Lagrangian Relaxation / Dual Approaches For Solving Large-Scale Linear Programming Problems

by

Ananth R. Madabushi

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Industrial and Systems Engineering

APPROVED:

Dr. Hanif D. Sherali, Chairman

Dr. Sheldon H. Jacobson

Dr. John E. Kobza

February, 1997
Blacksburg, Virginia.

(Keywords: Lagrangian Relaxation, Subgradient, Line Search, Primal Recovery, Penalty Function)

Lagrangian Relaxation / Dual Approaches for Solving

Large-Scale Linear Programming Problems

by

Ananth R. Madabushi

Dr. Hanif D. Sherali, Chairman

Industrial and Systems Engineering

ABSTRACT

This research effort focuses on large-scale linear programming problems that arise in the context of solving various problems such as discrete linear or polynomial, and continuous nonlinear, nonconvex programming problems, using linearization and branch-and-cut algorithms for the discrete case, and using polyhedral outer-approximation methods for the continuous case. These problems arise in various applications in production planning, location-allocation, game theory, economics, and many engineering and systems design problems. During the solution process of discrete or continuous nonconvex problems using polyhedral approaches, one has to contend with repeatedly solving large-scale linear programming(LP) relaxations. Thus, it becomes imperative to employ an efficient method in solving these problems. It has been amply demonstrated that solving LP relaxations using a simplex-based algorithm, or even an interior-point type of procedure, can be inadequately slow (especially in the presence of complicating constraints, dense coefficient matrices, and ill-conditioning) in comparison with a Lagrangian Relaxation approach. With this motivation, we present a practical primal-dual subgradient algorithm

that incorporates a dual ascent, a primal recovery, and a penalty function approach to recover a near optimal and feasible pair of primal and dual solutions.

The proposed primal-dual approach is comprised of three stages. Stage I deals with solving the Lagrangian dual problem by using various subgradient deflection strategies such as the Modified Gradient Technique (MGT), the Average Direction Strategy (ADS), and a new direction strategy called the Modified Average Direction Strategy (M-ADS). In the latter, the deflection parameter is determined based on the process of projecting the unknown optimal direction onto the space spanned by the current subgradient direction and the previous direction. This projected direction approximates the desired optimal direction as closely as possible using the conjugate subgradient concept. The step-length rules implemented in this regard are the Quadratic Fit Line Search Method and a new line search method called the Directional Derivative Line Search Method in which we start with a prescribed step-length and then ascertain whether to increase or decrease the step-length value based on the right-hand and left-hand derivative information available at each iteration. In the second stage of the algorithm (Stage II), a sequence of updated primal solutions is generated using some convex combinations of the Lagrangian subproblem solutions. Alternatively, a starting primal optimal solution can be obtained using the complementary slackness conditions. Depending on the extent of feasibility and optimality attained, Stage III applies a penalty function method to improve the obtained primal solution toward a near feasible and optimal solution.

We present computational experience using a set of randomly generated, structured, linear programming problems of the type that might typically arise in the context of discrete optimization.

Acknowledgments

To my parents for their love and affection. I also wish to express my deepest gratitude to all those people who have offered enormous amount of support for my higher studies.

First, my sincere gratitude goes to Dr. Hanif D. Sherali, who has advised and led my research. He has always been and will always be a good influence on my life both academically and personally. I am deeply indebted to both his invaluable professional and personal assistance. I would also like to thank Dr. Sheldon H. Jacobson and Dr. John E. Kobza, who are my committee members for their comments and consideration.

Next, I wish to acknowledge my family, especially my parents, Sri. Ramacharyulu and Smt. Usha, who have provided support and love throughout my life. Without their encouragement, I wouldn't have made it so far. Also, I would like to thank my beloved brother Seshu who has always been very supportive and encouraging. I also owe a lot to my friends, near and dear who have helped me either directly or indirectly in this endeavor.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

This research effort focuses on large-scale linear programming problems that arise in the context of solving various problems such as discrete linear or polynomial, and continuous nonlinear, non-convex programming problems. These problems are typically solved by employing methods such as linearization and branch-and-cut algorithms for the discrete case, and using polyhedral outer-approximation methods for the continuous case. Non-convex problems of this type arise in various contexts in production planning, location-allocation, game theory, economics, and several engineering design applications. The generation of tight linear programming relaxations is very essential for the success of these methods. While solving discrete or continuous non-convex problems using the foregoing types of approaches, one has to repeatedly solve large-scale linear programmming relaxations. Thus, it becomes imperative to employ an efficient method to solve these problems.

It has been amply demonstrated that solving LP relaxations using a simplex-based algorithm, or even an interior-point procedure, can be inadequately slow in comparison with a Lagrangian Relaxation / Lagrangian Dual approach. For example, in many studies, LP relaxations were solved using standard software such as CPLEX and OB1, and it was found that the execution times using these softwares were high when compared with Lagrangian Relaxation approaches (see Adams and Sherali, 1993, and Sherali and Tuncbilek, 1992). However, for a successful use of the Lagrangian approach, firstly, an appropriate formulation of the Lagrangian Dual (LD) has to be constructed (see Fisher, 1981). Sherali and Myers (1989) discuss and test various strategies and provide guidelines for such Lagrangian Dual formulations. Second, an appropriate nondifferentiable optimization technique (NDO) must be employed to solve the Lagrangian Dual (LD) problem.

In the area of mathematical programming, NDO techniques are employed in the contexts of large-scale discrete optimization, exact penalty function methods, Lagrangian duality, and Dantzig-Wolfe decomposition. Most of the theoretical background for NDO approaches can be traced to convex analysis and the study of directional derivatives. Rockafellar (1970) provides the necessary mathematical background, while the optimality theory and algorithms for NDO problems are covered by Demyanov and Vasilev (1985) and Shor (1985).

Some linear programming problems are difficult to solve using classical optimization methods. For such types of problems, we can obtain a lower bound relatively quickly by applying Lagrangian duality and relying on easier-to-solve subproblems obtained by dualizing a set of constraints using this method. It has been demonstrated empirically that subgradient based approaches are the most promising in practice among several alternative solution methods for solving LDs. However, there is still a need for developing efficient and practical NDO algorithms for solving arbitrary Lagrangian dual problems.

To present our approach, consider a linear programming problem of the form

LP: Minimize cx

subject to $\quad A_1x \geq b_1$

$$A_2x = b_2$$

$$x \in X \subseteq \{ x : 0 \leq x \leq u \}$$

where A_1 is an $m_1 \times n$ matrix and A_2 is an $m_2 \times n$ matrix. Let $m = m_1+m_2$. Here, we assume that X has some special structure that can be advantageously exploited, so that linear programming problems over X are easy to solve. We also assume that the problem has been scaled by first performing column / variable scaling so that each variable $x_j \in [0,u]$ where $u\in[1,10]$, and then performing row scaling by dividing each constraint by its average absolute coefficient. The Lagrangian dual for problem LP can be written as :

<u>LD</u> $\quad$ Maximize $\quad \theta(\alpha, \beta)$ $\quad$ (1.1a)

subject to $\quad \alpha \geq 0$, β unrestricted,

where $\theta(\alpha, \beta)$ is evaluated via the <u>Lagrangian subproblem:</u>

<u>LS(α,β)</u> $\quad \theta(\alpha, \beta) = \text{minimum}\{ cx - \alpha (A_1x - b_1) - \beta (A_2x - b_2) : x\in X \}$. $\quad$ (1.1b)

That is,

$$\theta(\alpha, \beta) = (\alpha b_1 + \beta b_2) + \text{minimum}\{(c - \alpha A_1 - \beta A_2) x : x\in X \}. \qquad (1.1c)$$

For convenience, let us denote $\pi = (\alpha,\beta)$, and denote I_1 as the indices of π corresponding to α and I_2 as the remaining indices corresponding to β. It is well known that the objective function $\theta(\pi)$ is concave and piecewise linear, and thus, problem LD can be solved by any nondifferentiable optimization technique. For any given dual solution , $\pi_k = (\alpha_k , \beta_k) \in R^m$, let us define $X_{\pi k}$ as the set of optimal solutions for the subproblem LS (π) . Then, a subgradient g_k of θ at π_k is given by

$$g_k = \begin{bmatrix} b_1 - A_1 x_{\pi_k} \\ b_2 - A_2 x_{\pi_k} \end{bmatrix}$$

for any $x_{\pi k} \in X_{\pi k}$.

Note that in our context, when applying Lagrangian duality, we are not only interested in finding a lower bound for the primal problem but also a primal optimal solution. In general, we cannot obtain primal feasible solutions via subgradient methods for solving LD problems. However, under suitable assumptions, it is known that each accumulation point of a sequence generated by some convex combination of the optimal solutions to the LD subproblems, is a primal optimal solution (see Shor, Larson and Liu, 1988 and Sherali and Choi, 1993).

We propose a Primal-Dual subgradient algorithm that makes use of certain subgradient deflection strategies such as the Modified Gradient Technique (MGT) proposed by Camerini, Fratta, and Maffioli (1975), the Average Direction Strategy (ADS) of Sherali and Ulular (1989), and a new direction strategy called the Modified Average Direction Strategy (M-ADS). In concert with this, we employ judicious step-length strategies, including a new line search method called the Directional Derivative Line Search Method, and also the Quadratic Fit Line Search Method (see Bazaraa, Sherali, and Shetty, 1993). This yields a near optimal solution to LD, and constitutes Stage I of our approach.

In the second stage of the algorithm (Stage II), a sequence of updated primal solutions is generated using some convex combinations of the Lagrangian subproblem solutions. Alternatively, a starting primal optimal solution can be obtained using the complementary slackness conditions. Depending on the extent of feasibility and optimality attained, Stage III then applies a penalty function method to polish the obtained primal solution toward a near feasible and optimal solution. A more detailed discussion of the solution procedure is given below.

1.2 Summary of Various Proposed Algorithms

A Primal-Dual Subgradient Algorithm

Lagrangian Dual (LD) methods are frequently used to derive lower bounds for discrete optimization problems by dualizing the complicating constraints in order to obtain easier to solve subproblems. When applied to large-scale linear programming problems, this yields the exact optimal value. However, the non-differentiability of the Lagrangian dual problems precludes the use of classical optimization methods. The methods for solving such problems can be broadly classified into cutting plane methods and subgradient based methods.

In this research we present a Primal-Dual Subgradient Algorithm that uses a certain combination of subgradient deflection strategies in conjunction with suitable line search methods to solve the Lagrangian dual problem. This algorithm is divided into three stages. Stage I deals with solving the Lagrangian dual problem by using subgradient deflection based methods such as the aforementioned Modified Gradient Technique, the Average Direction Strategy, and a new deflection strategy called the Modified-Average Direction Strategy, in turn. The step-length rules implemented in this regard are -

(a) Directional Derivative Line Search Method : In this method, we start with a prescribed step-length and then determine whether to increase or decrease the step-length value based on the right-hand and left-hand derivative information available at each iteration.

(b) Quadratic Fit Line Search Method : In this method, we find a three point pattern and then fit a curve to find the step-length value at which the maximum of the curve occurs, and continue this process until a suitable stopping criterion is met.

(c) Certain prescribed step-length rules.

A more detailed discussion of the deflection strategies and the line search methods are presented toward the end of this section. As mentioned earlier, following this, we move onto the second stage of the algorithm wherein a sequence of updated primal solutions is generated using some convex combinations of the Lagrangian subproblem solutions. Alternatively, a starting primal optimal solution can be obtained using the complementary slackness conditions. Depending on the extent of feasibility and optimality attained, Stage III then applies a penalty function method to polish the obtained primal solution toward a near feasible and optimal solution.

Deflected Subgradient Directions

Regardless of what step-length rule is implemented, the pure subgradient method has a limit on its computational performance, due to the direction of motion used. The subgradient direction for these problems results in a zig-zagging phenomenon that might cause the procedure to crawl toward optimality. As a tool to overcome this difficulty, the conjugate subgradient concept has been introduced by imitating the conjugate gradient method for the differentiable case. The concept of combining the current subgradient with the previous direction is used as a strategy to deflect the subgradient.

Accordingly, the direction of motion d_k at x_k is computed as

$$d_k = g_k + \Psi_k d_{k-1} \qquad (1.2)$$

where $\Psi_k \geq 0$ is a deflection parameter, g_k is a subgradient of function f at x_k, and d_{k-1} is the previous direction. Then, the new iterate is computed as

$$x_{k+1} = P_X[x_k + \lambda_k d_k] \qquad (1.3)$$

where λ_k is a suitable step-length and $P_X(.)$ is the operator that projects onto the set X.

The concept of deflecting the subgradient direction has been developed in various algorithms such as the Modified Gradient Technique (MGT) proposed by Camerini, Fratta, and Maffioli (1975) and the Average Direction Strategy (ADS) of Sherali and Ulular (1989). These methods are superior in performance to the pure subgradient methods when applied along with a promising step-length rule. Although MGT moves closer to an optimal solution at each iteration, the rate of improvement is hampered by the fact that it often generates the subgradient direction. Furthermore, although ADS has proven to yield an effective and robust scheme for the most part, it merely bisects the angle between the subgradient and the previous direction. We propose a new direction strategy called the Modified Average Direction Strategy (M-ADS) in which the deflection parameter Ψ_k is determined by projecting the optimal direction ($\pi^*-\pi$) onto the space spanned by the current gradient g_k and the previous direction d_{k-1}. An estimate of Ψ_k in (1.2) is hence derived so that this projected direction approximates the optimal direction as closely as possible using (1.2).

Line Search Methods

Subgradient methods are invariably used for solving nondifferentiable optimization (NDO) problems. In the pure subgradient method, the direction of motion is taken as an anti-subgradient direction and some prescribed step-length rules are employed, instead of a usual line search, to generate a sequence of iterates. However, in order to assure convergence, a suitable step-length should be taken so that the iterates get progressively closer to an optimal solution. It has been observed that the decrement in the Euclidean distance to an optimal solution is relatively smaller than the adopted step-length, and moreover, such a process using an anti-subgradient as a direction of motion can generate a zig-zagging path, resulting in a slow convergence.

Step-length rules play an important role in subgradient based approaches, governing not only an ultimate convergence, but also the practical rate of convergence to optimality. In this research, we present a new line search method called the Directional Derivative line search method, which determines a step-length to be taken based on the left-hand and right-hand derivative information available at each iteration. In this method, after detecting an ascent direction, we start with a prescribed step-length and based on the value of right-hand and left-hand derivatives, we ascertain whether to increase or decrease the present step-length value toward the optimal step-length. In addition to the above method, we design and test a quadratic fit line search method. This method fits a quadratic curve through some three suitable points, that satisfy the so-called three point pattern, and the step-length at which the maximum on this curve occurs is determined. This process is continued until a specified stopping criterion is met.

Chapter 2

Literature Review

Optimization problems having convex, but not necessarily differentiable, objective functions are called Nondifferentiable Optimization (NDO) problems. These problems have received wide attention due to their applications in various fields of science, engineering, and economics. They also appear as part of several optimization algorithms belonging to the class of min-max problems, piecewise approximations, dynamic programming problems, and stochastic optimization problems. NDOs also arise in the context of Dantzig-Wolfe decomposition, Lagrangian duality, exact penalty function methods and other mathematical programming approaches. Rockafellar (1970) provides the fundamental mathematical background, while optimality theory and algorithms have been studied by Demyanov and Vasilev (1985) and Shor (1985). Insights into these types of algorithms are provided by Shor (1983), Polyak and Maine (1984), and Zowe (1985).

Our focus in this research is on Lagrangian duality, which is one of the important areas of application of NDO techniques. The concept of Lagrangian duality(LD) was first adopted by Everett (1963) to solve constrained problems as an infinite sequence of unconstrained problems. Falk (1967) presents the use of this concept for nonlinear programming. Following this, Geoffrion (1970, 1971, 1974) provided a significant impetus to this subject through his seminal work on Lagrangian relaxation. For discrete optimization problems, the LD concept was used by Held and Karp (1970, 1971) in the context of solving the traveling salesman problem (TSP). They popularized the use of subgradient optimization for solving LD problems to obtain lower bounds on the primal (min) discrete problem by using this concept.

The important issues that arise in LD approaches are the dualization strategies, solution methods, and primal recovery techniques. In particular, dualization schemes affect the quality of lower bound obtained by solving LD problems due to the presence of duality gap. Related issues have been dealt with by Fisher (1981), Parker and Rardin (1988), Sherali and Myers (1988), and Nemhauser and Wolsey (1989). It is known that subgradient methods are quite promising for solving the dual problem but, in general, the optimal solution of the subproblem is not feasible to the primal problem. Shor (1985) presents a convex combination weighting rule as a function of the step-lengths in a dual subgradient method, that leads to a sequence of convex combinations of the optimal LD subproblem solutions that converges to a primal optimum. Also, Larsson and Liu (1988) presented a similar result using an average weighting scheme. Sherali and Choi (1993) have more recently presented some primal convergence theorems that generalize the previous results, and provide a more flexible choice of admissible step-length rules and convex combination weighting strategies.

There are several methods for solving Lagrangian dual problems. These methods can be broadly classified as cutting plane methods and subgradient methods. Erlenkotter (1978), Fisher et al. (1986), and Fisher and Kedia (1990) have proposed several heuristic approaches that were specific to their problem structures. The subgradient based methods, introduced by Held and Karp (1971) and Held, Wolfe, and Crowder (1974) are applicable to more general classes of problems.

The important issues in the context of subgradient methods are finding a direction of motion, and determining the step-length to be adopted at each iteration. The choice of direction of motion affects the computational performance of the algorithm and choosing the anti-subgradient direction for nondifferentiable (min) problems typically results in a zig-zagging phenomenon. The concept of conjugate gradient based directions was introduced by Hestenes and Stiefel (1952) and Fletcher and Reeves (1959) for differentiable optimization problems. In the nondifferentiable case, this concept has been extended to deflect the subgradient based direction. Some promising deflection algorithms of this type are the Modified Gradient Technique (MGT) of Camerini, Fratta, and Maffioli (1975), the Average Direction Strategy (ADS) of Sherali and Ulular (1989) and the Optimally Deflected Subgradient Algorithm (ODSA) proposed by Sherali and Choi (1993).

It is known that step-length rules play an important role in subgradient based approaches, governing not only an ultimate convergence, but also the practical rate of convergence to optimality. There are several step-length rules available in the literature, but the following are the most widely used in practice and also guarantee convergence to near optimality.

$$\lambda_k = h_k \text{, where } h_k \geq 0,\ \lim_{k\to\infty} h_k = 0, \text{ and } \sum_{k=1}^{\infty} h_k = \infty. \tag{2.1a}$$

$$\lambda_k = h_k / \|d_k\| \text{, where } h_k \geq 0,\ \lim_{k\to\infty} h_k = 0, \text{ and } \sum_{k=1}^{\infty} h_k = \infty. \tag{2.1b}$$

$$\lambda_k = \beta_k \frac{f(x_k) - w}{\|d_k\|^2} \text{, where } 0 < \varepsilon_1 \leq \beta_k \leq \varepsilon_2 < 2, \text{ and w is a target value.} \tag{2.1c}$$

Here d_k is the direction of motion at a current iterate x_k, and λ_k is the prescribed step-length. In the above rules, the direction of motion d_k was taken as $-g_k$, the anti-subgradient of function f at x_k. Note that in our presentation, we have used a general d_k for the sake of convenience in future reference. The rule (2.1a) has been used by Held and Karp (1971) in their well known 1-tree Lagrangian relaxation of the Traveling Salesman Problem. Polyak (1967) provides the general convergence arguments for the rule (2.1b), and Shor uses this in conjunction with generalized gradient concepts. Although the above two rules have good convergence properties, the main problem with these rules lies in their slow convergence behavior in practice.

The rule (2.1c) can be viewed as three different rules depending on the choice of the target value w. Agmon (1954), Eremin (1968), Motzkin and Schoenberg (1954), and Oettli (1972) use the exact target value $w = f^*$, where f^* is the optimal objective function value. Also, Motzkin and Schoenberg (1954) and Oettli (1972) proved the geometric convergence of the above method. However, information regarding the optimal objective value has to be known and hence the rule (2.1c) is not directly implementable. It must be coordinated with some varying target value technique. Polyak (1969) and Held et al. (1974) made recommendations for some specially structured problems for using variants of step-length rule (2.1c). Also, Bazaraa and Sherali (1981) present some rules to choose the target value as a convex combination of a fixed lower bound and the current value, and Kim et al. (1991) use a similar idea and present a variable target value method. But, in these two works, some initial lower bound estimates are assumed to be known. Sherali, Choi and Tuncbilek (1993) have recently developed a new variable target value method which assumes no a priori known bounds on the optimal value, and they show that their process is not only theoretically convergent, but also yields superior performance in practice.

Sen and Sherali (1986) present some convergent primal-dual subgradient algorithms for decomposable convex programs. In their algorithm, a sequence of primal and dual iterates is generated by employing a Lagrangian dual function along with a suitable penalty function that satisfies a specified set of properties. (Several currently popular and classical types of penalty functions satisfy these properties.) Consequently, not only does one obtain both primal and dual optimal solutions, but also one has a natural stopping criterion based on the gap between the penalty and Lagrangian functions. As an extension to the above work, Sherali and Ulular (1989) proposed some algorithms that admit a larger variety of penalty function choices due to less restrictive assumptions. An additional degree of flexibility is also permitted in the design of the Lagrangian dual function.

The algorithm proposed in this thesis consists of three stages. Stage I deals with the solution of the Lagrangian dual problem. Well known subgradient deflection based methods like Modified Gradient Technique (MGT) proposed by Camerini, Fratta and Maffioli (1975) and the Average Direction Strategy (ADS) of Sherali and Ulular (1989) are employed. They are superior in performance to the pure subgradient methods when applied along with a promising step-length rule. In this research, we propose a new direction strategy called the Modified Average Direction Strategy (M-ADS) in which the deflection parameter is determined by projecting the optimal direction onto the space spanned by the current gradient and the previous direction. The optimal direction is approximated as closely as possible, by such a projected direction. The Quadratic Fit Line Search Method which enjoys global convergence under certain assumptions is also employed. In this method, a three point pattern is found and the step-length is given by the maximum of the curve that fits the pattern. A new step-length rule called the Directional Derivative Line Search Method is developed. This method starts with a prescribed step-length and then it is ascertained whether to increase or decrease the step-length value based on the right-hand and left-hand derivative information available at each iteration.

In Stage II of the proposed algorithm, a sequence of updated primal solutions is generated using some convex combinations of the Lagrangian subproblem solutions. In this regard, Sherali and Choi (1993) presented some primal convergence theorems that generalize the results provided by Shor (1985) and Larsson and Liu (1988). These theorems analyze the relationships between the convex combination weights and the step-length rules adopted by the subgradient methods in order to generate primal optimal solutions. Stage III of the algorithm applies a penalty function method to improve the obtained primal solution toward a near feasible and optimal solution.

Chapter 3

Subgradient Deflection Techniques

3.1 Introduction

As the type of function f(x) that we are optimizing is nondifferentiable, the application of classical methods that use gradients and Hessian matrices computed based on finite differences to find the direction of motion may not be successful in leading to an optimal solution. One instance in which such a classical method fails was illustrated by Wolfe (1975) who presented an example in which the application of a standard steepest descent method generates a sequence of iterates that converges to a non-optimal point. The best approach for such nondifferentiable optimization (NDO) problems is to use a subgradient optimization method. There are two types of subgradient methods, the pure subgradient method and the conjugate subgradient method. In the pure subgradient method, the direction of motion is taken as the subgradient direction and some prescribed step length rule is used to generate a sequence of iterates. This can yield a zig-zagging path, resulting in a slow convergence behavior, and thus, the computational performance of the pure subgradient method is limited due to the direction of motion used. To overcome this difficulty, the concept of conjugate subgradient method has been introduced which is similar to conjugate gradient methods for the differentiable case. The conjugate gradient direction for the differentiable case is obtained by combining the current subgradient with the previous direction, while satisfying some conjugacy requirement (see Hestenes and

Steifel, 1952, and Fletcher and Reeves, 1959). This concept has been used in the nondifferentiable case to deflect the subgradient. Accordingly, the direction of motion d_k at x_k is computed as

$$d_k = g_k + \Psi_k \, d_{k-1} \qquad (3.1)$$

where Ψ_k is a deflection parameter, g_k is a subgradient of the function f at x_k and d_{k-1} is the previous direction ($d_0 = 0$, to start with). The new iterate is then computed according to

$$x_{k+1} = P_X [\, x_k + \lambda_k \, d_k \,] \qquad (3.2)$$

where λ_k is a suitable step length and P_X (.) is the operation that projects a given point onto the feasible region X.

The remainder of this chapter presents some subgradient deflection algorithms that can be used with specially designed step-length rules to be discussed in the following chapter. Section 2 deals with the Modified Gradient Technique (MGT) proposed by Camerini et al. (1975), Section 3 presents the Average Direction Strategy of Sherali and Ulular (1989), and a new deflection strategy, called the Modified Average Direction Strategy, is proposed in Section 4.

3.2 Modified Gradient Technique

This deflection technique was proposed by Camerini, Fratta, and Maffioli (1975). In practice, this method was found to be superior to the pure subgradient method when used in concert with a specially designed step-length selection rule. The deflection parameter Ψ_k is computed according to

$$\Psi_k = \begin{cases} \dfrac{-1.5(g_k^t d_{k-1})}{\|d_{k-1}\|^2} & \text{if } g_k^t d_{k-1} < 0, \\ 0 & \text{otherwise.} \end{cases}$$

However, the direction of motion generated by MGT frequently turns out to be selected as simply the subgradient direction itself, especially at the final stages of the procedure, thereby inhibiting the rate of convergence.

3.3 Average Direction Strategy

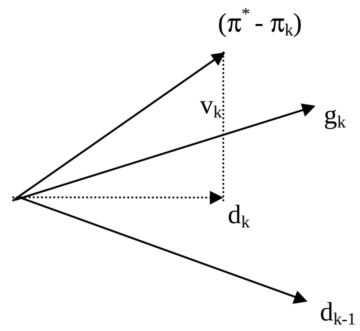
The Average Direction Strategy was proposed by Sherali and Ulular (1989). They found that this particular method performed better than several other methods, including the MGT method of Camerini et al., and different standard conjugate gradient procedures. Here, the deflection parameter Ψ_k is computed according to

$$\Psi_k = \frac{\|g_k\|}{\|d_{k-1}\|} . \tag{3.3}$$

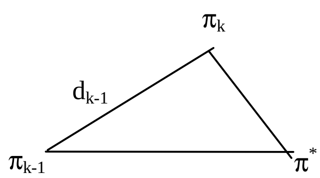
Note that with the choice (3.3), the direction d_k simply bisects the angle between the gradient g_k and the previous direction d_{k-1} and, in this sense, is an average direction.

3.4 Modified Average Direction Strategy

In this section, we present a new subgradient deflection technique called the Modified Average Direction Strategy which determines the deflection parameter ψ_k by projecting the optimal direction onto the plane spanned by the current gradient and the previous direction. That is, ψ_k is determined so that the current direction d_k of (3.1) estimates the projection of the direction (π^* - π_k) (π^* represents the (unknown) optimal solution to the Lagrangian Dual (LD) for the given LP problem), onto the plane spanned by the gradient g_k and the previous direction d_{k-1} as shown in the figure.



In the remainder of this section, the previous direction d_{k-1} will be redefined as (π_k - π_{k-1}), and we will assume that the step lengths are estimated so as to make the current iterate π_k as close as possible to π^*, an optimal dual solution. This is done by trying to make the direction (π^* - π_k) orthogonal to the previous direction d_{k-1} , as shown in the figure below.



Hence, we assume that

$$(\pi^* - \pi_k)^t d_{k-1} = 0. \qquad (3.4)$$

Let us now modify (3.1) as

$$d_k = \psi_1 g_k + \psi_2 d_{k-1}. \qquad (3.5)$$

Note that

$$(\pi^* - \pi_k) = d_k + v_k \text{ , where } \quad v_k^t g_k = v_k^t d_{k-1} = 0. \qquad (3.6)$$

Using the above expressions we get,

$$(\pi^* - \pi_k)^t g_k = \psi_1 \|g_k\|^2 + \psi_2 g_k^t d_{k-1} \qquad (3.7)$$

$$(\pi^* - \pi_k)^t d_{k-1} = \psi_1 g_k^t d_{k-1} + \psi_2 \|d_{k-1}\|^2 . \qquad (3.8)$$

From (3.4) and (3.8), we get

$$\frac{\Psi_2}{\Psi_1} = \frac{-g_k^t d_{k-1}}{\|d_{k-1}\|^2} .$$

Using this in (3.7) yields

$$\psi_1 = \frac{(\pi^* - \pi_k)^t g_k}{\|g_k\|^2 [1 - \cos^2 \phi]} \qquad (3.9)$$

where ϕ is the angle between g_k and d_{k-1}.

Assuming that ϕ is neither 0^0 nor 180^0 and noting by the concavity of the Lagrangian dual function θ that

$(\pi^* - \pi_k)^t g_k \geq \theta(\pi^*) - \theta(\pi_k) > 0$, if π_k is not optimal, we have $\psi_1 > 0$.

Hence, scaling (3.5) by ψ_1, and equating $\Psi_k = \Psi_2 / \Psi_1$, we get the deflection parameter as

$$\psi_k = \frac{-g_k^t d_{k-1}}{\|d_{k-1}\|^2} .$$

This is now used to determine d_k . Note that in order to derive the accompanying step length λ_k to take along d_k so that this would yield an optimal step length from π_k to π_{k+1} with respect to coming closest to π^*, we would like to have, as above,

$$(\pi^* - \pi_{k+1})^t d_k = 0, \qquad \text{where } \pi_{k+1} = \pi_k + \lambda_k d_k .$$

This yields

$$\lambda_k = \frac{(\pi^* - \pi_k)^t d_k}{\left\|d_k\right\|^2} . \tag{3.10}$$

Since π^* is unknown, we use the fact that $d_k = g_k + \Psi_k\, d_{k-1}$ along with (3.4) to deduce that

$$(\pi^* - \pi_k)^t\, d_k = (\pi^* - \pi_k)^t\, g_k \tag{3.11}$$

and we use the inequality that

$$(\pi^* - \pi_k)^t\, g_k \geq \theta(\pi^*) - \theta(\pi_k)$$

to estimate

$$(\pi^* - \pi_k)^t\, g_k = [w_k - \theta(\pi_k)]\, \beta_k \tag{3.12}$$

where w_k is an upper bound on $\theta(\pi^*)$ and β_k is some suitable parameter. Substituting (3.11) and (3.12) into (3.10), we get

$$\lambda_k = \frac{\beta_k(w_k - \theta(\pi_k))}{\left\|d_k\right\|^2} .$$

The block halving strategy of Sherali and Ulular (1989) is adopted to control the parameter β_k in the above expression. For BLOCK=1, 2, 3, the step length parameter β_k for iterates within the corresponding block is taken as β(BLOCK) = 0.75, 0.5, 0.25, respectively.

The prescribed direction finding and step-length strategies are summarized below.

Direction : $d_k = g_k + \Psi_k\, d_{k-1}$, Step-length : $\lambda_k = \dfrac{\beta_k(w_k - \theta(\pi_k))}{\left\|d_k\right\|^2}$ (3.13a)

where,

deflection parameter $\Psi_k = \dfrac{-1.5(g_k^{\,t}\, d_{k-1})}{\left\|d_{k-1}\right\|^2}$ if $g_k^{\,t}\, d_{k-1} < 0$, and (3.13b)

$$0 \qquad \text{otherwise, using strategy MGT,}$$

$$\text{or,} \qquad \Psi_k = \frac{\|g_k\|}{\|d_{k-1}\|} \qquad \text{using strategy ADS,}$$

$$\text{or,} \qquad \Psi_k = \frac{-g_k{}^t d_{k-1}}{\left\|d_{k-1}\right\|^2} \qquad \text{using strategy M-ADS,}$$

and where d_{k-1} is the previous direction ($d_0 = 0$ to start with), g_k is the subgradient at each iteration x_k , β_k is the step length parameter, $\theta(\pi_k)$ is the dual objective function value at iteration k, and w_k is an upper bound on $\theta(\pi^*)$, the optimal dual objective function value.

Chapter 4

Line Search Methods

4.1 Introduction

In subgradient based approaches, step-length rules play an important role in governing not only an ultimate convergence, but they also control the practical rate of convergence to optimality. Typically, a suitable step-length is to be selected, so that the iterates get progressively closer in Euclidean distance to an optimal solution. Several step-length rules and some test results are available in the literature, as discussed earlier in Chapter 2.

This chapter is organized as follows. In Section 2, we present a new line search method which is based on the directional derivative information available at each iteration. Next, in Section 3, we present an alternative Quadratic Fit line search method.

4.2 Directional Derivative Line Search Method.

Suppose that the prescribed step-size strategy yields an increase in the value of the dual objective function, i.e. $\theta(\pi_{k+1}) > \theta(\pi_k)$, so that $d = \pi_{k+1}-\pi_k$ is an ascent direction for the dual objective function θ. The maximum permissible step-length for feasibility in Problem LD can be determined as:

$$\lambda_{max} = \max \{ \lambda : (\pi_{ki} + \lambda d_i) \geq 0 \quad \forall \ i \in I_1 \} \text{ via the minimum ratio test}$$

i.e., $\lambda_{max} = \min \{ \frac{\pi_{ki}}{(-d_i)} : d_i < 0, \ i \in I_1 \}$, (4.1d)

where I_1 corresponds to the nonnegatively restricted components of π. Note that the value of $\lambda_{max} \geq 1$ based on the fact that $\pi_{k+1} = \pi_k + d$ is feasible. We find the step-length $\lambda = \lambda^*$ which solves the following problem :

$$\text{maximize } \{ \theta(\pi_k + \lambda d) : 0 \leq \lambda \leq \lambda_{max} \}. \tag{4.1e}$$

Dropping the subscript k and denoting $\pi_k = \pi = (\alpha,\beta)$ and accordingly, $d = (d_1,d_2)$. From (1.1c), we have the expression for the objective function in (4.1e) :

$$\theta(\pi + \lambda d) = (\alpha b_1+\beta b_2) + v_0\lambda + \min \{ \sum_{j=1}^{n} (\bar{c}_j - \lambda v_j) x_j : 0 \leq x_j \leq u_j \ \forall j \} \tag{4.1f}$$

where,

$$v_0 = d_1 b_1 + d_2 b_2$$

$$\bar{c}_j = c_j - \alpha A_{1j} - \beta A_{2j} \qquad \forall\, j = 1,....,n$$

and $v_j = d_1 A_{1j} + d_2 A_{2j} \qquad \forall\, j = 1,....,n.$

Note that the right hand derivative of $\theta(\pi + \lambda d)$ with respect to λ is given by

$$\frac{d^+}{d\lambda} \theta(\pi + \lambda d) = v_0 - \sum_{j \in J+} v_j u_j \tag{4.1g}$$

where,

$$J^+ = \{ j : [\bar{c}_j - \lambda v_j < 0] \ \text{ or } \ [\bar{c}_j - \lambda v_j = 0 \text{ and } v_j > 0] . \tag{4.1h}$$

The left hand derivative of $\theta(\pi + \lambda d)$ with respect to λ is given by

$$\frac{d^-}{d\lambda} \theta (\pi + \lambda d) = v_0 - \sum_{j \in J-} v_j u_j \tag{4.1j}$$

where,

$$J^- = \{ j : [\bar{c}_j - \lambda v_j < 0] \ \text{ or } \ [\bar{c}_j - \lambda v_j = 0 \text{ and } v_j < 0] . \tag{4.1k}$$

Initialization

The current dual solution $\pi = (\alpha,\beta)$ and the improving direction of motion $d = (d_1,d_2)$ are given. The maximum permissible step-length λ_{max} is obtained using (4.1d). Start with a prescribed step-size of $\lambda = 1$. Determine the index sets of J^+ using (4.1h) and J^- using (4.1k). The right-hand derivative $\frac{d^+\theta}{d\lambda}$ is obtained using (4.1g), and the left-hand derivative $\frac{d^-\theta}{d\lambda}$ is obtained using (4.1j).

Step 1

Compute Δ (the required variation in the step-size value) using the following expressions :

If $\frac{d^+\theta}{d\lambda} > 0$,

$$\Delta = \min_{j=1,..,n} \{\frac{\bar{c}_j - \lambda v_j}{v_j} : \text{(a) } \bar{c}_j - \lambda v_j < 0 \text{ and } v_j < 0 \text{ or (b) } \bar{c}_j - \lambda v_j > 0 \text{ and } v_j > 0\} \quad (4.1m)$$

If $\frac{d^-\theta}{d\lambda} < 0$,

$$\Delta = \min_{j=1,..,n} \{\frac{\bar{c}_j - \lambda v_j}{-v_j} : \text{(a) } \bar{c}_j - \lambda v_j < 0 \text{ and } v_j > 0 \text{ or (b) } \bar{c}_j - \lambda v_j > 0 \text{ and } v_j < 0\} \quad (4.1n)$$

Step 2

If Δ is obtained using (4.1m), then

increment the step-length $\lambda \leftarrow \lambda + \Delta$. (4.1p)

Else, if Δ is obtained using (4.1n), then

decrement the step-length $\lambda \leftarrow \lambda - \Delta$. (4.1q)

If $\lambda \geq \lambda_{max}$, then stop with λ_{max} as the prescribed optimal step-length.

Else, go to Step 3.

Step 3

If λ is obtained using (4.1p), then

update the set $J^+ \leftarrow J^+ \cup$ { indices evaluating Δ via case (b) in (4.1m) } - { indices evaluating Δ via case (a) in (4.1m) } and accordingly, update the right-hand derivative value $\frac{d^+\theta}{d\lambda}$.

Else, if λ is obtained using (4.1q), then

update the set $J^- \leftarrow J^- \cup$ { indices evaluating Δ via case (b) in (4.1n) } - { indices evaluating Δ via case (a) in (4.1n) } and accordingly, update the left-hand derivative value $\frac{d^-\theta}{d\lambda}$.

If $\frac{d^+\theta}{d\lambda} > 0$ or $\frac{d^-\theta}{d\lambda} < 0$, then go to Step 1.

Else, stop with λ as the optimal step-length value.

Line Search For Penalty Function

Suppose, we determine via our prescribed step-size strategy that $h(x_k) < h(x_{k-1})$, so that $d = x_k - x_{k-1}$ is a descent direction for the penalty function h.

We can determine the maximum permissible step length for feasibility as

$$\lambda_{max} = \max \{ \lambda : 0 \leq (x_{(k-1)i} + \lambda\, d_i) \leq u_i , \forall\, i \}.$$

By the minimum ratio test, we get

$$\lambda_{max} = \min \left\{ \frac{x_{(k-1)i}}{-d_i} : d_i < 0 \, , \, \frac{u_i - x_{(k-1)i}}{d_i} : d_i > 0 \right\}. \tag{4.1r}$$

The optimal step-length $\lambda = \lambda^*$ can be obtained by solving

$$\min \{ h(x_{k-1} + \lambda d) : 0 \le \lambda \le \lambda_{max} \}. \tag{4.1s}$$

Dropping the subscript k, we have

$h(x+\lambda d) =$

$$cx + c\lambda d + \sum_{i \in I_1} (\alpha_i^* + \mu) \max\{0, b_{1i} - A_{1i}(x + \lambda d)\} \; + \; \sum_{i \in I_2} (|\beta_i^*| + \mu) |b_{2i} - A_{2i}(x + \lambda d)| \; . \tag{4.1t}$$

The right-hand and left-hand derivatives of $h(x+\lambda d)$ with respect to λ are given by

$$\frac{d^+ h(x + \lambda d)}{d\lambda} = cd - \sum_{i \in I_1^+} (\alpha_i^* + \mu)(A_{1i}d) \; + \; \sum_{i \in I_2^+} (|\beta_i^*| + \mu)|(A_{2i}d)| \; - \; \sum_{i \in I_2^-} (|\beta_i^*| + \mu)|(A_{2i}d)| \tag{4.1u}$$

where,

$I_1^+ = \{ i \in I_1 : b_{1i}-A_{1i}(x+\lambda d) > 0$, or $b_{1i}-A_{1i}(x+\lambda d) = 0$ and $A_{1i}d < 0 \}$ and

$I_2^+ = \{ i \in I_2 : b_{2i}-A_{2i}(x+\lambda d) > 0$ and $A_{2i}d < 0$, or $b_{2i}-A_{2i}(x+\lambda d) < 0$ and $A_{2i}d > 0$,

or $b_{2i}-A_{2i}(x+\lambda d) = 0$ and $A_{2i}d \neq 0 \}$, and

$I_2^- = \{ i \in I_2 : b_{2i}-A_{2i}(x+\lambda d) < 0$ and $A_{2i}d < 0$, or $b_{2i}-A_{2i}(x+\lambda d) > 0$ and $A_{2i}d > 0 \}$.

(4.1v)

$$\frac{d^- h(x + \lambda d)}{d\lambda} = cd - \sum_{i \in \mathcal{I}_1^+} (\alpha_i^* + \mu)(A_{1i}d) \; + \; \sum_{i \in \mathcal{I}_2^+} (|\beta_i^*| + \mu)|(A_{2i}d)| \; - \; \sum_{i \in \mathcal{I}_2^-} (|\beta_i^*| + \mu)|(A_{2i}d)| \tag{4.1w}$$

where,

$I_1^+ = \{ i \in I_1 : b_{1i}-A_{1i}(x+\lambda d) > 0$, or $b_{1i}-A_{1i}(x+\lambda d) = 0$ and $A_{1i}d > 0 \}$ and

I_2^+ = { $i \in I_2$: $b_{2i}-A_{2i}(x+\lambda d) > 0$ and $A_{2i}d < 0$, or $b_{2i}-A_{2i}(x+\lambda d) < 0$ and $A_{2i}d > 0$ }, and

I_2^- = { $i \in I_2$: $b_{2i}-A_{2i}(x+\lambda d) < 0$ and $A_{2i}d < 0$, or $b_{2i}-A_{2i}(x+\lambda d) > 0$ and $A_{2i}d > 0$,

or $b_{2i}-A_{2i}(x+\lambda d) = 0$ and $A_{2i}d \neq 0$ }. (4.1x)

Initialization

The current dual solution $\pi = (\alpha,\beta)$ and primal solution x and the improving direction of motion d are given. The maximum permissible step-length λ_{max} is obtained using (4.1r). Start with a prescribed step-size of $\lambda = 1$. Determine the index sets of I_1^+, I_2^+, I_2^-, using (4.1v) and (4.1x). The right-hand derivative $\frac{d^+h}{d\lambda}$ is obtained using (4.1u), and the left-hand derivative $\frac{d^-h}{d\lambda}$ is obtained using (4.1w).

Step 1

Compute Δ (the required variation in the step-size value) using the following expressions :

If $\frac{d^+h}{d\lambda} < 0$,

$$\Delta = \min \left\{ \frac{b_{1i} - A_{1i}(x + \lambda d)}{A_{1i}d} : \text{(a) } b_{1i}-A_{1i}(x+\lambda d) > 0 \text{ and } A_{1i}d > 0 \right.$$

$$\text{(b) } b_{1i}-A_{1i}(x+\lambda d) < 0 \text{ and } A_{1i}d < 0,$$

$$\frac{b_{2i} - A_{2i}(x + \lambda d)}{A_{2i}d} : \text{(c) } b_{2i}-A_{2i}(x+\lambda d) > 0 \text{ and } A_{2i}d > 0$$

$$\left. \text{(d) } b_{2i}-A_{2i}(x+\lambda d) < 0 \text{ and } A_{2i}d < 0 \right\} \quad (4.1y)$$

If $\dfrac{d^{-}h}{d\lambda} > 0$,

$$\Delta = \min \left\{ \frac{b_1i - A_1i(x + \lambda d)}{-A_{1i}d} : \text{(a) } b_{1i}\text{-}A_{1i}(x+\lambda d) > 0 \text{ and } A_{1i}d < 0 \right.$$

$$\text{(b) } b_{1i}\text{-}A_{1i}(x+\lambda d) < 0 \text{ and } A_{1i}d > 0,$$

$$\frac{b_2i - A_2i(x + \lambda d)}{-A_{2i}d} : \text{(c) } b_{2i}\text{-}A_{2i}(x+\lambda d) > 0 \text{ and } A_{2i}d < 0$$

$$\left. \text{(d) } b_{2i}\text{-}A_{2i}(x+\lambda d) < 0 \text{ and } A_{2i}d > 0 \right\} \qquad (4.1z)$$

Step 2

If Δ is obtained using (4.1y), then

increment the step-length $\lambda \leftarrow \lambda + \Delta$. (4.11y)

Else, if Δ is obtained using (4.1z), then

decrement the step-length $\lambda \leftarrow \lambda - \Delta$. (4.11z)

If $\lambda \geq \lambda_{max}$, then stop with λ_{max} as the prescribed optimal step-length.

Else, go to Step 3.

Step 3

If λ is obtained using (4.11y), then update the sets

$I_1^+ \leftarrow I_1^+$ U {case (b) in (4A)} - {case(a) in (4A)}, $I_2^+ \leftarrow I_2^+$ U {case (d) in (4A)} U {case(c) in (4A)}, $I_2^- \leftarrow I_2^-$ - {case (d) in (4A)} - {case(c) in (4A)}, and accordingly update $\dfrac{d^{+}h}{d\lambda}$.

Else, if λ is obtained using (4.1q), then update the sets

$I_1^+ \leftarrow I_1^+ \cup$ {case (b) in (4B)} - {case(a) in (4B)}, $I_2^+ \leftarrow I_2^+$ - {case (d) in (4B)} - {case(c) in (4B)}, $I_2^- \leftarrow I_2^- \cup$ {case (d) in (4B)} U {case(c) in (4B)}, and accordingly update $\frac{d^- h}{d\lambda}$.

If $\frac{d^+h}{d\lambda} < 0$ or $\frac{d^-h}{d\lambda} > 0$, then go to Step 1.

Else, stop with λ as the optimal step-length value.

4.3 Quadratic Fit Line Search Method

The Quadratic Fit line search method is a very popular technique and it enjoys global convergence under appropriate assumptions such as pseudo-convexity. This method is discussed in detail below.

Step 1

Initialize m = 0, where m is a counter for the number of quadratic fits that have been used. Let $F(\lambda) = \theta(\pi_k + \lambda d_k)$, the dual objective function value at $\pi_k+\lambda d_k$. Compute the value of the dual objective function $\theta(\pi_k) \equiv F(\lambda_1)$ at $\lambda_1 = 0$ and $\theta(\pi_{k+1}) \equiv F(\lambda_2)$ at $\lambda_2 = 1$. Let $\lambda_3 = \min\{2, \lambda_{max}\}$ and compute $F(\lambda_3)$.

Since $F(\lambda_1) < F(\lambda_2)$, if $F(\lambda_2) \geq F(\lambda_3)$, then, set $\lambda^* = (\lambda_1, \lambda_2, \lambda_3)$ as a three point pattern (TPP) to fit a curve (see Bazaraa, Sherali and Shetty, 1993). Go to Step 2.

Else, if $\lambda_3 = \lambda_{max}$, then stop with λ_{max} as the optimal step-size value.

Else, if $\lambda_3 < \lambda_{max}$, then replace λ_1 by λ_2, λ_2 by λ_3, and λ_3 by min $\{2\lambda_2, \lambda_{max}\}$ and repeat Step 1.

Step 2

Fit a curve $c(\lambda)$ through the given three points of λ^*, where $c(\lambda)$ is given by

$$c(\lambda) = \sum_{i=1}^{3} F(\lambda_i)\left[\frac{\prod_{j\neq i}(\lambda-\lambda_j)}{\prod_{j\neq i}(\lambda_i-\lambda_j)}\right].$$

Compute $\bar{\lambda}$, the maximum of the fitted curve, by solving $c'(\lambda) = 0$. Let $F(\bar{\lambda})$ be the dual objective function value at $\bar{\lambda}$. Go to Step 3.

Step 3

Consider the following cases :

Case (a) : $\bar{\lambda} > \lambda_2$

If $F(\bar{\lambda}) \leq F(\lambda_2)$, then set $\lambda^* = (\lambda_1, \lambda_2, \bar{\lambda})$ and go to Step 2.

Else, set $\lambda^* = (\lambda_2, \bar{\lambda}, \lambda_3)$ and go to Step 2.

Case (b) : $\bar{\lambda} < \lambda_2$

If $F(\bar{\lambda}) > F(\lambda_2)$, then set $\lambda^* = (\lambda_1, \bar{\lambda}, \lambda_2)$ and go to Step 2.

Else, set $\lambda^* = (\bar{\lambda}, \lambda_2, \lambda_3)$ and go to Step 2.

Case (c) : $\bar{\lambda} = \lambda_2$

In this case there is no third point to obtain TPP. A new $\bar{\lambda}$ is computed as follows:

If $(\lambda_2 - \lambda_1) \geq (\lambda_3 - \lambda_2)$,

then set $\bar{\lambda} = \dfrac{\lambda_1 + \lambda_2}{2}$ and return to Step 3.

Else set $\bar{\lambda} = \dfrac{\lambda_2 + \lambda_3}{2}$ and return to Step 3.

If at any iteration $F(\lambda_1) = F(\lambda_2) = F(\lambda_3)$ then stop with λ_2 as the optimal step-length.

Else, we have a new TPP.

Increment $m \leftarrow m + 1$.

If $m \geq 2$, then stop with λ_2 as the optimal step-size value.

Else, return to Step 2.

Chapter 5

A Primal-Dual Subgradient Algorithm

4.1 Introduction.

In this chapter, we present a Primal-Dual Subgradient Algorithm that uses a certain combination of subgradient deflection strategies in conjunction with the line search methods presented in Chapter 4 to solve the Lagrangian dual problem. Along with this, it adopts the primal convergence theorems discussed in Chapter 2 to recover a primal solution via a convex combination of dual subproblem solutions. Furthermore, if necessary, this algorithm uses a penalty function method to polish the recovered primal solution toward both a near feasible and near optimal solution . To present this scheme, a linear programming problem, similar to the one discussed in Chapter 1 is considered. The Lagrangian dual (LD) for this problem is constructed as in (1.1a). Many methods, such as cutting plane methods, steepest ascent methods, subgradient-based methods, and heuristic methods have been proposed for solving problem LD. These have been discussed in Chapter 2.

The Primal - Dual Subgradient Algorithm proposed in this chapter consists of three stages. In the first stage, the Lagrangian dual problem is solved by using three different subgradient deflection methods, namely ADS, MGT and M-ADS in conjunction with the line search methods that have been dealt with in Chapter 4. The algorithm moves on to the second stage after a certain number of iterations, or when certain specified stopping criteria such as non-improvements in the objective value over a number of consecutive iterations are met. In the second stage of the algorithm, a sequence of updated primal solutions is generated using some convex combinations of the Lagrangian subproblem solutions. Alternatively, the algorithm can obtain a starting primal solution by applying the complementary slackness conditions. The algorithm then evaluates the extent of feasibility and optimality of the incumbent primal solution, and if necessary, it applies a penalty function method to the primal problem similar to the one developed in Sen and Sherali (1986) and Sherali and Ulular (1989), to further improve the primal solution toward a near feasible and optimal solution in the final, third stage.

This chapter is organized as follows. In Section 2, we provide a combination of several subgradient deflection strategies, in concert with line search methods, for use in the first stage. Section 3 deals with the primal update scheme for the second stage. Following this, Section 4 addresses the third stage that is concerned with the refinement of the primal solution using a penalty function method. Finally, some practical implementation guidelines are discussed in Section 5.

5.2 Stage I : Subgradient Deflection Strategies for Dual Optimization.

It is well known that the computational performance of the pure subgradient method is limited due to the direction of motion used, regardless of the step-length rule implemented. The subgradient direction for the nondifferentiable case results in a zig-zagging phenomenon that might cause the algorithm to crawl toward an optimal solution. The conjugate subgradient or deflected subgradient direction concept, similar to the conjugate gradient direction for the differentiable case, has been used to overcome this problem.

In our context, such a direction of motion d_k at π_k is computed as

$$d_k = g_k + \Psi_k\, d_{k-1} \qquad (5.1)$$

where Ψ_k is a deflection parameter, g_k is a subgradient of θ at π_k, and where d_{k-1} is the previous direction. We take $d_0 \equiv 0$. The new iterate is then computed as

$$(\pi_{k+1}) = P_{\alpha \geq 0}(\pi_k + \lambda_k\, d_k) \qquad (5.2)$$

where λ_k is the prescribed step length.

Among the various subgradient deflection algorithms, we have discussed the Modified Gradient Technique (MGT) proposed by Camerini, Fratta, and Maffioli (1975), and the Average Direction Strategy (ADS) of Sherali and Ulular (1989). In Chapter 3 we have proposed a new strategy called the Modified Average Direction Strategy (M-ADS). In the proposed algorithm, we permit the user to implement any one of these strategies, or use a combination of all three by appropriately shifting from one to another in turn, depending on the algorithmic progress. This approach is addressed below in detail.

Subgradient Deflection Schemes

At each iteration k, the direction of motion is given by

$d_k = g_k + \Psi_k\, d_{k-1}$, where Ψ_k is computed as follows, depending on the value of deflection indicator IND(DEFL).

(1) ADS If IND(DEFL) = 0, compute

$$\Psi_k = \frac{\|g_k\|}{\|d_{k-1}\|}.$$

(2) MGT If IND(DEFL) = 1, compute

$$\Psi_k = \frac{-1.5 g_k{}^t d_{k-1}}{\|d_{k-1}\|^2} \quad \text{if} \quad g_k{}^t d_{k-1} < 0$$

$$0 \quad \text{otherwise.}$$

(3)M-ADS If IND(DEFL) = 2, compute

$$\Psi_k = \frac{-g_k{}^t d_{k-1}}{\|d_{k-1}\|^2} .$$

Step - length rules:

In order to assure convergence, a suitable step-length rule needs to be employed so that the iterates get progressively closer to an optimal solution. As discussed earlier, the step-length rule based on a target value w and some positive parameter β, performs well in practice and is given by

$$\lambda_k = \beta \frac{w - \theta(\pi_k)}{\|d_k\|^2} . \qquad (5.3)$$

In the proposed algorithm, the target value w is chosen as the minimum of a fixed upper bound and an estimated target value. Also, two line search methods have been proposed and the user can specify which line search method is to be implemented in advance, if any. (Note that when no line search is used, the algorithm simply adopts the prescribed step-size given by (5.3)). The first line search method is based on the use of directional derivatives. We start with a prescribed step-size value and based on the directional derivative we decide whether to increase or decrease the step-size. This method was described in detail in Chapter 4. The second line search method is the Quadratic Fit method in which a three point pattern is first obtained based on the objective function values. Then a quadratic curve is fit using the three points and the maximum of the curve is found. Using this new point another three point pattern is obtained and a similar procedure is followed. The process is terminated after a certain stopping criterion is met or after some maximum number of quadratic interpolations has been performed (usually, just one or two). This method was also described in detail in Chapter 4.

Algorithm for Stage I (Dual Optimization)

The projection operators used in this algorithm are defined as:

$P_{F(\pi_k)}$ (.) projects a direction onto the feasible directions space at π_k.

$P_{\alpha \geq 0}$(.) projects an iterate (.) onto the feasible region defined by LD.

$P_{F(\pi_k)}$ (g_k) has components given by	0	if $g_{ki} < 0$ and $\alpha_{ki} = 0$ for $i \in I_1$
	g_{ki}	otherwise.
$P_{\alpha \geq 0}(\pi)$ has components given by	0	if $\pi_i < 0$ for $i \in I_1$
	π_i	otherwise.

Initialization: Put the iteration counter k = 1, select a starting solution $\pi_1 = 0$ and set $d_0 = 0$. Evaluate the dual objective function value $\theta(\pi_1)$ and hence determine a subgradient g_1 of θ at π_1. Set π_1 as the best known dual solution π^*, and $\theta(\pi_1)$ as the best known functional value z^*. Let $g_1^{proj} = P_{F(\pi_1)}(g_1)$ and let $g^* = g_1^{proj}$, and compute $\|g_1^{proj}\|$. Update the upper bound value using UB $\leftarrow$ min{UB, {max(cx) - min(cx) : $x \in X$}}.

If $(g^* \pi^* \neq z^*$ and $\|\pi^*\| \neq 0)$

$$\text{let } K = \frac{2\{z^* - g^*\pi^*\}}{\|\pi^*\|^2},$$

and otherwise, let K = 1.

Compute the initial target value using (with k=1)

$$w_k = \min\{ UB, \theta_k + \frac{\|g_k^{proj}\|^2}{2K} \}, \qquad (5.4)$$

where $\theta_k = \theta(\pi_k)$.

Set $\delta_w = 0$ if w_1 is given by the first term in (5.4); else, set $\delta_w = 1$. Compute the target proximity measure as

$$\Delta = 0.15(w_k - z^*). \qquad (5.5)$$

The step-length parameter β is controlled using three blocks in the block halving strategy of Sherali and Ulular (1989). Set the value of β(BLOCK) = 0.95, 0.5 and 0.25 for BLOCK = 1, 2, and 3, respectively. Currently, put BLOCK = 1 and β = 0.95. Set the maximum iteration limit $k_{max} = \max\{200, m\}$ and the maximum block iteration limit $k_{Bmax} = \left\lfloor \frac{k_{max}}{3} \right\rfloor$. Select the deflection scheme to be used by setting the value of

IND(DEFL) =	0	ADS method
	1	MGT method
	2	M-ADS method
	3	Switching between the above is desired.

If IND(DEFL) = 3, then set IND(SWITCH) = 1 and IND(DEFL) = 0 to begin with. Otherwise set IND(SWITCH) = 0.

Set the value of IND(LS) = 1	if line search is to be performed in Stage I
0	otherwise.

Put the consecutive failure counter $\gamma = 0$, RESET=1, and the target increase parameter $n_w = 1$. Set the threshold failure limit $\bar{\gamma} = 20$ and set $n_f = 0$, which is the number of resets made within the block.

Step 1. Compute g_k^{proj} (if not already available). If $\|g_k^{proj}\| \leq 10^{-6}$, then transfer to Stage II. Compute the direction of motion d_k using (5.1) if RESET = 0, and otherwise, let $d_k = g_k$. Compute $\|d_k\|$. If $\|d_k\| \leq 10^{-6}$, then d_k must be given by (5.1), and so, put $d_k = g_k$. Compute the step-length λ_k using (5.3). Update the dual solution π_{k+1} using (5.2), and evaluate the dual objective function value $\theta(\pi_{k+1})$ and find a subgradient g_{k+1} of θ at π_{k+1} .Obtain g_{k+1}^{proj} using $P_{F(\pi_{K+1})}(g_{k+1})$. Increment k by one.

If $k > k_{Bmax}$, go to Step 5.

If $\theta_k > z^*$, go to Step 3.

Else, let $d_{k-1} = (\pi_K - \pi_{K-1})$ and proceed to Step 2.

Step 2 (Failure iteration)

Increment $\gamma \leftarrow \gamma+1$.

If $\gamma < \bar{\gamma}$,

put RESET = 0 and return to Step 1.

Else,

set $\gamma = 0$, $\bar{\gamma} = \min\{\bar{\gamma} + 10, 50\}$, $\pi_k = \pi^*$, $\theta_k = z^*$, $g_k = g^*$,

RESET = 1 , $\beta \leftarrow \beta / 2$, and $n_f \leftarrow n_f + 1$.

If $n_f > 10$ and BLOCK ≤ 2 , go to Step 5.

Else, if IND(SWITCH) = 1, replace

IND(DEFL) by (IND(DEFL)+1)mod 3, and return to Step 1.

STEP 3 : (Line search)

Put $d_{k-1} = (\pi_K - \pi_{K-1})$.

If IND(LS) = 0, go to Step 4, else,

find $\lambda^* \approx \max \{ \theta (\pi_{k-1} + \lambda d_{k-1}) : 0 \le \lambda \le \lambda_{max} \}$,

and set $\pi_k = \pi_{k-1} + \lambda^* d_{k-1}$.

Compute θ_k , and g_k ,

and put

$$d_{k-1} = (\pi_K - \pi_{K-1}).$$

Proceed to Step 4.

STEP 4 : (Success iteration)

Put $\gamma = 0$, $z^* = \theta_k$, $\pi^* = \pi_k$, and $g^* = g_k$.

If $z^* \ge w_k - \Delta$, then if $\delta_w = 0$, go to Stage II, and

if $\delta_w = 1$, put $n_w \leftarrow n_w + 1$ and update w_k as $z^* + n_w \Delta$.

If w_k exceeds UB, then put w_k = UB and set $\delta_w = 0$.

Put RESET = 0 and return to Step 1.

STEP 5 (New block initialization or termination)

If $\theta_k > z^*$,

put $z^* = \theta_k$, $\pi^* = \pi_k$, and $g^* = g_k$.

If BLOCK = 3 , transfer to Stage II.

Else, set

BLOCK $\leftarrow$BLOCK + 1 .

Put $k_{Bmax} = \lfloor (\text{BLOCK}) k_{max} / 3 \rfloor$, $\pi_k = \pi^*$, $\theta_k = z^*$, $g_k = g^*$, and RESET = 1. Update w_k and Δ using (5.4) and (5.5), respectively. Set $\beta \equiv \beta(\text{BLOCK})$, $\gamma = 0$, $\bar{\gamma} = 10$, $n_f = 0$, n_w = 1, and return to Step 1.

5.3 Stage II : Recovery of Primal Solutions

This section deals with a procedure for recovering primal solutions via the dual subgradient based method. Note that solving the dual subproblem might not provide a primal optimal, or even a feasible, solution. In Chapter 2, we have already discussed certain generic primal convergence theorems that deal with the recovery of primal optimal solutions via a suitable convex combination of the dual subproblem solutions. In the second stage of the algorithm, we not only update the dual solutions but also recover primal solutions. In this stage, we adopt the pure subgradient method along with an average weighting primal recovery scheme. The algorithmic statement for Stage II is presented below.

<u>Algorithm for Stage II (Dual Optimization and Primal Recovery)</u>

Initialization Select the maximum block iteration limit $k_{Bmax} = \lceil 50 [2\text{-}e^{-(kmax-200)}] \rceil$ (Note that the maximum iteration limit k_{max} is the same as the one used in Stage I.) Initialize the dual solution $\pi_1 = \pi^*$, the primal solution $x_1 = x_{\pi 1}$, and the direction of motion $d_k = g^*$. (Note that π^* , $x_{\pi 1}$ and g^* are obtained from Stage I.) Calculate the target value as

$$w_1 = z^* + 0.05 |z^*| \quad \text{if} \quad |z^*| \geq 0.01$$

$$\text{and} \quad w_1 = z^* + \min \{ 1 , \|g^{*proj}\|^2 / 2 \} \text{ otherwise.}$$

Set the iteration counter k = 1. Compute the target proximity measure using $\Delta = 0.001 (w_1 - z^*)$. Put the consecutive failure counter $\gamma = 0$ and target increase parameter $n_w = 1$. Set the threshold failure limit $\bar{\gamma} = 10$.

Step 1 (Dual Optimization) Compute the step length λ_k and the new iterate π_{k+1} using (5.3) and (5.2), respectively.

Step 2 Evaluate the dual objective function value $\theta(\pi_{k+1})$, subproblem optimal solution $x_{\pi_{k+1}}$ and the subgradient g_{k+1} at the new iterate. Find a new primal solution using

$$x_{k+1} = \frac{k}{k+1} x_k + \frac{1}{k+1} x_{\pi_{k+1}}$$

and increment k by one.

If $\theta(\pi_k) > z^*$, then set $\pi^* = \pi_k$ and $z^* = \theta(\pi_k)$ and reset γ to zero. If $k > k_{Bmax}$ then transfer to Stage III with $x^* = x_k$. If $z^* \geq (w_{k-1} - \Delta)$ then increment n_w by one and update the target value using $w_k \leftarrow (z^* + n_w \Delta)$ and return to Step 1.

Else, if $\theta(\pi_k) \leq z^*$ and if $k \leq k_{Bmax}$, increment γ by one. If $\gamma < \bar{\gamma}$ then return to Step 1. Else, halve the parameter β, reset γ to zero and return to Step 1.

5.4 Stage III : Primal Penalty Function Method

It has to be noted that the primal convergence theorems discussed in Chapter 2 guarantee only that every accumulation point of the sequence of the primal iterates $\{x_k\}$ generated at Stage II is an optimal solution to the primal problem. Hence, a primal solution obtained at some maximum allowed iteration limit could be neither optimal nor even feasible to the primal problem. Therefore, there is a need to further polish the primal solution to achieve near feasibility and optimality. For this purpose, we adopt the penalty function method of Sen and Sherali (1986) and Sherali and Ulular (1989) at the final stage of the algorithm.

For a given dual solution $\pi^* = (\alpha^*, \beta^*)$, let us define the penalty function as

$$h(x) = c^t x + \sum_{i\in I_1} (\alpha_i^* + \mu) \max\{0, b_{1i} - A_{1i}x\} + \sum_{i\in I_2} (|\beta_i^*| + \mu)|b_{2i} - A_{2i}x| \qquad (5.5)$$

where,

$$\mu = \text{maximum } \{\alpha_i^* , i \in I_1 , |\beta_i^*| , i \in I_2 \} + 1.$$

A subgradient ξ of h at any x is given by

$$\xi = c - \sum_{i\in I_{1+}} (\alpha_i^* + \mu) A_{1i} - \sum_{i\in I_{2+}} (|\beta_i^*| + \mu) \mathbf{A}_{2i} + \sum_{i\in I_{2-}} (|\beta_i^*| + \mu) \mathbf{A}_{2i} . \qquad (5.6)$$

where,

$I_1^+ = \{ i \in I_1 : (b_{1i} - A_{1i} x) > 0 \}$,

$I_2^+ = \{ i \in I_2 : (b_{2i} - A_{2i} x) > 0 \}$, and

$I_2^- = \{ i \in I_2 : (b_{2i} - A_{2i} x) < 0 \}$.

For any $x_k \in X$ at some iteration k of an algorithmic process , we have

$$h(x_k) \geq c^t x_k + \sum_{i\in I_1} (\alpha_i^*)(b_{1i} - A_{1i}x) + \sum_{i\in I_2} |\beta_i^*||b_{2i} - A_{2i}x| \geq \theta(\alpha^* , \beta^*).$$

Sen and Sherali (1986), and Sherali and Ulular (1989) have proposed some theorems for terminating the algorithm with a near feasible and optimal solution. Similar conditions are presented below for (5.5)

<u>Proposition:</u>

Suppose that the penalty parameter μ is selected as $\mu = M + \Delta$ where

$M = \text{maximum } \{\alpha_i^* , i \in I_1 , |\beta_i^*| , i \in I_2 \}$ and $\Delta > 0$ is some constant.

If $\theta(\alpha^* , \beta^*) \geq h(x_k) - \varepsilon$, for some $x_k \in X$ and for some $\varepsilon > 0$, then we have

$b_{1i} - A_{1i} x_k \leq \varepsilon / \mu$ for all $i \in I_1$

$|b_{2i} - A_{2i}x_k| \leq \varepsilon / \mu$ for all $i \in I_2$ and

$\alpha_i^* |b_{1i} - A_{1i}x_k| \leq \varepsilon$ for all $i \in I_1$

$$|\beta_i^*||b_{2i} - A_{2i}x_k| \le \varepsilon \qquad \text{for all } i \in I_2 .$$

PROOF : By assumption, we have,

$$c^t x_k + \alpha^* (b_1 - A_1 x_k) + \beta^* (b_2 - A_2 x_k) \ge \theta(\alpha^*, \beta^*) \ge h(x_k) - \varepsilon.$$

Thus,

$$\alpha^* (b_1 - A_1 x_k) + \beta^* (b_2 - A_2 x_k) \ge \sum_{i\in I_1} (\alpha_i^* + \mu)\max\{0, b_{1i} - A_{1i}x\} + \sum_{i\in I_2} (|\beta_i^*| + \mu)|b_{2i} - A_{2i}x| - \varepsilon \tag{5.7}$$

$$\ge \alpha^* (b_1 - A_1 x_k) + \beta^* (b_2 - A_2 x_k) + \mu \left[\sum_{i\in I_1} \max\{0, b_{1i} - A_{1i}x\} + \sum_{i\in I_2} |b_{2i} - A_{2i}x|\right] - \varepsilon .$$

Hence ,

$$\mu \left[\sum_{i\in I_1} \max\{0, b_{1i} - A_{1i}x\} + \sum_{i\in I_2} |b_{2i} - A_{2i}x|\right] \le \varepsilon .$$

That is,

$$\left[\sum_{i\in I_1} \max\{0, b_{1i} - A_{1i}x\} + \sum_{i\in I_2} |b_{2i} - A_{2i}x|\right] \le \varepsilon / \mu .$$

Therefore,

$$b_{1i} - A_{1i} x_k \le \varepsilon / \mu \qquad \text{for all } i \in I_1 \quad \text{and}$$

$$|b_{2i} - A_{2i}x_k| \le \varepsilon / \mu \qquad \text{for all } i \in I_2 .$$

It is clear then that for equality constraints

$$|\beta_i^*||b_{2i} - A_{2i}x_k| \le \varepsilon \qquad \text{for all } i \in I_2 .$$

For inequality constraints , from (5.7) , we have ,

$$\alpha^* (b_1 - A_1 x_k) \ge \sum_{i\in I_1} (\alpha_i^* + \mu)\max\{0, b_{1i} - A_{1i}x_k\} - \varepsilon$$

$$= \sum_{i\in I_1, i\ne j} (\alpha_i^* + \mu)\max\{0, b_{1i} - A_{1i}x_k\} + [\alpha_j^* + \mu]\max\{0 , b_{1j} - A_{1j}x_k\} - \varepsilon$$

$$\ge \sum_{i\in I_1, i\ne j} (\alpha_i^*)(b_{1i} - A_{1i}x_k) + [\alpha_j^* + \mu]\max\{0 , b_{1j} - A_{1j}x_k \} - \varepsilon .$$

Hence, for any $j \in I_1$,

$$\alpha_j^* (b_{1j} - A_{1j}x_k) \geq [\alpha_j^* + \mu]\max\{ 0 , b_{1j} - A_{1j}x_k \} - \varepsilon \tag{5.8}$$

$$\geq - \varepsilon.$$

Therefore,

$$\alpha_j^* (b_{1j} - A_{1j}x_k) \geq - \varepsilon . \tag{5.9}$$

Moreover , if $(b_{1j} - A_{1j}x_k) \geq 0$, then from (5.8) , we have ,

$$\alpha_j^* (b_{1j} - A_{1j}x_k) \geq [\alpha_j^* + \mu]\max\{ 0 , b_{1j} - A_{1j}x_k \}- \varepsilon$$

$$\geq 2 \alpha_j^* (b_{1j} - A_{1j}x_k) - \varepsilon .$$

Hence , regardless of the sign on $(b_{1j} - A_{1j}x_k)$, we have,

$$\alpha_j^* (b_{1j} - A_{1j}x_k) \leq \varepsilon . \tag{5.10}$$

Therefore, from (5.9) and (5.10) , we get ,

$$\alpha_j^* \left| b_{1j} - A_{1j}x_k \right| \leq \varepsilon \qquad \text{for all } j \in I_1 .$$

This completes the proof.

Based on the above conditions the algorithm can be stated as follows.

Algorithm for Stage III (Primal Penalty Function Method)

The projection operators used in this algorithm are defined as follows :

$P_{F(x_k)}(.)$ projects a direction onto the feasible directions space at x_k.

Assuming that $X = \{ x : 0 \leq x \leq u \}$,

$P_{F(x_k)}[d]$ has components given by $\quad 0 \quad$ if $[(x_k)_j = 0$ and $d_j < 0]$ or $[(x_k)_j = u_j$ and $d_j > 0]$

$\qquad d_j \qquad$ otherwise.

Define $\xi_k^{proj} = P_{F(x_k)} (\xi_k)$ for any subgradient ξ_k of h at x_k .

$P_X (.)$ projects an iterate onto the region feasible to (X).

$P_X(x)$ has component j given by	0	if $x_j < 0$	
	u_j	if $x_j > u_j$	
	x_j	otherwise	$\forall$ components j.

Initialization

If IND(LUB) = 1, apply the complementary slackness conditions to determine a starting primal solution x^* as opposed to using the solution x* obtained via Stage II. Compute the maximum iteration limit as $k_{max} = 200 + 1800\ [\ 1 - e^{-\max\{0\ ,\ n\ -\ 500\}}\ ]$ and fix the target value at $w = z^*$. Set the starting solution $x_0 = x^*$, and the iteration counter k = 0.

The step-length parameter β is defined as in Stage I of the algorithm. Set the value of β(BLOCK) = 0.75, 0.5 and 0.25 for BLOCK = 1, 2, and 3, respectively. Currently, put BLOCK = 1 and β = 0.75. Select the deflection scheme to be used by setting the value of

IND(DEFL) =	0	ADS method
	1	MGT method
	2	M-ADS method
	3	Switching between the above is desired.

If IND(DEFL) = 3, then set IND(SWITCH) = 1 and IND(DEFL) = 0 to begin with. Otherwise set IND(SWITCH) = 0.

Set the value of IND(LS) = 1	if a line search is to be performed in Stage III
0	otherwise.

Put the consecutive failure counter γ = 0, RESET=1, and the target increase parameter n_w = 1. Set the threshold failure limit $\bar{\gamma}$ = 10 and set n_f = 0, which is the number of resets made within any block.

Assess the initial solution by evaluating

$RFEAS(x_k)$ = Average over constraints of

{ Violation in constraint i } / { # of non-zeroes in constraint i } and

$$RGAP(x_k) = \frac{(c^t x^* - z^*)}{\max\{|z^*|, 1\}} .$$

If $RFEAS(x_k)$ and $RGAP(x_k) \leq 0.05$, then stop with x_0 as the prescribed near optimal solution.

Else, compute the penalty function value $h(x_0)$ and determine a subgradient ξ_0 of h at x_0 using (5.5) and (5.6). Set $h(x_0)$ as the best known objective function value h* and let $\xi^* = \xi_0$.

STEP 1 : If $\|\xi_k^{proj}\| \leq 10^{-6}$, then stop. Else, compute the new direction

$$d_k = -\xi_k + \Psi_k d_{k-1}$$

where

Ψ_k =	0	if RESET = 1 or $\|d_{k-1}\| \leq 10^{-6}$
	ADS value	if RESET = 0 and IND(DEFL) = 0
	M-ADS value	if RESET = 0 and IND(DEFL) = 1
	MGT value	if RESET = 0 and IND(DEFL) = 2.

Compute $\|d_k\|$.

If $\|d_k\| \leq 10^{-6}$,

put $d_k = -\xi_k$.

Compute the step-length

$$\lambda_k = \frac{\beta[h_k - w]}{\|d_k\|^2}.$$

Determine the new iterate as

$$x_{k+1} = P_X [x_k + \lambda_k d_k]$$

and increment k by 1.

Find h_k , ξ_k , and ξ_k^{proj}.

If $k > k_{Bmax}$, go to Step 5 .

If $h_k < h^*$, go to Step 3 .

Else,

let $d_{k-1} = (x_k - x_{k-1})$ and proceed to Step 2.

STEP 2 : (Failure iteration)

Increment $\gamma \rightarrow \gamma + 1$.

If $\gamma < \bar{\gamma}$

put RESET = 0 and return to Step 1.

Else,

set $\gamma = 0$, $\bar{\gamma} = \min \{20 , \bar{\gamma} + 1 \}$, $x_k = x^*$, $h_k = h^*$, $\xi_k = \xi^*$,

RESET = 1 , $\beta \leftarrow \beta / 2$ and $n_f \leftarrow n_f + 1$.

If $n_f > 10$ and BLOCK ≤ 2 , go to Step 5.

Else, if IND(SWITCH) = 1, replace

IND(DEFL) by (IND(DEFL)+1)mod 3, and return to Step 1.

STEP 3 : (Line search)

Put $d_{k-1} = (x_k - x_{k-1})$.

If IND(LS) = 0, go to Step 4. Else, compute

$$\lambda^* = \min \{ h (x_{k-1} + \lambda d) : 0 \leq \lambda \leq \lambda_{max} \},$$

where $\lambda_{max} = \min \{ \frac{x_{(k-1)i}}{(-d_i)} : d_i < 0 , \frac{u_i - x_{(k-1)i}}{d_i} : d_i > 0 \}$ as in chapter 4.

Set $\quad x_k = x_{k-1} + \lambda^* d_{k-1}$.

Compute $\quad h_k$, ξ_k, and ξ_k^{proj},

and put

$$d_{k-1} = (x_k - x_{k-1}) .$$

Proceed to Step 4.

STEP 4 : (Success iteration)

Put $\quad \gamma = 0$, $x^* = x_k$, $h^* = h_k$, and $\xi^* = \xi_k$.

If $\quad RFEAS(x_k) \leq 0.05$ and $RGAP(x_k) \leq 0.05$,

stop with x_k as a near optimum.

Else,

put RESET = 0 and return to Step 1.

STEP 5 :

If $\quad h_k < h^*$,

put $\quad x^* = x_k$, $h^* = h_k$, and $\xi^* = \xi_k$.

If $\quad$ BLOCK = 3 , stop with x^* as a near optimum.

Else, set

BLOCK $\leftarrow$ BLOCK + 1 ,

put $k_{Bmax} = \lfloor (BLOCK) k_{max} / 3 \rfloor$, $x_k = x^*$, $h_k = h^*$, $\xi_k = \xi^*$, RESET = 1,

$\beta \equiv \beta(BLOCK)$, $\gamma = 0$, $\bar{\gamma} = 10$, $n_f = 0$, and return to Step 1.

5.5 Implementation and Computational Experience

For computational purposes, we have randomly generated a set of test problem (see Rosen and Suzuki, 1965). Components for the primal solutions x are generated uniformly on [0,1]. In particular we generate $m_2 + \lfloor m_1/2 \rfloor$ of these components on (0,1) and the remainder of them are set to either 0 or 1. Next, we generate the dual solutions α associated with the inequality constraints uniformly on [0,5]. Again $m_1 - \lfloor m_1/2 \rfloor$ components are specifically generated in this manner, while the remaining components are set to zero. The dual variables associated with the equality constraints (β) are generated uniformly on [-10,10], on third of these being put to zero. The coefficients matrix A_1 for the inequality constraints is about 5% dense having nonzero elements ± 1. The equality constraints coefficient matrix A_2 is generated in a similar fashion. The right-hand side for the inequality constraints b_1 is set to A_1 with a unit being subtracted if the components value of α is zero. The right hand side vector for the equality constraints b_2 is set to A_2x. In a similar fashion we generate the objective coefficients c as $A^t{}_1\alpha + A^t{}_2\beta$, with a unit being added if the corresponding primal solution x components is zero. Finally, we normalize the coefficients of the constraints and also the right-hand side values by dividing each of the components by the norm of the vector (A_1,b_1) or (A_2,b_2) as the case may be.

Computational results for Stages I-II-III are presented in Table 5.1 through Table 5.5. Each of the following tables lists the problem size in terms of the number of variables n, number of equality constraints m1 and the number of inequality constraints m2. The number of iterations for which the algorithm runs is also indicated. The tables also include results for various line search and deflection strategies employed for each of the test cases. All the test problems were also solved using CPLEX for comparison purposes. Solution times and the solutions obtained by CPLEX are also included in the table. (It has to be noted that the computational time of our algorithm is considerably higher because it does not make use of sparsity of the matrix and the computations are not performed in packed-form as in CPLEX.)

The Lagrangian dual problem is solved using three different subgradient deflection methods, namely ADS, MGT and M-ADS in conjunction with the Directional Derivative line search method and also using the Quadratic Fit line search method. At the end of Stage I, we obtain a good lower bound for the optimal objective value.

In Stage II, we generate a sequence of updated primal solutions using some convex combinations of the Lagrangian subproblem solutions. The main objective of this stage is to find a primal solution that is near-feasible and optimal. At the start of Stage III, we assess the initial solution in terms of feasibility and also the amount of duality gap. Depending on this information we proceed to apply a penalty function method to the primal problem.

The dual problems for Stage I were solved as in Section 5.2. It can be observed from the results in tables 5.1 through 5.5 that the bound obtained at the end of Stage I is not a very favorable one. For all the test cases the improvement in the objective value from the initial objective value is not very significant. In order to obtain a better bound we have tried the Variable Target Value Method (see Sherali, Choi and Tuncbilek, 1993) to determine the step-length at each iteration. There was no significant improvement in the solution value and hence this method was abandoned. Our attempts to get a better bound by projecting the dual values onto some pre-determined boxes of various sizes was not successful. It has to be observed that we have obtained a very good bound by projecting the dual variables onto the boxes centered around the optimal dual values obtained from CPLEX. A box size of two was used for the above purpose. One other method that we have implemented is the "Averaging Strategy" for compiling the primal solution. Whenever the reduced cost is zero, then the primal solution is taken as the average of the current primal solution instead of setting it at either the upper bound or the lower bound zero. The idea behind this strategy was that the subsequent subgradients, that are computed via the Lagrangian subproblem's primal solution would have a stabilizing/averaging effect that might be stronger than the deflected subgradient strategy. The same procedure was also used by the Stage II (recovery of primal solutions) to improve the dual solution value without much success. Other attempts to change parameters such as the threshold failure limit $\bar{\gamma}$, density of the input problem, the step length parameter β etc., have not yielded any fruitful results.

In Stage II, there is an insignificant amount of improvement in the objective value as our goal at this stage was to obtain a sequence of updated primal solutions using some convex combinations of the Lagrangian subproblem solutions. In order to reduce the burden on Stage III, we have attempted to improve the objective function value by employing the "averaging strategy" as discussed earlier. We ran this process for one thousand iterations but without any significant improvement and hence this idea was abandoned.

Finally Stage III, which employs a penalty function method to the primal problem has done reasonably well when compared to the previous stages. This can be observed from the results compiled in the tables below. Throughout these test runs we have observed that the performance of the primal penalty function method depends very much on the quality of the dual incumbent solution, since this influences the definition of the penalty function as well as the determination of the step-length. Consequently, the final solution value we have obtained was affected. It can also be observed from the results that as the problem size increases, we have obtained better solution values at the end of Stage III. As the value of the penalty parameter μ is dependent on the incumbent dual solution values, it is possible that this value is not sufficiently large enough to force the solution value to near optimality.

To summarize, in this research effort, we have proposed a primal-dual subgradient algorithm that enjoys the global convergence property, and produces a feasible and acceptable solutions in a reasonably efficient manner. However, we remark that there is still a need for developing further improved Lagrangian dual solution procedures, accompanied with improved convex combination weighting rules used to recover a primal optimal solution.

Table
5.1

Variables n = 100
Inequality constraints m1 =
40
Equality constraints m2 =
10
Iterations = 1000
CPLEX solution = 10.68
CPLEX CPU seconds =
0.26
CPLEX Iterations = 692

Line Search	Deflection Strategy	Optimal objective	Initial objective	Stage 1	Stage 2	Stage 3	Total CPU sec
	No deflection	10.68	-33.36	-18.96	-17.99	10.22	1.01
	ADS	10.68	-33.36	-24.27	-23.03	7.54	1.02
None	MGT	10.68	-33.36	-25.93	-24.6	6.12	1.01
	MADS	10.68	-33.36	-19.09	-18.12	10.22	1.04
	Switch	10.68	-33.36	-18.96	-17.99	10.22	1.03
	No deflection	10.68	-33.36	-21.81	-20.69	9.19	1.03
	ADS	10.68	-33.36	-23.91	-22.69	7.93	1.02
Directional Derivative	MGT	10.68	-33.36	-24.88	-23.61	6.85	1.04
	MADS	10.68	-33.36	-21.81	-20.69	9.23	1.02
	Switch	10.68	-33.36	-18.29	-17.36	10.33	1.01
	No deflection	10.68	-33.36	-23.16	-21.98	8.68	1.01
	ADS	10.68	-33.36	-23.16	-21.98	8.65	1.02
Quadratic Fit	MGT	10.68	-33.36	-23.16	-21.98	8.93	1.01
	MADS	10.68	-33.36	-23.16	-21.98	8.67	1.02
	Switch	10.68	-33.36	-23.16	-21.98	8.69	1.02

Table
5.2

Variables n = 250
Inequality constraints m1
=100
Equality constraints m2 =
25
Iterations = 1000
CPLEX solution = -10.704
CPLEX CPU seconds =
0.42
CPLEX Iterations = 708

Line Search	Deflection Strategy	Optimal objective	Initial objective	Stage 1	Stage 2	Stage 3	Total CPU sec
	No deflection	-10.704	-99.278	-42.58	-40.42	-5.14	27.7
	ADS	-10.704	-99.278	-79.75	-75.68	-19.96	27.5
None	MGT	-10.704	-99.278	-14.1	-14.1	-3.46	27.3
	MADS	-10.704	-99.278	-42.46	-40.35	-5.27	27.6
	Switch	-10.704	-99.278	-42.58	-40.42	-5.14	27.3
	No deflection	-10.704	-99.278	-43.53	-41.33	-4.81	30.2
	ADS	-10.704	-99.278	-79.41	-75.36	-19.84	29.32
Directional Derivative	MGT	-10.704	-99.278	-13.46	-13.46	-3.93	29.6
	MADS	-10.704	-99.278	-42.41	-40.26	-5.69	28.2
	Switch	-10.704	-99.278	-43.53	-41.33	-4.81	27.9
	No deflection	-10.704	-99.278	-73.85	-70.09	-16.46	28.2
	ADS	-10.704	-99.278	-73.85	-70.09	-17.01	28.1
Quadratic Fit	MGT	-10.704	-99.278	-73.85	-70.09	-16.91	28.9
	MADS	-10.704	-99.278	-73.85	-70.09	-16.22	28.5
	Switch	-10.704	-99.278	-73.85	-70.09	-16.46	28.3

Table
5.3

Variables n = 500
Inequality constraints m1 = 200
Equality constraints m2 = 50
Iterations = 1000
CPLEX solution = -
45.8
CPLEX CPU seconds = 7.71
CPLEX Iterations =
1844

Line Search	Deflection Strategy	Optimal objective	Initial objective	Stage 1	Stage 2	Stage 3	Total CPU sec
	No deflection	-45.8	-309.81	-222.69	-211.33	-46.41	41.2
	ADS	-45.8	-309.81	-254.4	-241.4	-59.87	42.4
None	MGT	-45.8	-309.81	-71.57	-71.57	-32.09	41.3
	MADS	-45.8	-309.81	-204.47	-194.04	-37.67	40.9
	Switch	-45.8	-309.81	-222.69	-211.33	-46.43	41.64
	No deflection	-45.8	-309.81	-220.12	-208.88	-42.85	43.8
	ADS	-45.8	-309.81	-253.74	-240.79	-60.29	44.3
Directional Derivative	MGT	-45.8	-309.81	-78.83	-78.85	-34.75	44.2
	MADS	-45.8	-309.81	-205.94	-195.43	-41.66	43.9
	Switch	-45.8	-309.81	-220.12	-208.88	-42.88	41.7
	No deflection	-45.8	-309.81	-236.03	-224.01	-55.36	42.1
	ADS	-45.8	-309.81	-236.03	-224.01	-55.15	41.9
Quadratic Fit	MGT	-45.8	-309.81	-236.03	-224.01	-55.35	43.2
	MADS	-45.8	-309.81	-236.03	-224.01	-55.32	42.9
	Switch	-45.8	-309.81	-236.03	-224.01	-55.37	43.2

Table 5.4

Variables n = 750
Inequality constraints m1 = 300
Equality constraints m2 = 75
Iterations = 1000
CPLEX solution =
40.85
CPLEX CPU seconds = 47.03
CPLEX Iterations =
3402

Line Search	Deflection Strategy	Optimal objective	Initial objective	Stage 1	Stage 2	Stage 3	Total CPU sec
	No deflection	40.85	-406.04	-123.231	-117.085	64.44	513.3
	ADS	40.85	-406.04	-312.57	-296.591	36.743	563.2
None	MGT	40.85	-406.04	-312.68	-296.692	36.171	543.6
	MADS	40.85	-406.04	-105.97	-100.68	60.251	596.7
	Switch	40.85	-406.04	-123.231	-117.08	64.41	521.4
	No deflection	40.85	-406.04	-129.218	-122.647	64.758	598.3
	ADS	40.85	-406.04	-311.062	-295.196	37.256	620.2
Directional Derivative	MGT	40.85	-406.04	-310.76	-294.888	35.411	635.1
	MADS	40.85	-406.04	-110.11	-104.62	62.431	612.31
	Switch	40.85	-406.04	-129.218	-122.647	64.756	641.2
	No deflection	40.85	-406.04	-278.02	-264.13	41.11	622.8
	ADS	40.85	-406.04	-278.02	-264.13	40.787	616.2
Quadratic Fit	MGT	40.85	-406.04	-278.02	-264.13	41.877	621.3
	MADS	40.85	-406.04	-278.02	-264.13	41.467	641.1
	Switch	40.85	-406.04	-278.02	-264.13	41.1	614.7

Table 5.5

Variables n = 1000
Inequality constraints m1 = 400
Equality constraints m2 = 100
Iterations = 1000
CPLEX solution = -10.843
CPLEX CPU seconds = 167.38
CPLEX Iterations =
5288

Line Search	Deflection	Optimal	Initial	Stage 1	Stage 2	Stage 3	Total
	Strategy	objective	objective				CPU sec
	No deflection	-10.843	-501.625	-238.223	-226.08	2.407	1052.6
	ADS	-10.843	-501.625	-377.65	-377.665	-25.195	1091.5
None	MGT	-10.843	-501.625	-26.635	-26.635	12.092	1068.3
	MADS	-10.843	-501.625	-212.651	-202.32	4.695	1063.7
	Switch	-10.843	-501.625	-238.23	-226.83	2.401	1087.4
	No deflection	-10.843	-501.625	-234.96	-222.98	3.875	1099.9
	ADS	-10.843	-501.625	-395.38	-375.185	-23.478	1100.3
Directional Derivative	MGT	-10.843	-501.625	-26.04	-26.042	11.706	1141.6
	MADS	-10.843	-501.625	-208.97	-198.539	10.449	1086.4
	Switch	-10.843	-501.625	-234.96	-222.98	3.879	1090.4
	No deflection	-10.843	-501.625	-362.79	-344.29	-19.526	1089.3
	ADS	-10.843	-501.625	-362.79	-344.29	-19.332	1086.4
Quadratic Fit	MGT	-10.843	-501.625	-362.79	-344.29	-19.278	1082.3
	MADS	-10.843	-501.625	-362.79	-344.29	-19.218	1089.6
	Switch	-10.843	-501.625	-362.79	-344.29	-19.521	1086.7

Chapter 6

Summary and Conclusions

6.1 Summary of Results

In the area of mathematical programming, nondifferentiable optimization techniques have always been recognized as important tools. These techniques have been employed in the contexts of Lagrangian duality, exact penalty function methods and Dantzig-Wolfe decomposition. To expedite the solution process, many algorithmic approaches employ subgradient based methods. Our research effort was aimed at making some contributions in developing theoretically sound and computationally effective procedures for solving large-scale linear programming problems that arise in the context of solving various problems such as discrete linear or polynomial, and continuous nonlinear, nonconvex programming problems.

It has been empirically demonstrated that subgradient based approaches are the most promising in practice for solving the Lagrangian duals. In this research effort we have presented a primal-dual subgradient algorithm for solving the Lagrangian dual problem. This procedure uses computationally effective deflection strategies along with some line search methods to solve the dual problem, next an average weighting rule is implemented to recover primal optimal solutions and finally a primal penalty function method is used for further polishing the obtained primal solution to near feasibility and optimality.

Stage I of the proposed algorithm derives a lower bound via the Lagrangian dual problem. Stage II estimates a primal optimum during the latter part of this same process. Finally Stage III attempts to attain a near feasible and optimal solution by using a suitable penalty function method. This proposed algorithm includes several promising strategies. In particular, a new Modified Average Direction Strategy (M-ADS) is developed that derives a value for the deflection parameter by projecting the optimal direction onto the space spanned by the current gradient and the previous direction. We have also employed other deflection strategies like Modified Gradient Technique (MGT) and the Average Direction Strategy (ADS) in conjunction with suitable line search methods. Our Directional Derivative line search method which ascertains whether to increase or decrease the step-length value based on the right-hand and left-hand derivative information available at each iteration, has performed reasonably well when compared to the Quadratic fit line search method. We have also presented results for those cases wherein no line search method was employed.

To summarize, in this research effort we have proposed a hybrid primal-dual subgradient algorithm for solving the Lagrangian dual problem. Based on computational experiences using some randomly generated test problems, we have observed that the proposed algorithm performs reasonably well but we emphasize that there is still need for developing further improved Lagrangian dual solution procedures that will enhance the solution quality.

6.2 Recommendations for Further Research

Concluding this thesis, we would like to point out a few avenues that can lead to new research topics, or that can initiate extensions leading to further improvements in the proposed approaches.

Instead of the proposed approach, a more restrictive Lagrangian approach can be implemented. In this method, the constraint set can be partitioned into those which are predicted to be active at optimality and the remaining ones which are not. We can then perform a dual ascent as in Stage I by employing a set of very promising deflection strategies in conjunction with some suitable line search methods, and whenever the progress slows, we can switch to Stage II to obtain an average primal solution from the above subproblem solutions using some convex combination strategy. The new primal solution can then be used to re-partition the constraint set. This process can be repeated as long as there is an improvement in the dual. The solution obtained from the above procedure can be further refined by using a penalty function method similar to the one proposed in this thesis.

Alternatively, we can represent the primal problem in some non-basic variable space and then attempt to crash an advanced basis to construct a representation of the linear program. This solution then can be used to partition the constraint set as before and the process can be continued to successively improve the dual solution. A Penalty function approach can then be used to derive a near feasible and optimal primal solution.

The primal convergence theorems for Lagrangian dual subgradient based methods admit some possible rules for selecting convex combination weights along with corresponding step-length rules. So far the simple average weighting rule seems to perform well over others. But, a more practical and promising rule can be devised to enhance the performance of Stage II, which will reduce the burden on Stage III.

It is evident from the results presented that Stage III has performed well when compared to other stages. In view of this one can construct a penalty function for the dual problem and solve it using the primal-dual subgradient algorithm (similar to the one presented in Stage III).

Penalty function for the dual problem can be represented as

$$h(\alpha,\beta) = \alpha b_1 + \beta b_2 - \sum_{i=1,..,n} (x_i * + \mu) \max\{0, \mathrm{A}_{1i}{}^{t}\alpha + A_{2i}{}^{t}\beta - \mathrm{c}_i\}.$$

The subgradient ξ for the above function can be computed as

$$\xi = [b_1, b_2] - \sum_{i=1,..,n} (x_i * + \mu)[\mathrm{A}_{1i}{}^{t}, A_{2i}{}^{t}].$$

Some preliminary investigation into such a solution process was not very encouraging but it is one of the options that need a more detailed study.

Bibliography

Agmon, S. "The Relaxation Method for Linear Inequalities", *Canadian Journal of Mathematics*, 6, 382-392, 1954.

Bazaraa, M.S., J.J. Jarvis, and H.D.Sherali. *Linear Programming and Network Flows*, John Wiley and Sons, Second Edition, New York, NY, 1990.

Bazaraa, M.S., and H.D.Sherali. "On the Choice of Step Size in Subgradient Optimization", *European Journal of Operations Research*, 7, 380-388, 1981.

Bazaraa, M.S., H.D.Sherali, and C.M.Shetty. *Nonlinear Programming: Theory and Algorithms*, John Wiley and Sons, Second Edition, New York, NY, 1993.

Brezinski, C., and M.Redivo. "Treatment of Near-Breakdown in the CGS Algorithm", *Numerical Algorithms*, 7, 33-73, 1994.

Camerini, P.M., L. Fratta, and F. Maffioli. "On Improving Relaxation Methods by Modified Gradient Techniques", *Mathematical Programming Study*, 3, 26-34, 1975.

Chan, T.F., and R.E. Bank. "A Composite Step Bi-Conjugate Gradient Algorithm for Solving Nonsymmetric Linear Systems", *Numerical Algorithms*, 7, 1-16, 1994.

Chan, T.F., and T. Szeto. "A Composite Step Conjugate Gradients Squared Algorithm for Solving Nonsymmetric Linear Systems", *Numerical Algorithms*, 7, 17-32, 1994.

Demyanov, V.F. and L.V. Vasilev. *Nondifferentiable Optimization*, Springler-Verlag, 1985.

Erlenkotter, D. "A Dual-Based Procedure for Uncapacitated Facility Location", *Operations Research*, 26, 992-1009, 1978.

Eremin, I.I. "Methods of Fefer's Approximations in Convex Programming", *Mathematical Notes of Academy of Science USSR*, 3(2), 139-149, 1968.

Falk, J.E. "Lagrange Multipliers and Nonlinear Programming", *Journal of Mathematical Analysis*, 19, 141-159, 1967.

Fisher, M.L. "The Lagrangian Relaxation Method for Solving Integer Programming Problems", *Management Science*, 27, 1-18, 1981.

Fisher, M.L., and R. Jaikumar, and L. Van Wassenhove. "A Multiplier Adjustment Method for the Generalized Assignment Problem", *Management Science*, 32, 9, 1095-1103, 1986.

Fisher, M.L., and P. Kedia. "Optimal Solution of Set Covering/Partitioning Problem using Dual Heuristics", *Management Science*, 36, 674-688, 1990.

Fletcher, R., and C.M. Reeves. "Function Minimization by Conjugate Gradients", *Computer Journal*, 7, 149-154, 1959.

Geoffrion, A.M. "Elements of Large-Scale Mathematical Programming", *Management Science*, 10, 652-691, 1970.

Geoffrion, A.M. "Lagrangian Relaxation for Integer Programming", *Mathematical Programming Study*, 2, M.L.Balinski (Ed.), North-Holland, Amsterdam, 82-114, 1974.

Goffin, J.L. "On the convergence Rate of the Subgradient Methods", *Mathematical Programming*, 13, 329-347, 1977.

Held, M and R.M. Karp. "The Traveling Salesman Problem and Minimum Spanning Trees", *Operations Research*, 18, 1138-1162, 1970.

Held, M and R.M. Karp. "The Traveling Salesman Problem and Minimum Spanning Trees: Part II", *Mathematical Programming*, 1, 6-25, 1971.

Held, M., P.Wolfe, and H.D. Crowder. "Validation of Subgradient Optimization", *Mathematical Programming*, 6(1), 62-88, 1974.

Hestenes, M.R. and E. Stiefel. "Methods of Conjugate Gradients for Solving Linear Systems", *Journal of Research*, National Bureau of Standards, 29, 403-439, 1952.

Kim, S and H. Ahn. "Convergence of a Generalized Subgradient Method for Nondifferentiable Convex Optimization", *Mathematical Programming*, 50, 75-80, 1991.

Kim, S., H. Ahn and S. Cho. "Variable Target Value Subgradient Method", *Mathematical Programming*, 49, 359-369, 1991.

Larsson, T. and Z. Liu. "A Primal Convergence Result for Dual Subgradient Optimization with Application to Multi commodity Network Flows", *Research Report*, Dept of Math., Linkoping Institute of Technology, S-581 83, Sweden, 1989.

Motkin, T and I.J. Schoenberg. "The Relaxation Method for Linear Inequalities", *Canadian Journal of Mathematics*, 6, 393-404, 1954.

Nemhauser, G.L. and L.A. Wolsey. *Integer and Combinatorial Optimization*, John Wiley and Sons, New York, 1988.

Oettli, W. "An Iterative Method, Halving Linear Rate of Convergence, for Solving a Pair of Dual Linear Programs", *Mathematical Programming*, 3, 302-311, 1972.

Parker, G.R. and R.L. Rardin. *Discrete Optimization*, Academic Press, Inc., San Diego, CA 92101, 1988.

Polyak, B.T. "A General Method for Solving Extremal Problems", *Soviet Mathematics*, 8, 593-597,1967.

Polyak, B.T. "Minimization of Unsmooth Functionals", *USSR Computational Mathematics and Mathematical Physics*, 9, 509-521, 1969.

Polyak, B.T. and Q. Mayne. "On Three Approaches to the Construction of Nondifferentiabl Optimization Algorithms", *System Modeling and Optimization: Proceedings of the 11th IFIP Conference*, P.Thoft-Christensen (Ed.), Lecture Notes in Control and Information Science, 59, 331-337, 1984.

Rockafellar, R.T. *Convex Analysis*, Princeton Univ. Press, 1969.

Rosen, J.B. and S. Suzuki. "Construction of Nonlinear Programming Test Problems", *Communication of ACM*, 8, 113, 1965.

Sen, S and H.D. Sherali. "A Class of Convergent Primal-Dual Subgradient Algorithms for Decomposable Convex Programs", *Mathematical Programming*, 35, 279-297, 1986.

Sherali, H.D., G. Choi, and C.H. Tuncbilek. "A Variable Target Value Method for Nondifferentiable Optimization", *manuscript*, Dept of ISE, Virginia Polytechnic Institute And State University, Blacksburg, VA, 1993.

Sherali, H.D. and G. Choi. "Primal Recovery Theorems for Lagrangian Dual Subgradient Methods", *manuscript*, Dept of ISE, Virginia Polytechnic Institute And State University, Blacksburg, VA, 1993.

Sherali, H.D. and D.C. Myers. "Dual Formulations and Subgradient Optimization Strategies for Linear Programming Relaxations of Mixed-Integer Programs", *Discrete Applied Mathematics*, 20, 51-68, 1988.

Sherali, H.D. and O. Ulular. "A Primal-Dual Conjugate Algorithm for Specially Structured Linear and Convex Programming Problems", *Applied Mathematics and Optimization*, 20, 193-221, 1989.

Vita

Ananth Madabushi was born on September 4, 1972 in Guntur, India. He finished his schooling in 1990 from Loyola Academy. He received his B.S. in Mechanical Engineering in 1994 from S.K.D. University, Kurnool, India. In October 1994, he joined the M.S. program of the Industrial and Systems Engineering Department of Virginia Polytechnic Institute and State University. He is currently working as a Systems Engineer for TransQuest Information Solutions.

Ananth Madabushi