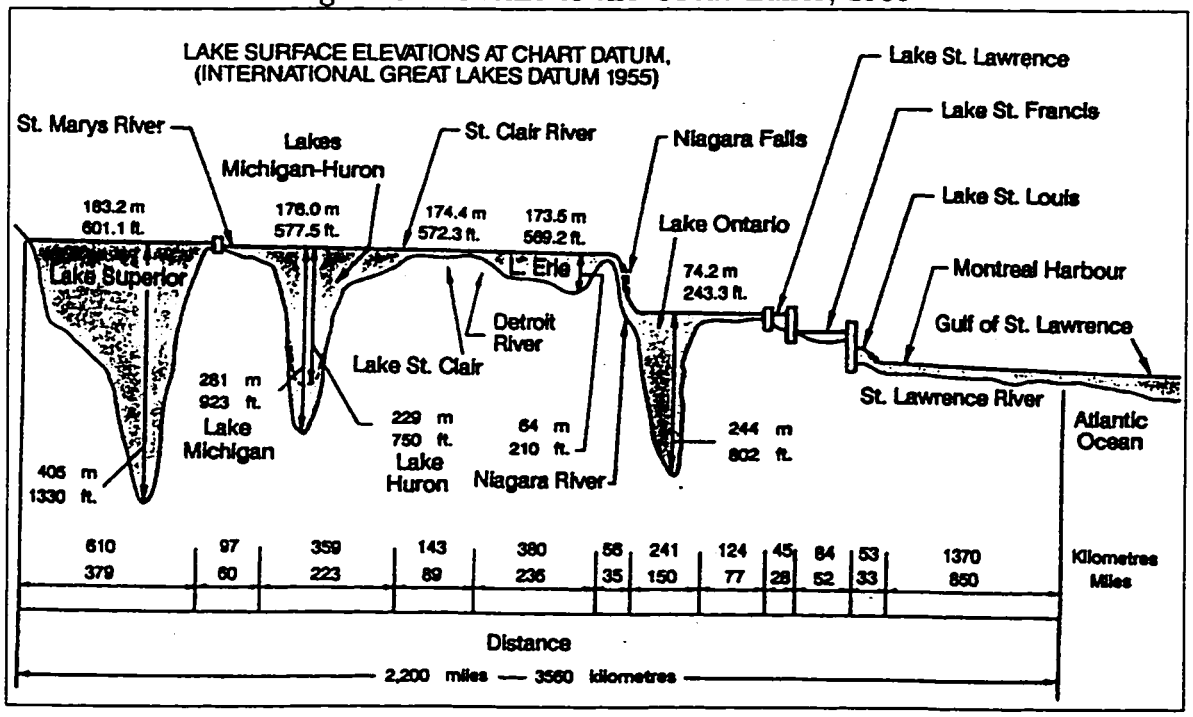
These Lakes have a water surface area of about 246,000 square kilometers. The area of the surrounding land and other smaller lakes draining into the Great Lakes is about 528,000 square kilometers. From Lake Huron, water flows to Lake Erie by way of the St. Clair River, Lake St. Clair, and the Detroit River. Lake Erie is the shallowest of the five Great Lakes and second smallest in surface area. Its outflow is mainly through the Niagara River and into Lake Ontario. Lake Ontario water in

turn flows into the St. Lawrence River which carries the total outflow of the Great Lakes some 870 kilometers to the Gulf of St. Lawrence.

The water level in each of the Great Lakes falls or rises according to the amount of water entering or leaving it. The amount entering a lake includes precipitation falling on the lake, runoff including snow-melt from the surrounding area, and ground water inflow. The water leaving a lake consists of evaporation from the lake's surface, ground water outflow, and outflow at the lake outlet. Water levels will rise when the amount entering a lake exceeds the amount leaving it. This happens every spring. The converse is true every fall and winter.

Total outflow into the St.Lawrence River depends on the water level of the middle lakes (Michigan, Huron and Erie). The higher the level of these lakes, the higher the total outflow. Low lake levels will bring low outflows. This self-regulating

Figure 5.1: Profile of the Great Lakes, 1985



feature helps to keep levels on the lakes within certain ranges. However, because of the large size of the Great Lakes and the limited discharge capacities of their outflow rivers, extremely high or low levels and flows sometimes persist for some considerable time even after factors which caused the extremes have changed.

5.3 Factors Affecting the Great Lakes Water Levels

- Natural Factors

  There are several natural parameters that affect the Great Lakes water levels including precipitation, evaporation, ice retardation, and others.

  Precipitation or lack of it is the most significant natural cause of long-term extreme fluctuations in lake levels. The intervals between high and low levels, and the durations of these highs and lows, can vary widely over a number of years.

  Evaporation has minimum effect on water level during the spring season with condensation occurring at times when the lakes are cold relative to the air. It has the maximum effect during the fall season and early winter when the lake water is warm relative to the air temperatures.

  As an ice cover forms on the connecting rivers of the Great Lakes the flow is reduced because of added surface friction and a reduction in the channel cross-section area. If the ice cover breaks, ice jams can form causing serious flooding upstream, ice damages along the shoreline, and greatly reduced flows downstream.

- Human Factors

  Human factors are believed to be the most influencing in the Great Lakes water regulation. Among these factors are Dredging, Consumptive Uses, and Regulation.

  Dredging is done primarily to maintain adequate depths in shipping lanes for navigation. Consumptive use is water that is withdrawn and not returned to the Great Lakes. The amount of consumptive use increases progressively through the Great Lakes system.

  Regulation is one of the most important among the human factors. To regulate a lake means to adjust or modify its outflows according to certain rules in order to bring about more desirable lake levels or outflows.

5.4 Effects of Water Level Fluctuations

5.4.1 Environmental Effects

Water level fluctuations are not necessarily bad. In fact, they form a process which is natural to the fish and wildlife inhabiting the Great Lakes. Extreme or extended periods of high and low water levels, however, can compound the effects of natural lake processes and cause undesirable results. Low water levels or flows, for example, can aggravate pollution problems in the Great Lakes by reducing their dilution. Shallow water environments provide important spawning and feeding grounds which are essential for the maintenance of fish stocks. High lake levels provide more favorable fish habitats whereas low levels could dry up spawning and feeding grounds. Wetlands require fluctuating lake levels to enhance their productivity. Periodic flooding is necessary to maintain a variety of plant communities at different stages

and thereby the wetland's ability to support a diversity of fish and wildlife species. Long-term lake level fluctuations, on the other hand, generally produce plants community shifts. Low water conditions generate an invasion of sedge/meadow plants by shrubs, displacement of emergent vegetation by sedge/meadow, and result in less open water and associated aquatic communities. High water conditions result in increased open water aquatic plants, as other communities decrease.

5.4.2 Hydro-Electric Power

Major hydro-electric power generation facilities are located on the Niagara, St. Lawrence and St. Mary's Rivers. These facilities use water from the rivers to generate electric power.

High lake levels bring about high river flows which increase power generation, while low flows cause reductions in power generation. Electric utilities use coal, oil, natural gas or nuclear thermal power plants to supplement the power produced by the hydro plants.

5.5 Problem Formulation

At present, regulation structures exist only on Lakes Superior and Ontario. The outflow from Lake Superior has been completely regulated, since 1921, at Sault Ste. Marie, using various control structures, seaway navigation locks and hydropower installations. Present regulation procedures call for outflows that maintain a desirable water level, with respect to long term averages, on both upstream (Lake Superior) and downstream (Lake Michigan-Huron) lakes. The present regulation

plan is known as Plan 1977-A. In keeping with the existing level of regulation for the Great Lakes system, the present study assumes full regulation only for Lakes Superior and Ontario.

The application of the Great Lakes water regulation has been studied by different people previously (See for examples [57, 23, 68]). Fletcher [23] in his stochastic model introduces a new formulation which can provide the mean and standard deviation of the optimal storages and releases over a long period of time. His model, which consists of highly nonlinear equations, is solved by the SQP method. Because of the nature of the formulation, the information of the first and the second derivatives of the equations are not available symbolically, therefore SQP estimates this information and the results may not reach a high degree of accuracy at the optimal solution. Seifi [68] uses 90 years of information on inflows as data for his stochastic formulation. The method minimizes the absolute value of deviation of outflows and releases from targets by using SLP along with IPM algorithms.

The next section introduces a new general formulation which can provide a way to determine the steady-state storages and releases for the Great Lakes water regulation problem.

5.6 Mathematical Formulation

5.6.1 Objective Function

The main objective of regulating the Great Lakes system is to keep the lake levels close to their long term averages and reduce the total range of fluctuations. The objective is specified in terms of minimizing the total deviations from the set of storage and release target values, $\hat{s}_i^t$ and $\hat{o}_i^t$, given by Environment Canada. Several

factors have been considered to set these target values, including Hydro-power, Environmental corners, Recreational boating, Commercial navigation, and Riparian or shore property [7]. In order to have a common basis to compare the effects of different regulation plans, the historical monthly information of inflows of the Great Lakes for the period 1900 to 1989 were considered. The objective function in this study is to minimize the total squared values of the monthly deviations from storage and release target values based on 90 years of information.

As figure (5.2) illustrates, the Great Lakes are connected together in a serial sequence. Let s_i^t be the Storage of Lake i at time period t and o_i^t be the Outflow of Lake i at time period t. Our purpose is to find the best storages and releases of all Lakes in order to minimize the fluctuation of the water levels and water flows from their respective target values within a time horizon of T. The objective function can be defined as:

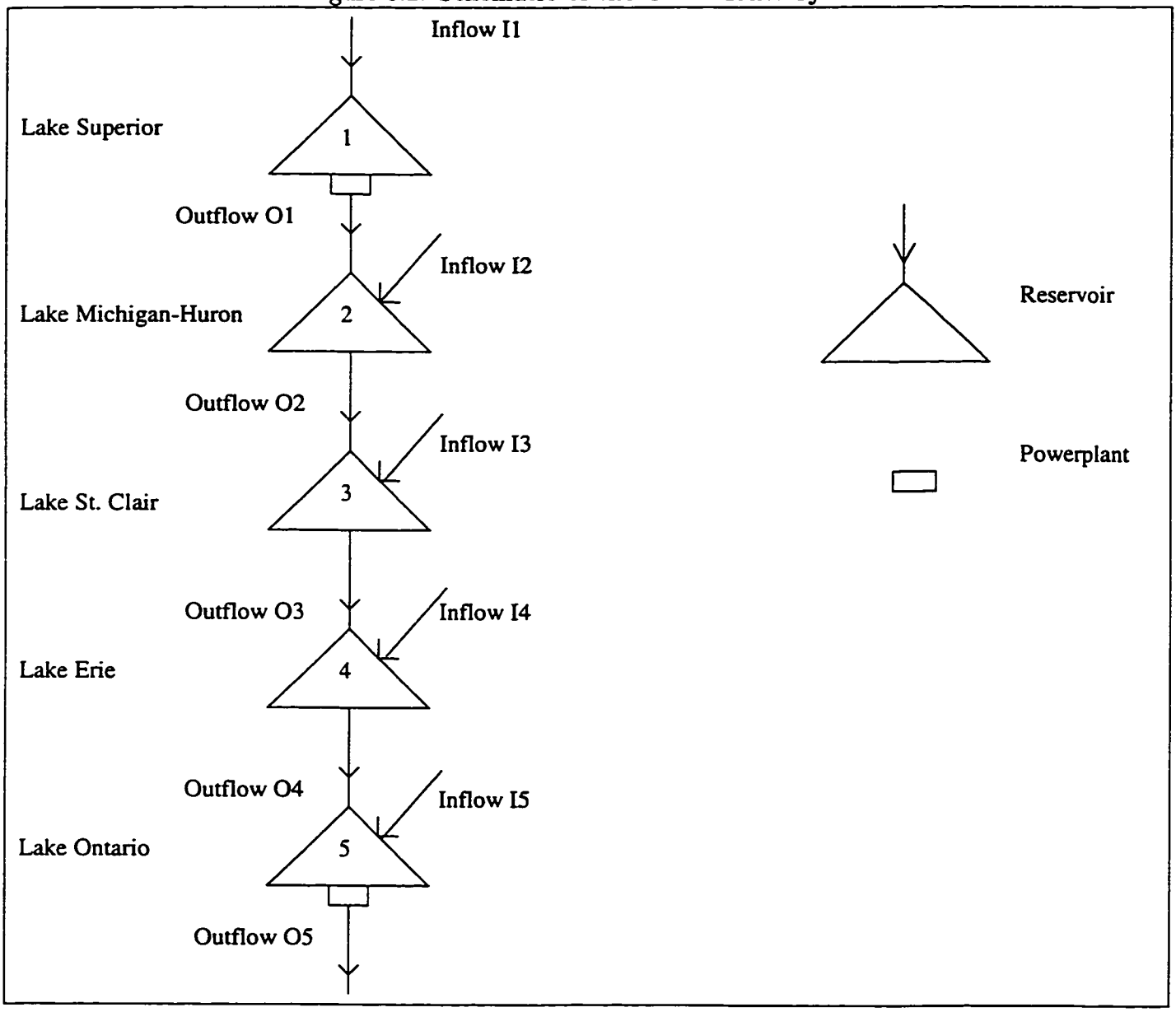
$$\min \sum_{t=1}^{T} \sum_{i=1}^{5} \left\{ (s_i^t - \hat{s_i^t})^2 + (o_i^t - \hat{o_i^t})^2 \right\} \tag{5.1}$$

where $\hat{s_i^t}$ and $\hat{o_i^t}$ are target values and are assumed to be given.

5.6.2 Linear Constraints

Let $s_1^t \ldots s_5^t$ and $o_1^t \ldots o_5^t$ be the storages and the outflows of Lakes superior, Michigan, Huron, Erie and Ontario respectively at time period t. The object is to perform the minimization over certain time period ($T = 12, 24, ..., 1080$). Therefore, the number of variables involved in the optimization problem is $10T$. Let I_i^t be the water inflow to reservoir i during time period t. The total water available in reservoir one during time period $t+1$ is equal to the total water storage s_1^t plus inflow I_1^t. Considering the total water outflow o_i^t coming out of reservoir during period t, the steady state equations for storage in that reservoir becomes $s_1^{t+1} = s_1^t + I_1^t - o_1^t$.

Figure 5.2: Schematic of the Great Lakes System



The same equations hold for the second reservoir except that the total water inflow are $I_2^t + o_1^t$ and so on.

$$\left.\begin{aligned} s_1^{t+1} &= s_1^t + I_1^t - o_1^t \\ s_2^{t+1} &= s_2^t + I_2^t + o_1^t - o_2^t \\ s_3^{t+1} &= s_3^t + I_3^t + o_2^t - o_3^t \\ s_4^{t+1} &= s_4^t + I_4^t + o_3^t - o_4^t \\ s_5^{t+1} &= s_5^t + I_5^t + o_4^t - o_5^t \end{aligned}\right\} t = 1, \ldots, T. \tag{5.2}$$

The above equations hold for a given time period $t = 1$ to $t = T$. Once $t = T$, the system of equations is reinitialized back to time period $t = 1$, i.e.

$$s_i^{T+1} = s_i^1, \quad , i = 1, \ldots 5$$

5.6.3 Nonlinear Constraints

With respect to Lakes Michigan, Huron, and Erie, the following discharge equations, representative of the existing natural channel conditions, are:

$$\left.\begin{aligned} o_2^{t+1} &= 0.0841168 \left[\frac{\frac{s_2^t}{480.8} + \frac{s_3^t}{4.6}}{2} - 543.4 \right]^2 \left(\frac{s_2^t}{480.8} - \frac{s_3^t}{4.6} \right)^{0.5} \\ o_3^{t+1} &= 0.1280849 \left(\frac{s_3^t}{4.6} - 543.4 \right)^2 \left(\frac{s_3^t}{4.6} - \frac{s_4^t}{105.15} \right)^{0.5} \\ o_4^{t+1} &= 0.2605000 \left(\frac{s_4^t}{105.15} - 550.11 \right)^{2.2} \end{aligned}\right\} t = 1, \ldots, T. \tag{5.3}$$

5.6.4 Bound Constraints

The storages and the outflows are bounded as follows:

Minimum Release and Storage Requirement

The low flow requirement on reservoir outflows is accommodated by imposing lower bounds on the release and storage rates.

$$\left.\begin{array}{ll} o_i^t & \geq O_{i,Min}^t \\ s_i^t & \geq S_{i,Min}^t \end{array}\right\} t = 1, 2, \ldots, T \ \ and \ \ i = 1, 2, \ldots, 5.$$

Maximum Release and Storage Requirement

Local control stations were selected for each reservoir in order to achieve flood protection in the reaches immediately downstream of the reservoirs during flood control operations. Maximum releases and storages are thus:

$$\left.\begin{array}{ll} o_i^t & \leq O_{i,Max}^t \\ s_i^t & \leq S_{i,Max}^t \end{array}\right\} t = 1, 2, \ldots, T \ \ and \ \ i = 1, 2, \ldots, 5.$$

5.6.5 Size of the Problem

There are five sets of equality constraints in (5.2) and three sets of nonlinear equations in (5.3). Therefore, for a total time period T, there are $8T$ equality constraints and $20T$ bound constraints. The number of unknown variables is $5T$ for outflows and $5T$ for storages, for a total of $10T$ unknown variables.

The problem is challenging for the following reasons,

1. The problem is non-convex.
2. A high accuracy solution is needed because of the scaling problems. For example, a one-foot change in the level of a large lake releases $O(10^6)$ cubic-feet/month.

3. The large number of variables possible, depending on the value of T.

5.7 Numerical Implementation for the Application

In this section we present some of the results for the implementation of the PM and compare them with the results using LANCELOT. We also tried to solve the Great Lakes problem with GAMS/MINOS, but in almost all cases MINOS cannot find a feasible solution. We do not report the output performance of GAMS/MINOS in this section.

All computations were performed in double precision, on a Sun Sparc2 workstation with FORTRAN77. The target values, $\bar{s}_i^t$ and $\bar{o}_i^t$ were used as an initial solution. Existing data on the monthly inflows of the Great Lakes system were used for a time period of up to 90 years [23]. We have also used the lower bound and upper bound as initial solution.

In order to have a fair and unbiased comparison between the performance of the PM versus LANCELOT, the exact first and the second derivatives information are used for LANCELOT. LANCELOT provides different features to solve problems under various conditions. Some of these features require access to the Harwell Subroutine Library[22]. At the time of our experimental results, we did not have access to the Harwell Library, therefore, we considered only the best options available directly through LANCELOT. Table (5.1) summarizes all of the features used for the output results. The proposed method uses the exact first derivatives and finite difference with scaling [14] to estimate the information of the second derivatives.

The systems of linear equations in LANCELOT were solved by a semi-band

Table 5.1: The Features Used in the Implementation of LANCELOT

Description	Case One Table (5.2)	Case Two Table (5.2)
The Inner-iteration Subproblem	Exact	Approximate
The Cauchy Point	Exact	Approximate
The Penalty Parameter	.1	.1
The Trust Region Norm	Infinity Norm	Two Norm
The Trust Region Radius	1.0	1.0
The Linear Equation Solver	Band=4	Band=10

Preconditioned Conjugate Gradient algorithm with band=4 and band=10. The systems of linear equations in the PM method were solved with the Preconditioned-Conjugate Gradient algorithm [28] using RCM ordering and Drop-tolerance scheme [36, 24]. Appendix (A) explains details of the implementations. Due to the existence of bound constraints in this application, it is quite possible to use the equations (3.15-3.18) to solve the linear equations as efficiently as possible. We have implemented the equation (3.7) in order to make sure that constraint qualification holds as x tends to x^*. However, we have also tested the equations (3.15-3.18) to show the impact of the smaller size of linear equations on running time. Appendix (C) contains a brief description of the algorithm used to provide $J^\top J$ in algorithm (3.2) and (3.4). Figure (5.3) shows the sparsity pattern of J_k and $J^\top J$. Figure (a) shows the nonzero elements of the Jacobian J_k when we use equations (4.5). The direct implementation of bound constraints as general inequality constraints increases the size of Jacobian significantly. Figure (b) demonstrates the sparsity pattern of $J^\top J$. The algorithm used for this multiplication needs $O(\alpha n)$ computations where α is the average number of nonzero elements in each row and n is the size of the matrix.

The results of J^TJ may or may not be sparse depending on the pattern of the matrices multiplied by each other.

We observed that the results of J^TJ (Figure(b)) remain sparse and the number of nonzero elements does not change significantly. Figure (c) shows the direct implementation of direct Choleskey factorization of J^TJ.

The algorithm which provides the multiplication of a matrix by its transpose only uses the nonzero elements in its calculation and the complexity of the computations is very cheap. However, for general purpose applications, using augmented equations is suggested as in Chen and Vannelli [9].

Figure (d) demonstrates the sparsity pattern of the Jacobian J_k by eliminating the bound constraints implicitly. Because the information of bound constraints lies on diagonal elements of Jacobian by implementing the equations (3.15-3.18), the result of $J_k^TJ_k$ becomes sparser and needs smaller space than direct implementation of bound constraints as observed on Figure(e). The direct Choleskey factorization of J_k by its transpose (Figure(f)) has obviously fewer nonzero elements.

We have used $\eta = 10^{-4}, \theta_1 = 10^{-7}, \theta_2 = 10^{-4}, \overline{M} = 10^9, \underline{M} = 10^{-5}$ for our implementation. For each case, we report:
$(T, n, q, Iter.)$: Time period, the number of variables, the number of equality constraints and the total number of iterations needed to reach the convergence criteria corresponding to the application, respectively.
($|h|_\infty$, Objective Func., CPU Time(Sec.)): The maximum violation of equality constraints, the value of the objective function, and the running time in seconds, respectively.
Tables (5.2) and (5.3) summarize the results of LANCELOT using two different features of the software. In both algorithms, the number of iterations does not

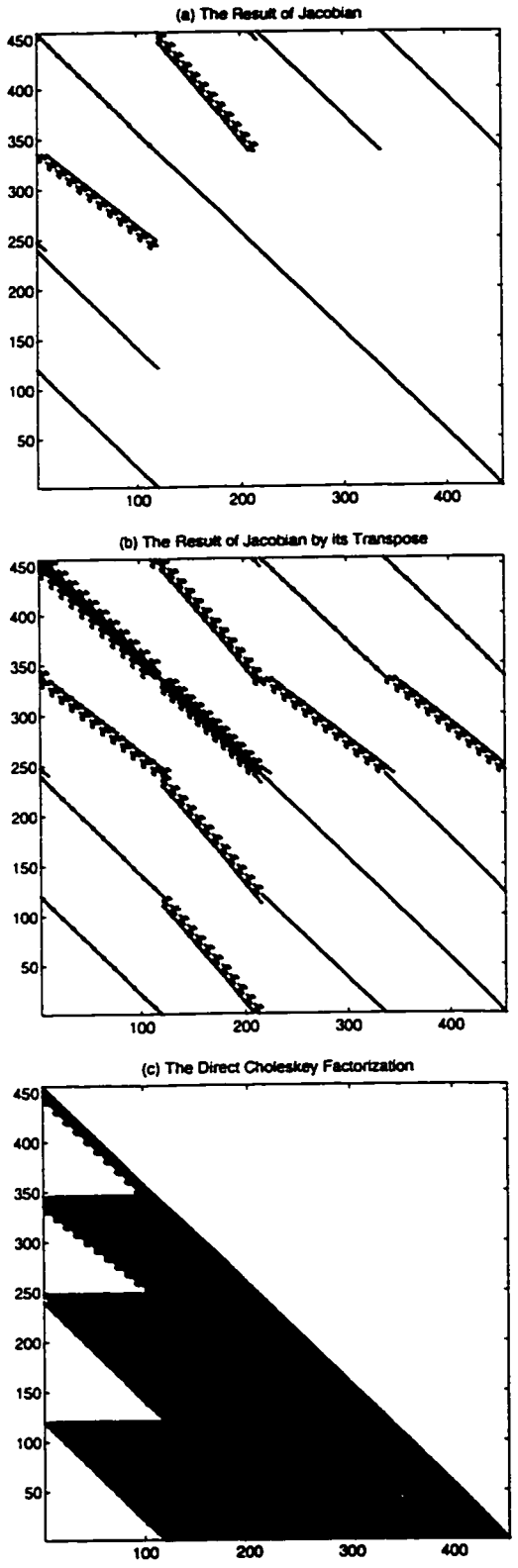


Figure 5.3: Sparsity Patterns for the Full ELF.

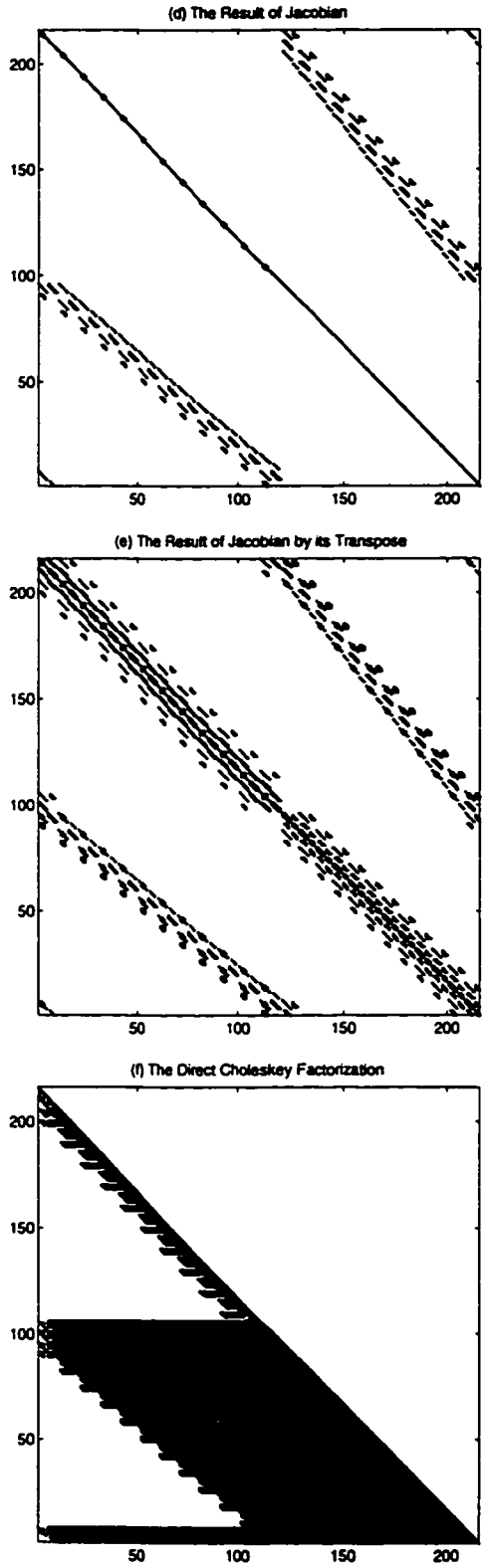


Figure 5.4: Sparsity Patterns for the Reduced ELF.

increase significantly with the size of problem.

In the output results of case one and case two shown, respectively, in tables (5.2), and (5.3) we have used the exact first and the second derivative information on the application.

Table 5.2: The Summary of LANCELOT Output Results (Case One)

T	n	q	Iter.	$\|h\|_\infty$	Objective Func.	CPU(Sec.)
12	120	96	133	5.0678D-09	1.64780402703858D+06	38.29
24	240	192	155	4.2935D-11	5.06152758271752D+06	97.98
36	360	288	139	8.3474D-09	7.62169805381109D+06	177.02
48	480	384	119	7.6016D-09	5.10989159972038D+06	248.40
60	600	480	126	2.5078D-09	3.68225450202697D+06	339.83
72	720	576	108	1.4610D-09	4.55428167037388D+06	449.51
120	1200	960	118	6.6187D-09	7.94389155258736D+06	1284.51
240	2400	1920	135	2.8412D-09	2.19258537107666D+07	3975.38
360	3600	2880	162	5.9674D-10	6.99035224883742D+07	11529.61
480	4800	3840	-	-	-	> 20000
780	7800	6240	-	-	-	> 20000
1080	10800	8640	-	-	-	> 20000

Note that different choices for each option may yield different results and this is an unfortunate reality of all methods. However, it is not always possible to try all options because, for example, if there are 6 options (see Table (5.1)) with each options taking even just two possible choices we result with a total of $2^6 = 64$ choices to try! The inner-iteration subproblem for case one (Table (5.2)) was solved exactly and the exact Cauchy point was used. The band linear equation solver was

Table 5.3: The Summary of LANCELOT Output Results (Case Two)

T	n	q	Iter.	$\|h\|_\infty$	Objective Func.	CPU(Sec.)
12	120	96	150	7.1526D-10	1.64780402953180D+06	26.57
24	240	192	151	1.1638D-10	5.06152758271459D+06	79.78
36	360	288	156	4.9823D-11	7.62169807336225D+06	128.03
48	480	384	147	6.6827D-10	5.10989161127648D+06	195.74
60	600	480	176	1.4361D-10	3.68225450258231D+06	225.58
72	720	576	152	4.2414D-10	4.55428167123117D+06	257.60
120	1200	960	128	1.4559D-10	7.94389155417452D+06	540.94
240	2400	1920	165	6.8326D-10	2.19258537116613D+07	1646.40
360	3600	2880	180	7.8691D-10	6.99035224884159D+07	2893.45
480	4800	3840	208	3.2690D-10	1.44810992658323D+08	4459.62
780	7800	6240	228	2.5818D-10	2.80880817723112D+08	8638.31
1080	10800	-	-	-	-	-

applied with band=4. We used the same exact first and the second derivative information for Case two (Table (5.3)) while the inner-iteration subproblem as well as the Cauchy point were estimated approximately. Case two seems to perform better than case one in terms of the CPU time. There does not seem to exist so much difference between the final solutions of cases one and two. However, case two, uses the approximation technique to estimate the solution of subproblems and the Cauchy point solves the problem faster than the case one. This is on the contrary to what we expect and what has been observed in the past mostly for small test problems. When both cases have solved the problems, the results of the case one and two indicate that the approximate methods may solve a large-scale problem much faster than using exact techniques and there would not be much difference between the final solutions in each case! Tables (5.4) and (5.5) summarize the performance of the PM using Martínez 's Method and the New Curvilinear Search. There does not seem to exist so much difference between the PM with the new curvilinear search and the Martínez's modified algorithm.

Table 5.4: The Summary of the PM Using Martínez 's Method

T	n	q	Iter.	$\|h\|_\infty$	Objective Function	CPU(Sec.)
12	120	96	37	1.0152D-10	1.6907769855525D+06	34.33
24	240	192	40	1.6051D-10	4.9336389910904D+06	88.52
36	360	288	43	1.6553D-10	7.4270041154615D+06	162.43
48	480	384	37	3.7778D-10	4.9604809581912D+06	199.57
60	600	480	28	4.8885D-11	3.6445097860100D+06	165.25
72	720	576	30	4.0898D-11	4.6076288095819D+06	218.17
120	1200	960	30	8.3986D-11	7.8517482607172D+06	417.11
240	2400	1920	45	6.0794D-09	2.1904356973965D+07	923.66
360	3600	2880	45	2.6613D-09	6.9515123414487D+07	3852.65
480	4800	3840	43	1.7833D-09	1.4276484327877D+08	4027.45
780	7800	6240	56	1.2790D-09	2.7703483527993D+08	10319.35
1080	10800	8640	55	1.9947D-09	3.4458742946506D+08	18322.80

Table 5.5: The Summary of the PM Using the New Curvilinear Search

T	n	q	Iter.	$\|h\|_\infty$	Objective Function	CPU(Sec.)
12	120	96	41	1.0152D-10	1.6907769855525D+06	38.98
24	240	192	39	1.7857D-10	4.9336389910904D+06	85.49
36	360	288	35	5.7943D-10	7.4270041154615D+06	123.06
48	480	384	37	9.1802D-11	4.9604809581912D+06	200.16
60	600	480	28	4.88853E-11	3.6445097860100D+06	163.08
72	720	576	30	4.08988E-11	4.6076288095819D+06	215.10
120	1200	960	30	8.3986D-11	7.8517482607172D+06	416.38
240	2400	1920	29	6.07940E-11	2.1904356973965D+07	923.42
360	3600	2880	45	2.6613D-09	6.9515123414487D+07	3852.65
480	4800	3840	58	4.5452D-09	1.4276484327877D+08	7567.43
780	7800	6240	58	4.5429D-09	2.7703483527993D+08	10443.95
1080	10800	8640	53	1.0093D-09	3.4458742946506D+08	19845.60

We also applied the linear equation solver with band=10 in case one in order to see whether providing a better preconditioner can improve the results. The results are summarized in table (5.6) and as we can see that not only there is no improvement on CPU time but some of the problems originally solved with Band=4 could not be solved. This may be due to ill-conditioning of the dense preconditioner.

Table 5.6: The Summary of LANCELOT Output Results (Case One & Band=10)

T	n	q	Iter.	$\|h\|_\infty$	Objective Func.	CPU(Sec.)
12	120	96	226	2.4760D-10	1.64780402941704D+06	81.09
24	240	192	176	9.4758D-11	5.06152758271241D+06	253.13
36	360	288	172	6.2688D-10	7.62169807334174D+06	424.92
48	480	384	180	5.9067D-11	5.10989161121162D+06	630.19
60	600	480	165	3.5412D-10	3.68225450263722D+06	1090.94
72	720	576	137	5.8894D-10	4.55428167124033D+06	959.24
120	1200	960	157	1.1516D-10	7.94389155418159D+06	2345.08
240	2400	1920	-	-	-	>7200
360	3600	2880	171	6.0901D-10	6.99035224884034D+07	23096.87

The proposed method (Table (5.5)) seems to perform reasonably well, especially for larger sizes of test problems. The main reason for the PM performing somewhat better than LANCELOT could be the iterative solver used in the PM. In our opinion, choosing an efficient preconditioner may have a dramatic effect on running time when using LANCELOT and in future these may be implemented.

Table (5.7) summarizes the results of the PM when we only use equations (3.15-3.18) in our implementation. The results show that although the use of equations

(3.15-3.18) reduces the size of the linear system solved significantly, the CPU time does not reduce correspondingly. This is due to the increase in the number of iterations. We compared the difference between the final objective function values

Table 5.7: The Summary of the Proposed Method Using (3.15-3.18)

T	n	q	Iter.	$\|h\|_\infty$	Objective Function	CPU(Sec.)
12	120	96	32	1.1835D-10	1.6907769855525D+06	26.38
24	240	192	38	1.7800D-10	4.9336389910904D+06	81.18
36	360	288	50	1.6419D-10	7.4270041154615D+06	161.69
48	480	384	38	1.0331D-08	4.9604809581912D+06	162.20
60	600	480	36	4.5162D-11	3.6445097860100D+06	194.37
72	720	576	35	5.5536D-11	4.6076288095819D+06	237.21
120	1200	960	37	1.5112D-10	7.8517482607172D+06	429.80
240	2400	1920	40	1.6719D-09	2.1904356973965D+07	964.27
360	3600	2880	45	8.62259E-08	6.9515123414487D+07	1845.33
480	4800	3840	64	3.01857E-07	1.4276484181949D+08	3578.69
780	7800	6240	79	1.29112E-07	2.7703483491521D+08	7937.04
1080	10800	8640	60	8.55041E-08	3.4458737468797D+08	8014.03

of both LANCELOT and the PM. The PM provides a better final solution in terms of the lower values on the objective functions compared to LANCELOT in most cases. Table (5.8) summarizes the output results of LANCELOT and the PM.

Table 5.8: The Summary of LANCELOT Output Results Versus the PM

T	n	LANCELOT[a] CPU(sec.)	LANCELOT[b] CPU(sec.)	Proposed Method(PM)[c] CPU(sec.)	Difference [d]
12	120	38.29	26.57	38.98	2.61 %
24	240	97.98	79.78	85.49	-2.53%
36	360	177.02	128.03	123.06	-2.55%
48	480	248.40	195.74	200.16	-2.92%
60	600	339.83	225.58	163.08	-1.03%
72	720	449.51	257.60	215.10	1.17%
120	1200	1284.51	540.94	416.38	-1.16%
240	2400	3975.38	1646.40	923.42	-0.1%
360	3600	11529.61	2893.45	3852.65	-0.56%
480	4800	-	4459.62	7567.43	-1.41%
780	7800	-	8638.31	10443.95	-1.37%
1080	10800	-	-	19845.60	-

[a] Case One.

[b] Case Two.

[c] The Proposed Method (PM).

[d] The Difference Between the Objective Function of LANCELOT and the Proposed Method (PM) in Percent (i.e. $\frac{PM-LANCLEOT}{LANCELOT} \times 100$);
Negative numbers indicate better results for the PM.

Tables (5.9) and (5.10) present results for the PM, using different initial solutions and various time periods.

Table 5.9: The Summary of the PM Using Lower Bound Values as Initial Solution

T	n	q	Iter.	$\|h\|_\infty$	Objective Func.	CPU(Sec.)
12	120	96	51	1.12795E-08	1.6907769805211D+06	40.22
24	240	192	44	2.01567E-07	4.9336386538520D+06	92.04
36	360	288	51	8.72546E-11	7.4270041157663D+06	134.26
48	480	384	50	2.53336E-07	4.9604158344990D+06	199.74
60	600	480	30	5.42911E-10	3.6445097862928D+06	149.31
72	720	576	30	2.09462E-09	4.6076288086282D+06	176.65
120	1200	960	35	7.66953E-10	7.8517482603868D+06	354.95
240	2400	1920	33	2.76344E-08	2.1904356983034D+07	715.90
360	3600	2880	45	2.52750E-07	6.9515137064238D+07	1494.25
480	4800	3840	85	1.10144E-06	1.4276483269624D+08	2746.08
780	7800	6240	85	8.35798E-06	2.7703486826814D+08	6513.32
1080	10800	8640	60	5.30675E-07	3.4458743269553D+08	6190.02

Table 5.10: The Summary of the PM Using Upper Bound Values as Initial Solution

T	n	q	Iter.	$\|h\|_\infty$	Objective Func.	CPU(Sec.)
12	120	96	29	1.71868E-08	1.6907769773364D+06	40.36
24	240	192	29	2.01085E-07	4.9336386537935D+06	54.27
36	360	288	49	1.01360E-08	7.4270040902692D+06	116.22
48	480	384	42	1.48196E-08	4.9604158295162D+06	156.79
60	600	480	28	5.06813E-09	3.6445097839904D+06	135.51
72	720	576	32	2.43659E-09	4.6076288082882D+06	131.49
120	1200	960	35	4.06146E-11	7.8517482607455D+06	269.07
240	2400	1920	33	2.73114E-08	2.1904356983217D+07	585.36
360	3600	2880	51	2.32730E-07	6.9515137063238D+07	1484.25
480	4800	3840	50	1.21906E-06	1.4276483163087D+08	3173.14
780	7800	6240	79	1.87226E-06	2.7703482931493D+08	9028.64
1080	10800	8640	60	4.28139E-07	3.4458740123190D+08	8865.68

Table (5.11) summarizes the effect of initial solutions on the PM.

Table 5.11: The Summary of the PM Using Different Initial Solutions.

		Target Value		Lower Bound		Upper Bound	
T	n	Obj. Fnc.	CPU	Obj. Fnc.	CPU	Obj. Fnc.	CPU
12	120	1.69D+6	38.98	1.69D+6	40.22	1.69D+6	40.36
24	240	4.93D+6	85.49	4.93D+6	92.04	4.93D+6	54.27
36	360	7.42D+6	123.06	7.42D+6	134.26	7.42D+6	116.22
48	480	4.96D+6	200.16	4.96D+6	199.74	4.96D+6	156.79
60	600	3.64D+6	163.08	3.64D+6	149.31	3.64D+6	135.51
72	720	4.60D+6	215.10	4.60D+6	176.65	4.60D+6	131.49
120	1200	7.85D+6	416.38	7.85D+6	354.95	7.85D+6	269.07
240	2400	2.19D+7	923.42	2.19D+7	715.90	2.19D+7	585.36
360	3600	6.95D+7	3852.65	6.95D+7	1494.25	6.95D+7	1484.25
480	4800	1.42D+8	7567.43	1.42D+8	2746.08	1.42D+8	3173.14
780	7800	2.77D+8	10443.95	2.77D+8	2746.08	2.77D+8	3173.14
1080	10800	3.45D+8	19845.60	3.45D+8	6190.02	3.45D+8	6853.26

5.8 Short Term Optimization, (Discussion)

To determine the optimal amount of water to be stored and released over a long period of time, we employ a rolling horizon procedure. Having used the PM, a Great Lakes water levels' regulation scenario is established using a short term optimization procedure based on the derived optimal releases and storages from Lake Superior to Lake Ontario for the historical information of up to 90 years. The short term procedure tries to find the optimal outflows and storages in every month by considering the information of the next twelve months each time. As a result, an optimization problem with 120 variables and 96 constraints has to be solved for each month of the entire 90 years. In other words, we start to solve the Great Lakes problem to get the optimal releases and storages for the month of January, 1900 based on the information of the whole year of 1900. Having obtained the optimal policy of S_1^* to S_5^* and O_1^* to O_5^* for the month of January, the time horizon is rolled from January to February to find the optimum releases and outflows for the month of February by considering the period of February, 1900 to February, 1901, and so on. Figure (5.5) compares the results of using the PM in a direct optimization versus the short term optimization technique. The results indicate that in the long term, both methods can reduce the fluctuation of water significantly. The short term optimization technique seems to perform better in terms of having smaller values for the objective function. Figure (5.5) also shows that, in long term, short term optimization scheme can minimize the fluctuation of water significantly.

In summary, the results indicate that the approach based on the formulation developed in this research can provide the capability of deriving satisfactory estimates of the Great Lakes levels' statistics, based on the preferences relating to lake levels for the interest group considered.

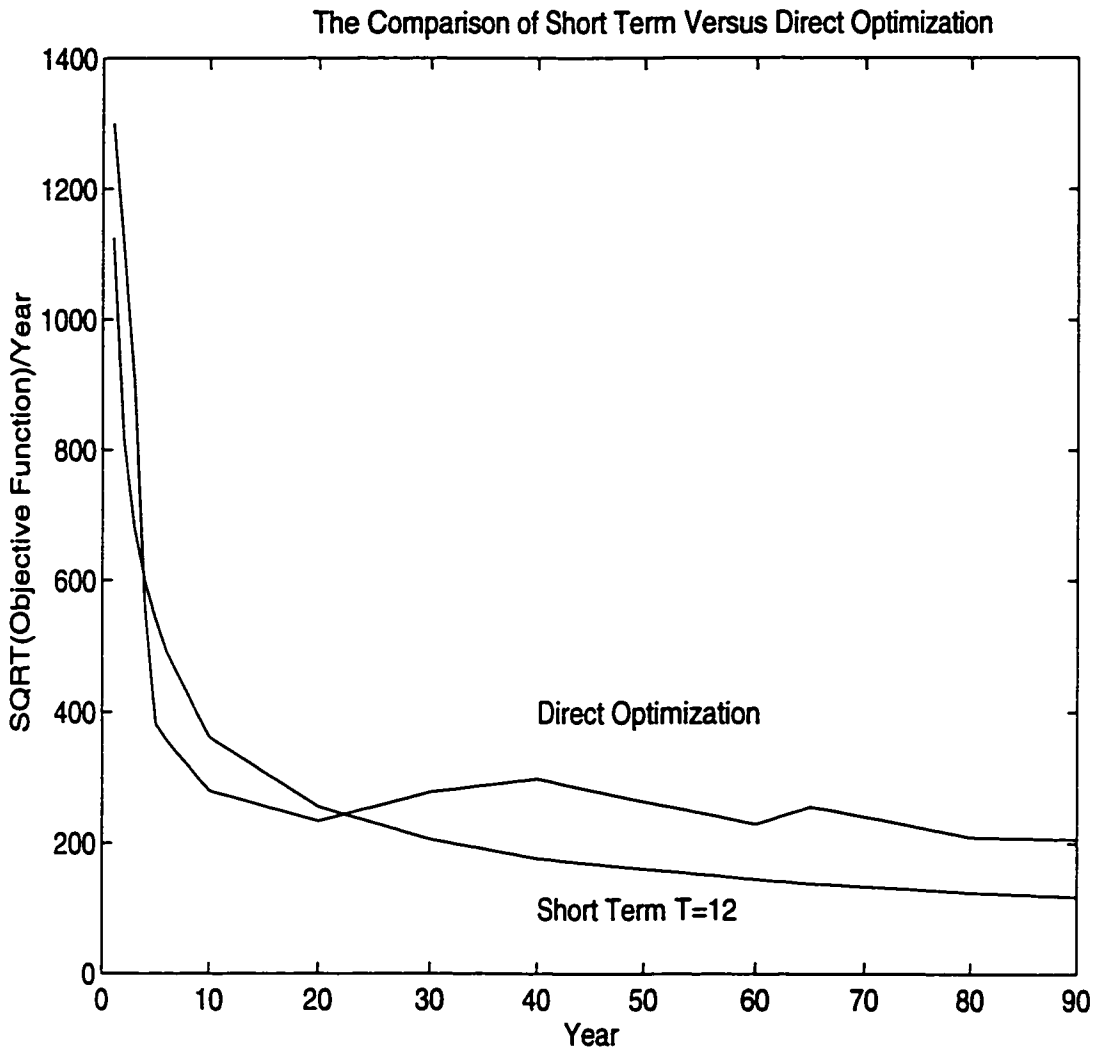


Figure 5.5: The Performance of Short Term Optimization With Direct Optimization.

5.9 The Comparison of the Optimal Policy from PM Versus Past Operations

We have compared the optimum values of the objective function (5.1) obtained from the PM and compared the results with what was implemented in practice from the year of 1900 to 1973 [7]. These compartive results are shown in Figures (5.6), (5.7) and (5.8). Figure (5.6) demonstrates the performance of the PM versus Past Operations for the available data. As the figure indicates, the optimal policy could potentially improve performance as shown in Figure (5.7). Lastly, Figure (5.8) compares average objective function values of past operations with the optimal policy from the PM. Once again it is clear that the policy from the PM is about 40% better than the results of past operations in the long run, although in our case we have had the benefit of hindsight!

Figure 5.6: The Performance of Optimal Policy Versus Past Operations

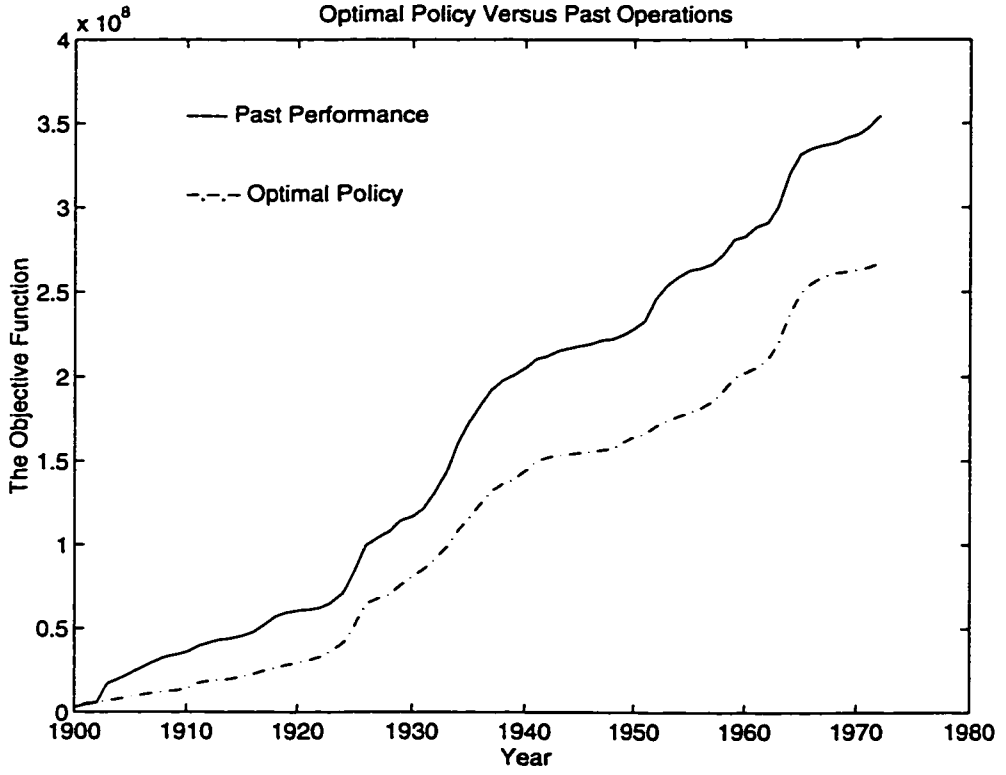


Figure 5.7: The Comparative Performance of the Optimal Policy

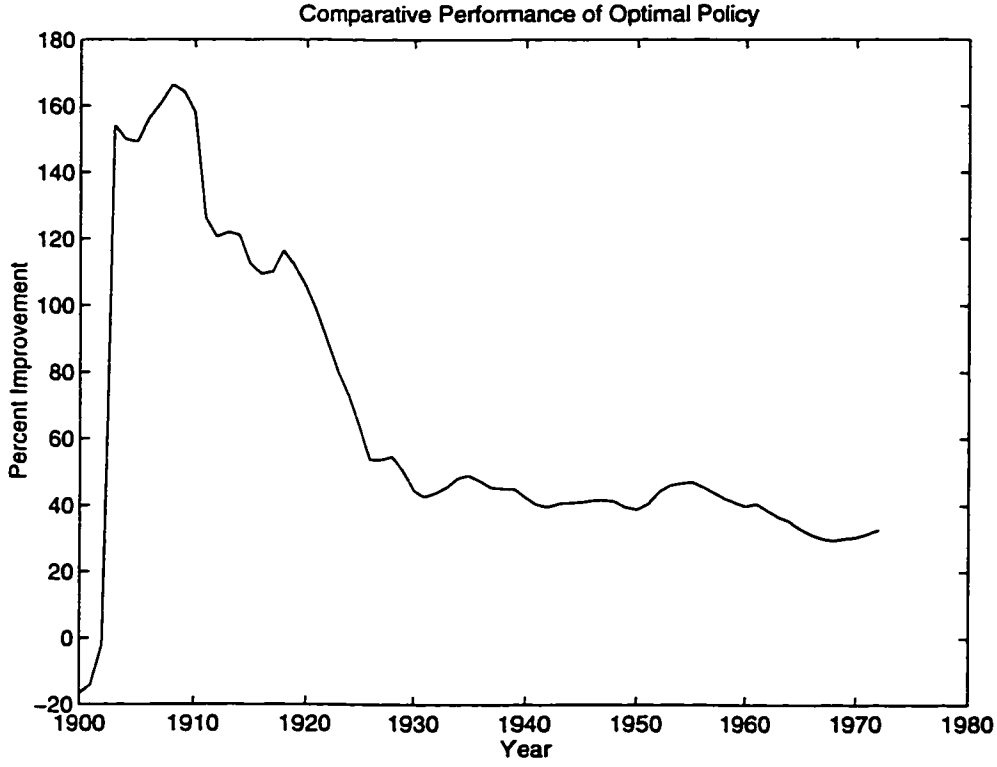
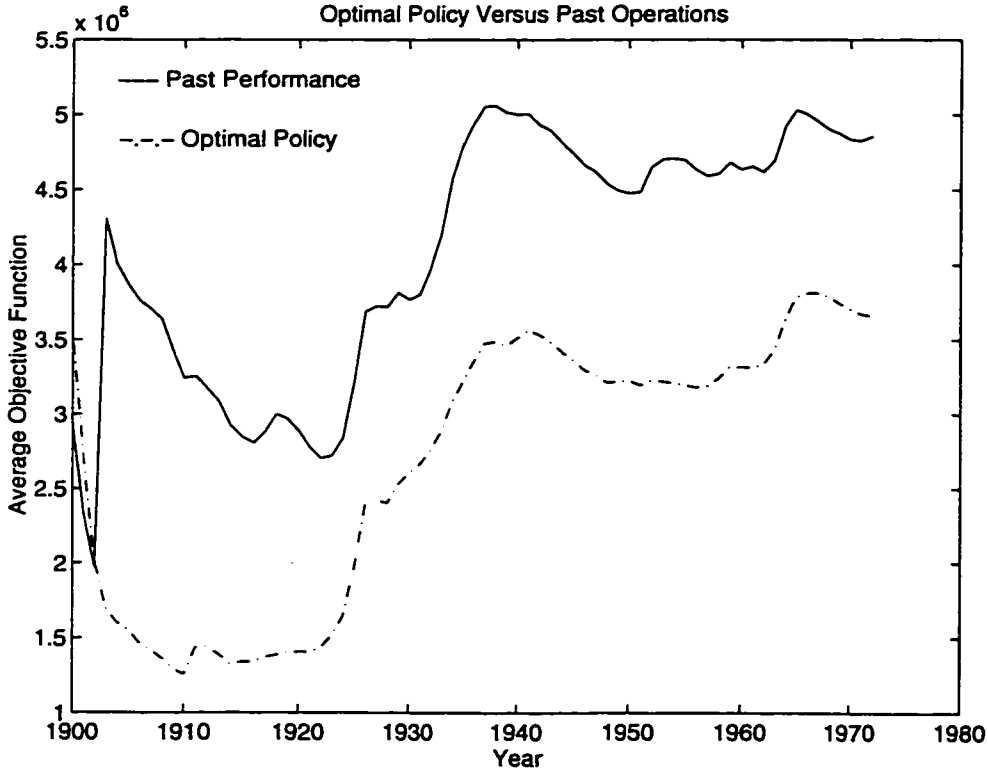


Figure 5.8: The Performance of the Optimal Policy Versus Past Operations



The next section explains the other example of optimization from economics which involves the minimization of a quadratic objective function and linear inequality and equality constrains.

5.10 An Investment Application

5.10.1 Introduction

In this section, an application from the field of financial engineering is presented. The main objective is to provide a practical and efficient method to solve some of the problems arising in this field. These applications can be generally formulated in quadratic form. This class of problem has already been well studied by others [77]. The main concern of this section is to show that the proposed method can also be applied to these kinds of problems and that the linear solver used is efficient.

The following section is organized as follows. We first introduce the parameters which are involved in the context of an investment. We then introduce the mathematical model of a typical portfolio problem. Finally, we explain the approach used to solve the problem.

5.10.2 Investment Definition

An investment can be defined in different ways and it may involve many aspects. It may involve putting money into bonds, common stocks, real estate, mortgages and oil ventures. It may involve speculating in bull markets or selling short in bear markets. It may involve puts and calls, rights, warrants, convertibles, financial futures, money market funds, commercial paper, Treasury bills and notes, gold, silver, commodities, or mutual funds. And, it may result in accumulation of wealth or dissipation of resources.

5.10.3 The Investment Process and Risk

Individual investors and professional money managers must decide on their investment objectives and then select from possible investment alternatives. The investment challenge involves certain fundamental choices. As part of controlling risks, decisions must be made on the selection of complementary assets so that a portfolio will provide a mix with an attractive rate of return balanced against acceptable risk.

Another criteria that determines an investment success is the time horizon. The choice of short-term, intermediate, or long-term investments can result in an advantageous return or lock the investor into a no-win situation. After a suitable amount of time, there should be a performance assessment. Investment decision making in all its phases is continuous.

5.10.4 Expectation

People with different ages may have different expectations from their investment. For example, a newly married couple, a middle-aged professional, and a retiree have different expectations for the return on their investment. These differences may also exist among institutional investors, pension funds, mutual funds, insurance companies, etc. Even people (or institutions) in similar circumstances may be adventurous or conservative investors. Age, lifestyle, and personality help determine the choice of an investment. Studies on the investment psychology of individuals indicate that most people are risk averse. At the same time about 50 percent of them, by owning only one or two stocks, have portfolios that are risky because they are un-diversified.

Throughout the next section we formulate a mathematical model in order to maximize these expectations and minimize the risk involved in investment.

5.11 Efficient Portfolios

Let us assume we want to invest in n different assets. Here assets may involve in any kinds of investment such as stocks, bonds, etc, but we assume that there is no short-selling on the market for these assets. Let c_i denote the expected return on asset i, $i = 1, ..., n$ and σ_{ij} denote the covariance between assets i and j, $j \leq i, j \leq n$. Let

$$c = (c_1, \ldots, c_n)^\top \text{ and } C = [\sigma_{ij}]. \tag{5.4}$$

C is called the covariance matrix for the assets and is symmetric positive semidefinite. Let x_i denote the proportion of wealth to be invested in asset i and let $x = (x_1, \ldots, x_n)^\top$. In terms of x, the expected return of the portfolio c_p and the variance of the portfolio σ_p^2 are given by

$$c_p = c^\top x \text{ and } \sigma_p^2 = x^\top C x. \tag{5.5}$$

Let $A = (1, 1, \ldots, 1)^\top$; i.e., A is n vector of ones. Since the components of x are proportions, they must sum to one; i.e., $A^\top x = 1$. The constraint $A^\top x = 1$ is usually called the *budget constraint.*

The goal is to choose a value for x which gives a large value for c_p and a small value for σ_p^2. These two goals tend to be in conflict. Suppose we have two portfolios, both having the same expected return but the first having a small variance and the

second having a large variance. The first portfolio is obviously more attractive because the second has more risk for the same expected return. This is the key idea behind H. Markowitz's definition of an *efficient portfolio.*

Definition 5.1 *A portfolio is* **efficient** *if for a fixed c_p, there is no other portfolio which has a smaller variance σ_p^2.*

Definition (5.1) implies that a portfolio is efficient if for fixed c_p, σ_p^2 is minimized. Thus the efficient portfolios are solutions of the optimization problem

$$\min\{x^\top Cx | c^\top x = c_p, A^\top x = 1\} \tag{5.6}$$

In (5.6), c_p is to vary over all possible values. For each value of c_p, we will, in general, get a different efficient portfolio. The mathematical structure of (5.6) is that of minimizing a quadratic function subject to linear equality constraints, the first of which is parametric (the parameter being c_p). Thus (5.6) is a parametric quadratic programming problem.

There is an alternative (and equivalent) definition of an efficient portfolio. Suppose we have two portfolios both having the same variance but the first having a large expected return and the second having a small expected return. The first portfolio is more attractive because it gives a higher expected return for the same risk as the second portfolio.

Definition 5.2 *A portfolio is* **efficient** *if for fixed σ_p^2, there is no other portfolio with a larger c_p.*

Definition (5.2) implies that a portfolio is efficient if for fixed σ_p^2, c_p is maximized. Using Definition (5.2), the efficient portfolios are the optimal solutions for

$$\max\{c^\top x | x^\top Cx = \sigma_p^2, A^\top x = 1\} \tag{5.7}$$

Note that (5.7) has a linear objective function, a quadratic equality constraint and a linear equality constraint.

There is a third optimization problem which also produces efficient portfolios. It is a somewhat more convenient formulation than (5.6), (5.7). Let t be a scalar parameter and consider the problem

$$\min\{-tc^\top x + \frac{1}{2}x^\top Cx | A^\top x = 1\} \tag{5.8}$$

The intuition behind (5.8) is as follows. For $t \geq 0$, the parameter t balances how much weight is placed on the maximization of $c^\top x$ (equivalently, the minimization of $-c^\top x$) and the minimization of $x^\top Cx$. If $t = 0$, (5.8) will find the minimum variance portfolio. As t becomes very large, the linear term in (5.8) will dominate and portfolios will be found with higher expected returns at the expense of variance. Finally, We assume that all variables are restricted by positive upper and lower bounds. (i.e. $0 \leq l \leq x \leq u$). It is possible to show that each of the optimization problems (5.6), (5.7) and (5.8) are equivalent in the sense that their optimal solutions, as all of the three parameters c_p, σ_p^2 and t are varied, are identical [46, 4].

5.12 Numerical Implementation

The algorithm explained in this section is based on the structure of the application of portfolio optimization. We consider formulation (5.8). It must be noted that most portfolio managers would like to have only a limited number of assets in their baskets, say 50 or 60 different funds. Thus, a typical problem with $n = 100$ variables, where C is dense, can demonstrate the efficiency of the proposed algorithm for the worst case scenario in terms of computational complexity.

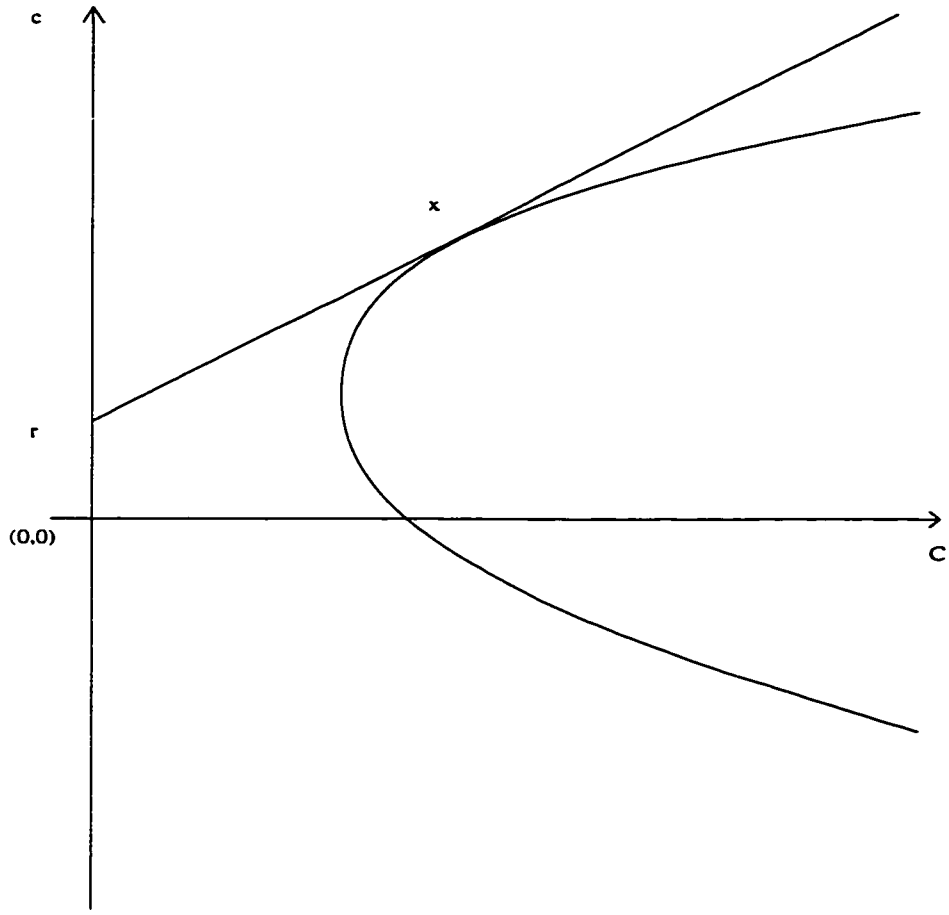


Figure 5.9: Risk Free Rate and Market Portfolio

The implementation of the modified Expanded Lagrangian Function (3.25) has a few advantages. First of all, the bound constraints do not increase the size of the problem. Secondly, for a given penalty parameter r at each minor iteration, only the diagonal elements of the matrix change; so it is enough to perform a full Choleskey factorization at the beginning of each minor iteration. This factorization can be used as a preconditioner for the next minor iterations when the Preconditioned Conjugate Gradient algorithm is implemented [47]. Algorithm (3.1) shows a simple prototype based on the modified Expanded Lagrangian System when $t = 1$. This algorithm is coded in MATLAB. We have generated uniform random numbers $U(0, 1)$ for both c and C. The matrix C is considered to be dense. We count for the average number of conjugate iterations. It is seen that the average number of conjugate iterations remains almost unchanged as the size of problem increases. Figure (5.10) shows the summary of the results.

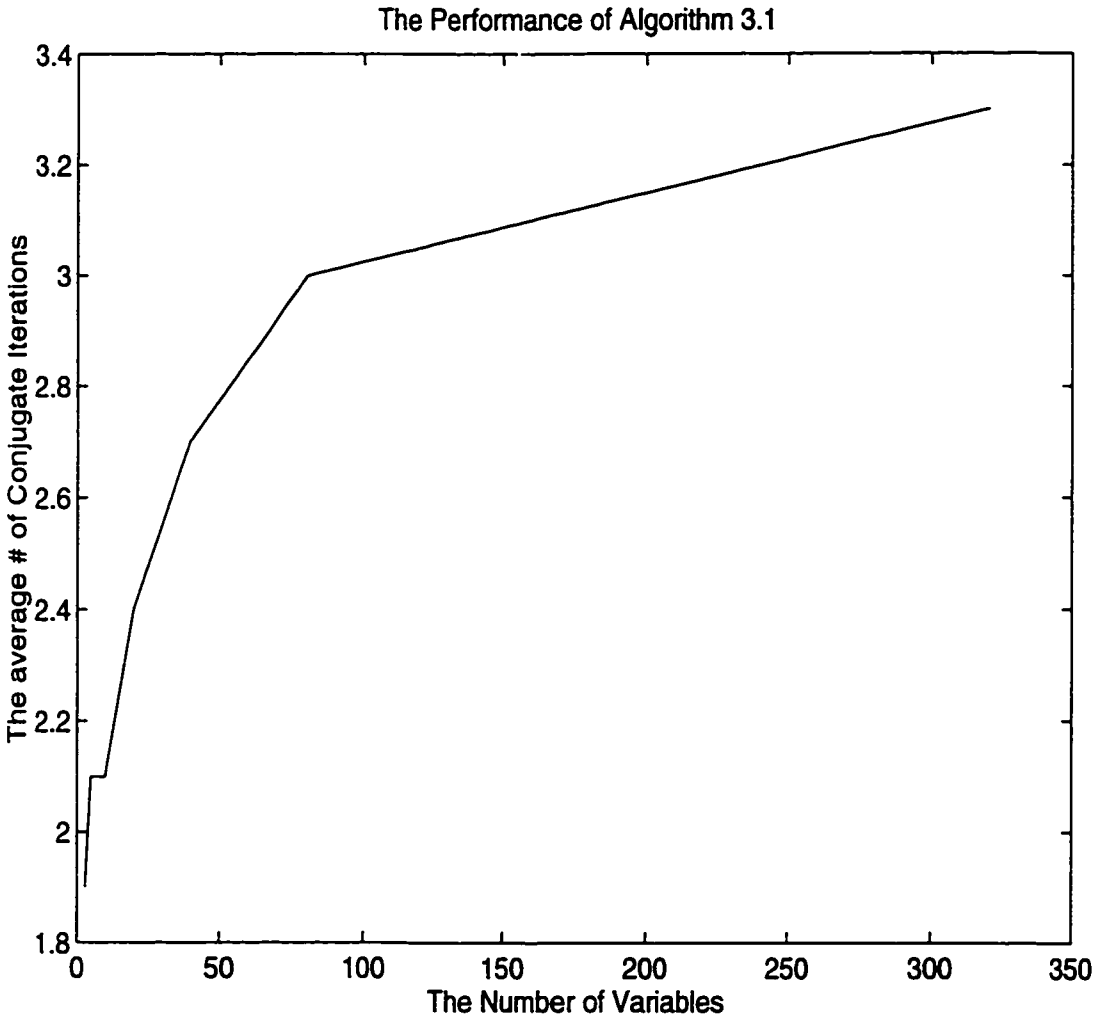


Figure 5.10: The Average Number of Conjugate Iterations in Each Minor Iteration.

5.13 Summary

In this section, we have explained the details of the implementation of the proposed method (PM) for two different applications. We have shown that the PM is able to provide an efficient tool to solve the Great Lakes water resource problem over a long period of time. The preliminary results from portfolio optimization have also shown that the Expanded Lagrangian Function can be implemented for Quadratic Programming and we can expect reasonable performance.

Chapter 6

Summary, Conclusions and Future Research

6.1 Summary

The research has been aimed at the development of a new method to solve the optimization problems arising in water resources management systems. The new technique has been tested on some small problems in order to show that the PM can be considered as an alternative for future research in optimization. However, based on this limited number of small benchmark problems, it is difficult to draw a general conclusion for the performance of the PM.

For a large-scale application, the proposed method has been implemented to find the optimum values of outflows and storages over the long term in a water resources problem. The research has provided practical tools in order to achieve, in a reasonable time, the minimum fluctuation of water for the Great Lakes reservoir system even for the largest possible case. The preliminary test results have also

indicated that the method can be applied on some simple but dense cases of portfolio optimization with good results.

Chapter (1) began by outlining the importance of the constrained optimization in water resources management problems. We explained that many water resources applications involve optimization problems characterized by high dimensionality, bad scaling, nonlinearity and non-convexity. These problems need to be solved with special techniques and algorithms. Chapter (1) continues by explaining that many large-scale water resources systems often serve multiple objectives and that these objectives are often in conflict with one another. Operation studies attempt to compromise between the different goals. Multi-reservoir operation planning (MROP) is modeled as a nonlinear, constrained optimization problem with the objective to determine the optimum storages and outflows of the reservoir over a certain time period. We have seen that the non-convexity characteristics constitute the primary difficulty in finding mathematical solution of the MROP problem. Although there are many methods to deal with these difficulties for small-scale problems, problems arise for large-scale MROP problems. The chapter introduced the objectives of the research. The organization of the Dissertation is then described by chapters.

Chapter (2) reviewed deterministic optimization procedures in the fields of linear programming and nonlinear programming as these are applied to planning and the real-time operation of multi-reservoir water systems for hydro-power and closely related purposes. Linear Programming (LP) has been introduced as the most widely used optimization technique among engineers. For some applications of water resources problems which have a simple formulation in nature, LP can be used directly. Two different methods based on piecewise linearization (PL) and the method based on repeated applications of LP to solve a series of approximated original problems called sequential linearization problem (SLP) were introduced. We

explained that SLP has been the most common method used in the field of reservoir operations, especially for the optimization of nonlinear large-scale reservoir systems, because of its ease of implementation.

Chapter (2) continues by introducing Dynamic Programming (DP) as another technique applicable to reservoir operation studies. The discrete and continuous methods of DP along with the advantages and disadvantages of these methods were explained. A literature review has shown that DP methods are widely used in deterministic and stochastic water resources management problems.

The chapter reviews the recent applications of water resources management problems based on direct use of Nonlinear Programming (NLP) techniques. We have seen that Sequential Quadratic Programming (SQP) is able to perform better on NLP problems than SLP. The reason is that (SQP) provides a better approximation of the original problem. We have explained that SQP can solve problems of small and medium sizes reasonably well. However, our survey has indicated that there has been little work using SQP for the application of large-scale MROP problems.

The chapter continues by studying the use of two different software packages, MINOS and LANCELOT. The research indicated that there has been a number of applications which have been successfully solved using MINOS. Large and Nonlinear Constrained Extended Lagrangian Optimization Techniques (LANCELOT) was introduced in this chapter as a reliable software package for solving large-scale nonlinear optimization problems. LANCELOT is a promising software package for solving all kinds of nonlinear applications in different fields of Engineering, including water resources management. However, the author could not find any other practical work in the field of water resources management, other than the one in this research, that has presented solution methodologies for large-scale MROPs.

Chapter (3) defines the problem statement that is of our concern in this dissertation. The chapter continues the study of traditional Penalty function and explains that it is possible to remove the singularities arising in the penalty function by expanding the equation. Some constrained qualification was introduced to insure that expanded equations do not diverge and also to insure that the optimum point is a Karush-Kuhn-Tucker point.

Special cases in which we can summarize the problem in a simple form have been studied. The previous work on the solution of Expanded Lagrangian System was based on the continuation method. The continuation method is promising on some highly nonlinear problems. However, it does not perform as expected on some problems with only a mild degree of nonlinearity.

Chapter (3) continues to study the use of the trust region algorithm to solve nonlinear equations. One algorithm, based on the concept of linear programming, minimizes the l_1-norm of the linearized vector within an l_∞-norm trust region. Finally, this chapter introduced the bi-dimensional trust region algorithm proposed by Martínez for solving nonlinear equations and then studied two similar curvilinear search direction algorithms. Convergence properties for both algorithms were proved at the end of the chapter. The chapter ends by describing the detailed computation of the two dimensional subproblem which arises in the bi-dimensional trust region algorithm.

Chapter (4) looks at various aspects of the implementation of the proposed method. The chapter compared the results of the proposed method with the performance of different codes and algorithms for several small test examples. The results indicate that the proposed method has good potential for solving general constrained optimization problems.

In Chapter (5), the effectiveness of the proposed method is examined by applying it to the Great Lakes levels' regulation problem. This five reservoir control problem involves the consideration of the natural and nonlinear stage-discharge relationships for the three middle lakes, Lakes Michigan-Huron, St-Clair and Erie, and full control only on Lakes Superior and Ontario. This case study gives some indication of the robustness of the method proposed in this research. The Great Lakes problem was solved for up to 90 years of monthly data. The implementation of LANCELOT on the Great Lakes problem indicates that the proposed method is able to provide somewhat better results than LANCELOT in terms of lower values for the objective function and CPU time in most cases. The Great Lakes problem was solved using different features of LANCELOT. Our experimental results have shown that providing exact information does not always improve the final solution. In fact we have obtained a better results using approximate techniques (Case two) in terms of CPU time. We have shown that the implicit implementation of the bound constraints on Expanded Lagrangian Function can save a significant amount of memory and reduce the size of the linear systems solved very significantly but need not affect CPU time significantly due to changes in the number of iterations.

In addition, the levels and releases of the Great Lakes was simulated over the long term using a rolling-horizon. The results were compared with the original objective. It was found that both methods can minimize the fluctuation of water significantly.

6.2 Conclusions

General Nonlinear Programming has always been considered a challenging area of research. Today, advances in the field of nonlinear programming methods and

the availability of fast and large-memory computers have motivated researchers to persist in this area of study and to provide several reliable software packages for practitioners.

We have proposed a new algorithm for the solution of large-scale nonlinear programming optimization arising in the context of water resources problems. The proposed method was used on a variety of small test problems. The modified Trust Region algorithm used in this dissertation, theoretically, has the same convergence properties as the original. It was shown that the proposed method can be practically used in the Great Lakes system to minimize the fluctuation of water significantly. Also, it seems that the proposed method can be applied to portfolio optimization. Based on the preliminary numerical test results of the algorithm, we hope that the algorithm will prove to be useful in practice for more general large-scale constrained optimization problems.

6.3 Future Research

Different aspects of research, both in general nonlinear programming and in Expanded Lagrangian Functions need to be explored. We believe there are many different areas in this field for future research and we would like to suggest some of them as possible future research areas:

Feasibility. The assumption in using the Expanded Lagrangian Function is that there is a feasible point for inequality constraints. It was suggested that one can use a penalty loss function to find a point which is at least close to the boundary and then switch to an expanded form of the equations by introducing a shift parameter. This alternative may introduce a singularity

as the solution x gets close to the optimal solution. Another alternative is to apply interior point algorithms to a linear or quadratic approximation of the inequality constraints. This is currently a new open research area.

Penalty Parameter. One of the most sensitive elements of the Expanded Lagrangian Function is the penalty parameter. There have been many suggestions on how to introduce a reasonable choice to reduce the penalty parameter in each major iteration. The proposed method has reduced the penalty parameter by a single factor and this is similar to LANCELOT. The main reason for using this alternative is to have a fair comparison with the output performance on our case study problem. However, we believe this is an open area of research and it is quite possible to improve the efficiency of the proposed method by choosing a more logical pattern to reduce the penalty parameter.

Special case studies. We have shown that the proposed method can be implemented to solve some special case studies very efficiently. The application of Portfolio Optimization has demonstrated the capability of the proposed method for handling of Quadratic Programming. However, we believe that the structure of the Expanded Lagrangian Function can be used in other applications of Engineering, Science, etc.

Testing. The proposed method was tested on some benchmark problems. It is obvious that these small test problems cannot demonstrate the robustness and the efficiency of the proposed method in general. Today, with the availability of the CUTE library with a different number of test problems, and the availability of LANCELOT as a competitive alternative to the proposed method, it is quite possible to have a better evaluation on the performance of the proposed method. However, this is very time consuming project. Another open

research area is when we do not have the information of the first and second derivative for some particular problem, or when they may be just too costly to evaluate analytically. These difficulties can create some new research to study the convergence of the proposed method when we apply Rank one or Rank two algorithms to estimate the derivative in the proposed method.

Appendix A

Preconditioned Conjugate Gradient Method

Preconditioned conjugate gradient (PCG) methods have recently become popular for solving large, symmetric, positive definite systems of linear equations. There are many different preconditioners to improve the efficiency of the traditional conjugate gradient method. Incomplete Cholesky (ILU) preconditioner is one of these commonly used preconditioners.

Section one of this appendix briefly describes the Conjugate Gradient (CG) method. In section two, we explain the description of the Preconditioned Conjugate Gradient method and finally the drop-tolerance preconditioner is discussed in section three.

A.1 Conjugate Gradient Method

Consider a system of linear equations

$$Ax = b \tag{A.1}$$

where $x,\ b\ \in\ \mathbb{R}^n$ and A is $n \times n$ matrix assumed to be symmetric and positive definite. It is an easy task to show that the minimum of the quadratic function $\Psi(x)$ is identical to answer of this system of linear equations.

$$\Psi(x) = \frac{1}{2}x^\top Ax - x^\top b \tag{A.2}$$

The minimum value of Ψ is achieved by setting the first derivative of $\Psi(x)$ equal to zero, or equivalently by setting[1] $x = A^{-1}b$. Thus, minimizing Ψ and solving equation (A.1) are equivalent problems.

The method of steepest descent is one method to minimize Ψ. At a current point x_c the function Ψ decreases most rapidly in the direction of the negative gradient

$$-\nabla\Psi(x_c) = b - Ax_c = r_c, \tag{A.3}$$

where r_c is defined as the residual at x_c. Now, if this residual is nonzero, then there exists a positive number η such that

$$\Psi(x_c - \eta r_c) < \Psi(x_c). \tag{A.4}$$

In the steepest descent method, one would move along the negative gradient and for problems with large condition numbers (ill-conditioned matrices) one is forced to traverse back and forth *across* the valley rather than *down* the valley[2]. In order

[1] A is assumed to be nonsingular

[2] except when $-\nabla\Psi$ coincides with the eigenvector corresponding to the smallest eigenvalue of the matrix

to avoid such a problem we switch to another approach called A-Conjugate search directions.

A.1.1 Conjugate Directions

If A is a symmetric, positive definite matrix, then two nonzero vectors $p, q \in \mathbb{R}^n$ are conjugate (with respect to A) if $p^\top Aq = 0$. A set of n nonzero vectors $p_1, \ldots, p_n$ are conjugate basis if p_i and p_j are conjugate for all $i \neq j$. (For a detailed description of these directions see [39]).

A.1.2 Conjugate Gradient Algorithm

The Conjugate gradient method in general is given as [27]:

STEP 1. Initialize x, r and the iteration counter k.

$$k = 0;\ x_o = 0;\ r_o = b;$$

STEP 2. Do while $r < toler$ and $k < k_{max}$

$$k \leftarrow k + 1;$$

if $k = 1$

$$p_1 \leftarrow r_o;$$

else

$$\beta_k \leftarrow r_{k-1}^\top r_{k-1} / r_{k-2}^\top r_{k-2};$$

$$p_k \leftarrow r_{k-1} + \beta_k p_{k-1};$$

end

$$\eta_k \leftarrow r_{k-1}^\top r_{k-1} / p_k^\top A p_k;$$

$$x_k \leftarrow x_{k-1} + \eta_k p_k;$$

$$r_k \leftarrow rk - 1 - \eta_k A p_k;$$

end loop

STEP 3. solution $x = x_k$ STOP

where r = residual, k = iteration counter, p = search direction, η = step length and β = ratio of the most recent and previous residual norm.

If exact arithmetic were to be performed then the conjugate gradient method would require at most n iterations to converge for a problem in $\mathbb{R}^n$. This result is explained in the following theorem.

Theorem A.1 *Given the quadratic function Ψ, where A is symmetric, positive definite matrix, then the conjugate gradient iterates of the above algorithm satisfy $x_m = A^{-1}b$ for some $m \leq n$. Moreover,*

$$(Ap_i)^T p_j = 0, \quad i \neq j, \quad 0 \leq i, \quad j \leq m \tag{A.5}$$

Proof: See [39] •

Details on the convergence properties of CG methods are discussed in various texts like the one by Luenberger [39].

Conjugate gradient methods work well on matrices that are either very well conditioned or have just a few distinct eigenvalues. Unfortunately, matrices in real world problems (usually) are not well conditioned. Hence, in order to come up with such problems it is necessary to use preconditioners so that they become easier to solve.

A.2 Preconditioned Conjugate Gradient method

Preconditioned conjugate gradient method is a modification of the conjugate gradient method where the matrix of coefficients, A, is preconditioned so that the system

becomes reasonably well conditioned.

A.2.1 Derivation

Consider the $n \times n$ symmetric positive definite linear system $Ax = b$. The idea behind PCG methods is to apply the *regular* conjugate algorithm to the transformed system

$$\hat{A}\hat{x} = \hat{b} \tag{A.6}$$

where

$\hat{A} = C^{-1}AC^{-1}, \quad \hat{x} = Cx, \quad \hat{b} = C^{-1}b$ and C is symmetric positive definite matrix.

In order to improve the efficiency of the algorithm it is essential to choose C such that $\hat{A}$ is well conditioned or has few clustered eigenvalues. Applying the conjugate gradient to this transformed system results in the PCG algorithm.

A.2.2 PCG Algorithm

The following algorithm [27] solves the linear system $Ax = b$ using the method of conjugate gradients with preconditioner $M \in \mathbb{R}^{n \times n}$.

STEP 1. Initialize x, r and the iteration counter k.

$k = 0$; $x_o = 0$; $r_o = b$;

STEP 2. Do while $r < toler$ and $k < k_{max}$

Solve $Mz_k = r_k$

$k \leftarrow k + 1$;

if $k = 1$

$p_1 \leftarrow z_o$;

else

$$\beta_k \leftarrow r_{k-1}^\top z_{k-1} / r_{k-2}^\top z_{k-2};$$

$$p_k \leftarrow z_{k-1} + \beta_k p_{k-1};$$

end

$$\eta_k \leftarrow r_{k-1}^\top z_{k-1} / p_k^\top A p_k;$$

$$x_k \leftarrow x_{k-1} + \eta_k p_k;$$

$$r_k \leftarrow rk - 1 - \eta_k A p_k;$$

end loop

STEP 3. solution $x = x_k$ STOP

A few important observations for this method are:

- The residuals and search directions satisfy

$$r_j^\top M^{-1} r_i = 0 \qquad i \neq j \tag{A.7}$$

$$p_j^\top (C^{-1} A C^{-1}) p_i = 0 \qquad i \neq j \tag{A.8}$$

- The denominators $r_{k-2}^\top z_{k-2} = z_{k-2}^\top M z_{k-2}$ never vanish because M is positive definite.

- For this algorithm to be effective, linear systems of the form $Mz = r$ must be rapidly solved.

The choice of a good preconditioner can have dramatic effect on the rate of convergence. There are many preconditioners which have been used and tested. Selecting the right preconditioner is often a difficult task, since some preconditioners work better on some classes of problems while others perform better on other classes of problems.

A.3 The Use of Drop Tolerances to Preserve Sparsity

One of the main problems associated with the use of preconditioners lies in the increase in the number of entries due to fill-in [16]. Indeed, the storage of the matrix factorization often limits the size of problem which can be solved. A simple way to extend this limit is to remove from the sparsity pattern and also from any subsequent calculations any entry which is less than some prescribed absolute or relative tolerance, usually called a *drop tolerance*. So the idea can be summarized as keeping any intermediate value satisfies the inequality

$$|a_{ij}^{(k)}| < TOL_a \tag{A.9}$$

or the inequality

$$|a_{ij}^{(k)}| < TOL_r \times max_l |a_{lj}^{(k)}|, \tag{A.10}$$

for a preset non-negative value of TOL_a or TOL_r, then $a_{ij}^{(k)}$ is dropped from the sparsity pattern and subsequent consideration. In order to have a useful preconditioner, tolerances have to be set high enough to reduce substantially the number of entries in the factors.

Appendix B

The Solution of Trust Region Subproblem

This appendix explains a method to minimize a quadratic objective function subject to a quadratic constraint. Section B.1 defines the problem and Section B.2 outlines the algorithm used for solving this problem.

B.1 The Problem

Consider the minimization of a quadratic objective function subject to a simple constraint with the following form,

$$\begin{aligned} \text{Minimize} \quad & f(x) = x^T C x + 2x^T c \\ \text{s.t.} \quad & x^T x \leq t^2 \end{aligned} \tag{B.1}$$

where C is 2×2 positive definite matrix, c is a vector of length 2, t is positive scalar, and x is the vector of length 2, which is being solved for.

B.2 The Algorithm

The system (B.1) is solved in two stages:

Stage 1 Solve the unconstrained quadratic directly by solving the following linear system obtained from the first order necessary conditions:-

$$\nabla f(x) = 0 \tag{B.2}$$

or

$$Cx = -c \tag{B.3}$$

which, for this two-dimensional problem, is equivalent to:

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Stage 1b Check if the constraint $x^T x \leq t^2$ is satisfied. If yes then stop, else; go to stage 2.

Stage 2 This stage is only executed when the constraint is not satisfied at the unconstrained minimum of the problem solved in stage 1. This implies that at the minimum of the constrained problem the quadratic constraint will be active. Therefore, stage 2 involves solving the modified problem:

$$\begin{aligned} \text{Minimize} \quad & x^\top C x + 2x^\top c \\ \text{s.t.} \quad & h(x) = t^2 - x^\top x = 0 \end{aligned} \tag{B.4}$$

This problem is solved using the Lagrange-Newton (SOLVER) method. Details on this method are covered in many books (see [21], [25]).

The Lagrangian function is given as:

$$\ell(x,\lambda) \;=\; x^{\top}Cx \;+\; 2x^{\top}c \;-\; \lambda(t^2 - x^{\top}x) \tag{B.5}$$

$\nabla\ell(x_k,\lambda_k)$ is the gradient of the Lagrangian and $W(x_k,\lambda_k)$ is the Hessian of the Lagrangian.

In its most straightforward form, the Newton method solves the Lagrange system:-

$$\begin{aligned} \nabla\ell(x,\lambda) \;&=\; 0 \\ h(x) \;&=\; 0 \end{aligned} \tag{B.6}$$

by solving the linearized version recursively. That is, given x_k, λ_k the new point x_{k+1}, λ_{k+1} is determined from the equations

$$\begin{aligned} \nabla\ell(x_k,\lambda_k) \;+\; W(x_k,\lambda_k)d_k \;+\; \nabla h(x_k)^T y_k \;&=\; 0 \\ h(x_k) \;+\; \nabla h(x+k)d_k \;&=\; 0 \end{aligned} \tag{B.7}$$

by setting $x_{k+1} = x_k + d_k$, $\lambda_{k+1} = \lambda_k + y_k$. In matrix form the above Newton equations are

$$\begin{bmatrix} W(x_k,\lambda_k) & \nabla h(x_k)^{\top} \\ \nabla h(x_k) & 0 \end{bmatrix} \begin{bmatrix} d_k \\ y_k \end{bmatrix} \;=\; \begin{bmatrix} -\nabla\ell(x_k,\lambda_k)^{\top} \\ -h(x_k) \end{bmatrix} \tag{B.8}$$

now by adding $\nabla h(x_k)^{\top}\lambda_k$ to the above equation, the system can be transformed to the form

$$\begin{bmatrix} W(x_k,\lambda_k) & \nabla h(x_k)^{\top} \\ \nabla h(x_k) & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_{k+1} \end{bmatrix} \;=\; \begin{bmatrix} -\nabla f(x_k)^{\top} \\ -h(x_k) \end{bmatrix}, \tag{B.9}$$

where again $\lambda_{k+1} = \lambda_k + y_k$. Solving the above system is very easy especially since the problem is only two dimensional, making the system (B.9) a 3×3 system of linear equations.

Stage 2b Repeat stage 2 until the gradient of the Lagrangian is approximately equal to zero.

Appendix C

Matrix by matrix products

This appendix briefly describes a simple algorithm to provide a product of two matrices A and B when both of them are conformable and sparse. For full matrices, the usual way of computing C is to consider each entry c_{ij} as an inner product of the i-th row $B_{i\bullet}$ of B and the j-th column $A_{\bullet j}$ of A, that is

$$c_{ij} = \sum_k b_{ik} a_{kj} = B_{i\bullet} A_{\bullet j}. \tag{C.1}$$

The trouble with this formula for a general sparse matrix is that it is very difficault to avoid performing multiplications $b_{ik}a_{kj}$ with one or other of the factors having the value zero. An explicit test for a zero is likely to be equally expensive. For example, if A is stored by columns and B is stored by rows, column j of A may be loaded into a full vector and c_{ij} may be calculated by scanning the entries of row i of B. This means that all entries of B are scanned for each column of A. If the sparsity pattern of C is already known, c_{ij} need not to be calculated unless it is an entry, but there are still likely to be many occasions when we multiply an entry of B by a zero in the full vector.

None of these unnecessary operations is performed with the *outerproduct* formulation as a sum of rank-one matrices,

$$C = \sum_k B_{\bullet k} A_{k\bullet}, \tag{C.2}$$

which is natural if B is stored by columns and A is stored by rows(note that $B_{\bullet k}$ is a column vector and $A_{k\bullet}$ is a row vector, so $B_{\bullet k} A_{k\bullet}$ has the shape of C). If both matrices are stored by columns, columns j of C may be accumulated as linear combination of the columns of B by expressing (C.2) in the form

$$C_{\bullet j} = \sum_k a_{kj} B_{\bullet k}. \tag{C.3}$$

If both matrices are stored by rows, row i of C may be accumulated as a linear combination of the rows of A by the formula

$$C_{i\bullet} = \sum_k b_{ik} A_{k\bullet}, \tag{C.4}$$

and the required work is identical to that when A and B are both stored by columns. An imortant special case is when $B = A^T$, which can arise in the least-squares problem, in which case C is the normal matrix. If A is stored by rows, we have the case (C.2) with $B_{\bullet k} = A^T_{k\bullet}$ which yields

$$C = \sum_k A^T_{k\bullet} A_{k\bullet}. \tag{C.5}$$

If A is stored by columns we have the case of (C.1) with $B_{i\bullet} = A^T_{\bullet i}$ which, in general, will not very efficient. Indeed it may be perferable to make a copy of A that is stored by rows. We have used this modification to provide the product of the Jacobian by its transpose in sparse form.

Bibliography

[1] Q. Al-Hassan. *Continuation methodology for solving constrained optimization problems.* PhD thesis, Colorado State University, Colorado State University,Fort Collins,CO, 1987.

[2] M. Bachem, A. Grotschel and B. Korte. *Penalty function, in Mathematical Programming: The State of Art.* Springer-Verlag, New York, 1983.

[3] R. Bellman. *Dynamic programming.* Princeton University Press, Princeton, 1962.

[4] J. Best, M. Grauer and R. Robert. Positively weighted minimum variance portfolios and the structure of asset expected returns. *Journal of Financial and Quantitative Analysis*, 27(4):513–537, 1992.

[5] M. J. Best and K. Ritter. *Active set analysis and computer programs*, volume 1. Prentice-Hall, Englewood Cliffs, 1985.

[6] A. Brooke, D. Kendrick, and A. Meeraus. *GAMS, A USER'S GUIDE.* The Scientific Press, 507 Seaport Court, Redwood City, CA 94063-2731, 1988.

[7] Environment Canada. Great lakes water levels, 1993.

[8] R.G. Carter. On the global convergence of trust region algorithms using inexact gradient information. *SIAM J. NUMER ANAL.*, 28(1):251–265, 1991.

[9] P. Chin and A. Vannelli. Iterative methods for the augmented equations in large-scale linear programming. Technical report, University of Waterloo, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada, 1994.

[10] S.N. Chow and J.K.Hale. *Methods of bifurcation theory.* Springer- Verlag, New York, 1982.

[11] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A).* Number 17 in Springer Series in Computational Mathematics. Springer Verlag, Heidelberg, Berlin, New York, 1992.

[12] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Large-scale nonlinear constrained optimization. In *Linear Algebra for Large Scale and Real-Time Applications*, volume 232, pages 21–48. Kluwer Academic Publishers, Netherlands, 1993.

[13] G. Dantzig. *Linear programming and extensions.* Princeton University Press, second edition, 1963.

[14] J.E. Dennis and R.B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall,Englewood Cliffs, NJ, 1983.

[15] G.E. Diaz and D.G Fontane. Hydro-power optimization via sequential quadratic programming. *Journal of Water Resources Planning and Management*, 115(6):715–734, 1989.

[16] A. M. Duff, I. S. Erisman and J. K. Reid. *Direct methods for sparse matrices*, volume 2. Ipswich Book Company Ltd, Great Britain, 1992.

[17] J. Duff, I.S. Nocedal and Reid K. The use of linear programming for the solution of sparse sets of nonlinear equations. *SIAM J. SCI. STAT. COMPUT.*, 8(2):99–108, 1987.

[18] J. H Ellis and C. S Revelle. Separable linear algorithm for hydro-power optimization. *Water Resources Bulltin WARBAQ*, 24(2):435–447, 1988.

[19] A.V. Fiacco and G.P. McCormick. *Nonlinear sequential unconstrained minimization techniques.* John Wiley, New York, 1968.

[20] R. Fletcher. A modified marquardt subroutine for nonlinear least squares. Technical report, Report R6799, Harwell, England, Atomic Energy Research Establishment, 1971.

[21] R. Fletcher. *Practical methods of optimization*, volume 2. John Wiley, New York, 1981.

[22] R. Fletcher. Harwell subroutine library. Technical report, Harwell, NS03A, 1997.

[23] S. G. Fletcher. *A new formulation for the stochastic control of systems with bounded state variables: An application to reservoir systems.* PhD thesis, University of Waterloo, Department of Systems Design Engineering, 1995.

[24] J.A. George. Computer implementation of the finite element method. *STAN-CS-,Computer Sci*, pages 71–208, 1971.

[25] W. Gill, P.E. Murray and M.H. Wright. *Practical optimization.* Academic Press, New York, 1981.

[26] W. Gill, P.E. Murray and M.H. Wright. On projected newton barrier methods for linear programming and an equivalence to karmarkar's projective method. *Mathematical Programming*, AB(36):183–209, 1986.

[27] G. H. Golub and C. Van Loan. *Matrix computations.* The Johns Hopkins University Press, Baltimore, Maryland, 1989.

[28] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. St. Comp.*, 49:409–436, 1952.

[29] K.L. Hiew. *Optimization algorithms for large-scale multi-reservoir hydropwer systems.* PhD thesis, Colorado State University, Colorado State University,Fort Collins,CO, 1987.

[30] K. Hock, W. Schittkowski. *Test examples for nonlinear programming codes*, volume 187. Springer-Verlag, New York, 1981.

[31] H. R. Howson and N. G. F. Sancho. A new algorithm for the solution of multi-state dynamic programming problems. *Mathematical Programming*, 8:101–116, 1975.

[32] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *16th Annual ACM Symposium on the Theory of Computing*, pages 302–311, 1984.

[33] V. Klee and G. L. Minty. How good is the simplex algorithm? *O. Shisha (ed.) Inequalities III, Academic Press*, pages 159–179, 1972.

[34] H.D. Kupper, T. Mittelmann and H. Weber. *Numerical methods for bifurcation problems.* Birkhauser-Verlag, Boston, 1984.

[35] K. E. Lansey and L. W. Mays. *Reliability Analysis of Water Distribution Systems*, volume 1. ASCE, New York, N.Y., 1989.

[36] W. Liu and A.H. Sherman. A comparison analysis of the cuthill-mckee and reverse chuthil-mckee ordering algorithms for sparse matrices. *SIAM-J.-Nummer.-Anal*, 13:198–213, 1976.

[37] F.A. Lootsma. *Numerical methods for nonlinear optimization.* Academic Press, New York, 1972.

[38] F.A. Lootsma. *A survey of methods for solving constrained minimization problems via unconstrained minimization,in Numerical Methods for Nonlinear Optimizaiion.* Academic Press, New York, 1972.

[39] D. G. Luenberger. *Linear and nonlinear programming.* Addison Wesley, second edition, 1984.

[40] D.G. Luenberger. *Linear and nonlinear programming.* Addison-Wesley, MA, 1984.

[41] B. N. Lundberg and A. B. Poore. *Smooth penalty functions and continuation methods for constrained optimization*, volume 26. American Mathematical Society, Lectures in Applied Mathematics, 1990.

[42] B. N. Lundberg and A.B. Poore. Variables order adams-bashforth predictors with an error-step-size control for continuation methods. *SIAM J. SCI. STAT. COMPUT.*, 12(3):695–723, 1991.

[43] B.N. Lundberg, A. B. Poore, and B. Yang. Smooth penalty functions and constrained methods for constrained optimization. *Computational Solution of Nonlinear Systems of Equations*, 26:389–412, 1990.

[44] N. Mao and L. W. Mays. Goal programming models for determining freshwater inflows to estuaries. *Journal of Water Resources Planning and Management*, 120(3):316–329, 1994.

[45] M.A. Marino and H.A Loaiciga. Dynamic model for multi-reservoir operation. *Water Resources Research*, 21(5):619–630, 1985.

[46] H.M Markowitz. *Portfolio selection: Efficient diversification of investments*, volume 1. Coyles Foundation Monograph, Yale University Press, NewHaven and London, 1959.

[47] J.M. Martínez. An algorithm for solving sparse nonlinear least-square problems. *Computing*, 39:307–325, 1987.

[48] J.M. Martínez. An algorithm for solving nonlinear least-square problems with a new curvilinear search. *Computing*, 44:83–90, 1990.

[49] G.P. McCormick. *Nonlinear programming, theory , algorithms , and applications.* John Wiley and Sons, New York, 1983.

[50] R. D. C. Monteiro, I Alder, and M. G. C. Resende. A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research*, 15:191–214, 1990.

[51] N. S. Munksgaard and J. K. Reid. Ns02-a fortran subroutine for sparse sets of nonlinear equations by powell's dog-leg algorithm. Technical report, Harwell, Report AERE R.11020, AERE, 1983.

[52] A. Ostfeld and U. Shamir. Optimal operation of multi-quality networks. *Journal of Water Resources Planning and Management I:Steady conditions*, 119(6):645–662, 1993.

[53] A. Ostfeld and U. Shamir. Optimal operation of multi-quality networks. ii: Unsteady conditions. *Journal of Water Resources Planning and Management*, 119(6):663–684, 1993.

[54] P. Palacios-Gomez, L. Lasdon, and M. Engquist. Nonlinear optimization by successive linear programming. *Management Science*, 2(10):1106–1120, 1982.

[55] B. J. C. Perera and G. P Codner. Reservoir targets for urban water supply systems. *Journal of Water Resource Planning and Management*, 122(4):270–279, 1996.

[56] S. Pezeshk, O. J. Helweg, and K. E. Oliver. Optimal operation of ground water supply distribution systems. *Journal of Water Resources Planning and Management*, 120(5):573–586, 1994.

[57] K. Ponnambalam. *Optimization of the integrated operation of multi-reservoir irrigation systems.* PhD thesis, University of Toronto, Dep. of Civ. Eng., Toronto, Ont., Canada, 1987.

[58] A.B. Poore and Q. AL-Hassan. The expanded lagrangian system for constrained optimization problems. *SIAM J.Control and Optimization*, 26:417–427, 1988.

[59] A.B. Poore and Q. Al-Hassan. The expanded lagrangian systems for constrained optimization problems. *SIAM J. Control and Optimization*, 26(2):417–427, 1988.

[60] A.B. Poore and C.A. Tiahrt. Bifurcation problems in nonlinear parametric programming. *Mathematical Programming*, 39:189–205, 1987.

[61] H. Raman and V. Chandramouli. Deriving a general operating policy for reservoirs using neural network. *Journal of Water Resources Planning and Management*, 122(5):342–347, 1996.

[62] J. K. Reid. Least squares solution of sparse systems of nonlinear equations by a modified marquardt algorithm, in decomposition of large-scale problems. *D. M. Himmelblau*, pages 437–445, 1973.

[63] K. K. Reznicek and S.P. Simonovic. An improved algorithm for hydro-power optimization. *Water Resource Research*, 2(26):189–198, 1990.

[64] W.C. Rheinboldt. *Numerical analysis of parameterized nonlinear equations.* John Wiley, New York, 1986.

[65] S.M. Robinson. A quadratically convergent algorithm for general nonlinear programming problems. *Mathematical Programming*, 3:145–156, 1972.

[66] G. C. Santana and S Soares. Optimization model for water distribution systems design. *Computer Techniques and Applications*, pages 299–307, 1992.

[67] R. Schnabel and E. Eskow. A new modified choleskey factorization. *SIAM J,Sci.Stat.Comput.*, 11(90):1136–1158, 1990.

[68] A. Seifi. *A constant-potential infeasible-interior-point algorithm with application to stochastic linear program.* PhD thesis, University of Waterloo, Department of Systems Design Engineering, 1996.

[69] L.F. Shampine and M.K. Gordon. *Computer solution of ordinary differential equations: The initial value problem.* W. H. Freeman, San Francisco, 1975.

[70] R. Srinivasan and S. P Simonovic. A reliability programming model for hydropower optimization. *Canadian Journal of Civil Engineering*, 21(6):1061–1073, 1994.

[71] T. H. Sun, W. W. Ye, and N. S Hsu. Generalized network algorithm for water supply system optimization. *Journal of Water Resources Planning and Management*, 121(5):392–399, 1995.

[72] Y. Sun, S. L. Yang, W. G. Yeh, and P. W. F. Louie. Modeling reservoir evaporation losses by generalized networks. *Journal of Water Resources Planning and Management*, 122(3):222–227, 1996.

[73] T. Tao and W. C. Lennox. Reservoir operations by successive linear programming. *Journal of Water Resources Planning and Management*, 117(2):274–280, 1991.

[74] J. A. Tejada-Guibert, J. R. Stedinger, and K. Staschus. Optimization of value of cvp's hydro-power production. *Journal of Water Resources Planning and Management*, 116(1):52–70, 1990.

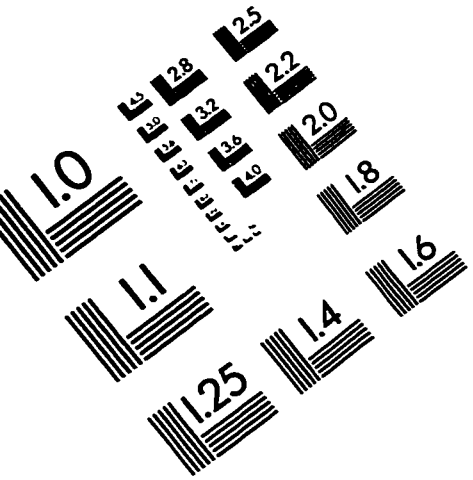
[75] W. J Trott and W. G Yeh. Use of stochastic dynamic programming for reservoir management. *Water Resource Research*, 23(6):983–996, 1987.

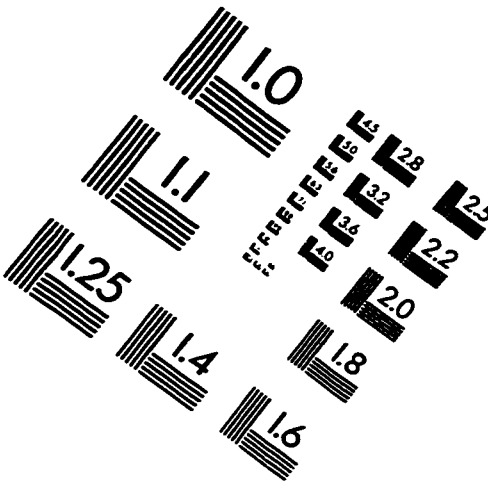
[76] A. Turgeon. Optimal short term scheduling of hydro-plants in series. In *Proceeding of international symposium on real time operation of hydro-systems*, volume 1, pages 108–122, 1981.

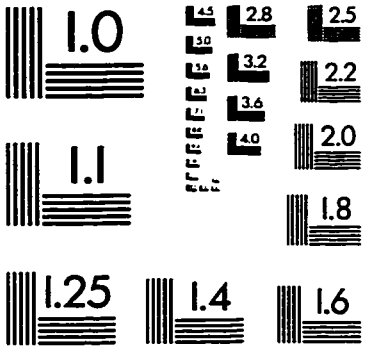
[77] R. J. Vanderbei and T. J. Carpenter. Symmetric indefinite systems for interior point methods. *Mathematical Programming*, 58:1–32, 1993.

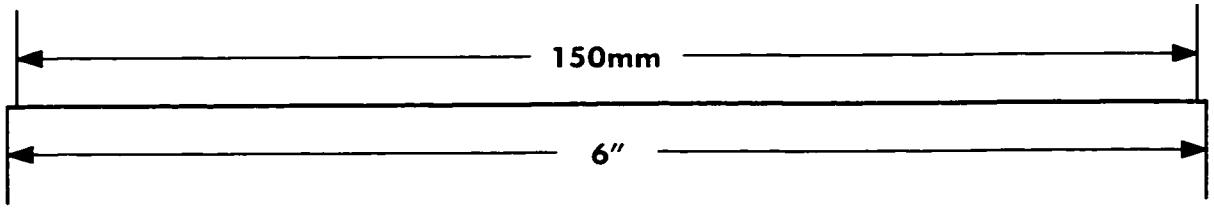
[78] K.A. Williamson. *A robust trust region algorithm for nonlinear programming.* PhD thesis, Department of Computational and Applied Mathematics, Rice University, TR90-22, 1991.

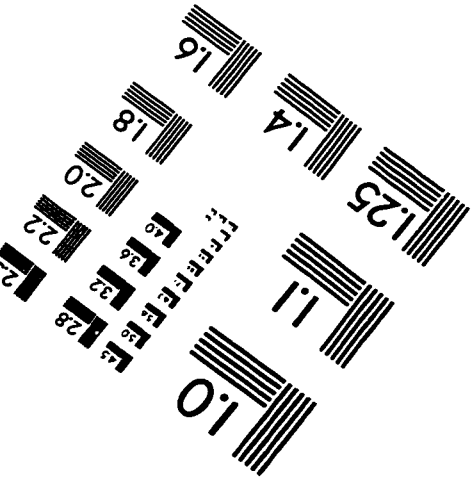
IMAGE EVALUATION
TEST TARGET (QA-3)

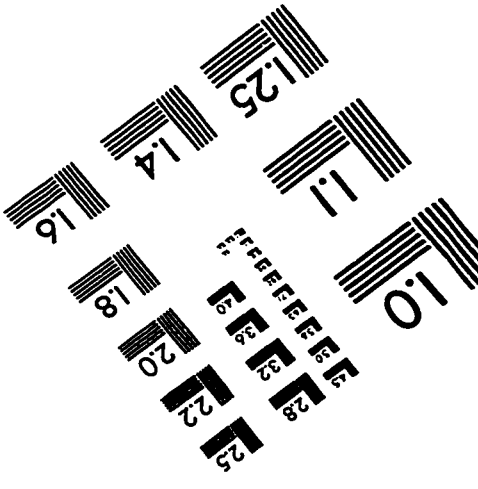












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989