Nondifferentiable Optimization Algorithms with Application to

Solving Lagrangian Dual Problems

by

Gyunghyun Choi

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Philosophy of Doctor

in

Industrial and Systems Engineering

APPROVED:

Dr. Hanif D. Sherali, Chairman

Dr. Christopher A. Beattie

Dr. G. V. Loganathan

Dr. C. Patrick Koelling

Dr. Subhash C. Sarin

December 1993

Blacksburg, Virginia

Nondifferentiable Optimization Algorithms with Application to Solving Lagrangian Dual Problems

by

Gyunghyun Choi

Dr. Hanif D. Sherali, Chairman

Industrial and Systems Engineering

(ABSTRACT)

In this research effort, we consider nondifferentiable optimization (NDO) problems that arise in several applications in science, engineering, and management, as well as in the context of other mathematical programming approaches such as Dantzig-Wolfe decomposition, Benders decomposition, Lagrangian duality, penalty function methods, and minimax problems. The importance and necessity of having effective solution methods for NDO problems has long been recognized by many scientists and engineers. However, the practical use of NDO techniques has been somewhat limited, principally due to the lack of computationally viable procedures, that are also supported by theoretical convergence properties, and are suitable for solving large-scale problems. In this research, we present some new algorithms that are based on popular computationally effective strategies, while at the same time, do not compromise on theoretical convergence issues.

First, a new variable target value method (VTVM) is introduced that has an ε-convergence property, and that differs from other known methods in that it does not require any prior assumption regarding bounds on an optimum or regarding the solution space. In practice, the step-length is often calculated by using an estimate of the optimal objective function value. For general nondifferentiable optimization prob-

lems, however, this may not be readily available. Hence, we design an algorithm that does not assume the possibility of having an a prior estimate of the optimal objective function value. Furthermore, along with this new step-length rule, we present a new subgradient deflection strategy in which a selected subgradient is rotated optimally toward a point that has an objective function value less than the incumbent target value. We also develop another deflection strategy based on Shor's space dilation algorithm, so that the resulting direction of motion turns out to be a particular convex combination of two successive subgradients and we establish suitable convergence results.

In the second part of this dissertation, we consider Lagrangian dual problems. Our motivation here is the inadequacy of the simplex method or even interior point methods to obtain quick, near-optimal solutions to large linear programming relaxations of certain discrete or combinatorial optimization problems. Lagrangian dual methods, on the other hand, are quite well equipped to deal with complicating constraints and ill-conditioning problems. However, available optimization methods for such problems are not very satisfactory, and can stall far from the optimal objective function value for some problems. Also, there is no practical implementation strategy for recovering a primal optimal solution. This is a major shortcoming, even if the method is used only for bounding purposes in the context of a branch and bound scheme. With this motivation, we present new primal convergence theorems that generalize existing results on recovering primal optimal solutions, and we describe a scheme for embedding these results within a practical primal-dual Lagrangian optimization procedure.

Acknowledgements

The faith and time of many people around me made this research work possible. I wish to express gratitude to the people who have offered unlimited support during my school days.

First, my gratitude goes to Dr. Hanif D. Sherali, who advised and led my research. He has introduced and guided me to the world of optimization and mathematical programming. I am deeply indebted to both his invaluable professional and personal assistance. I would also like to thank Dr. Christopher A. Beattie, Dr. C. Patrick Koelling, Dr. G.V. Loganathan, and Dr. Subhash C. Sarin, who are my committee members for their comments and consideration.

Next, I wish to acknowledge my family, especially my parents, Gonchil and Taehyang Choi, who have provided support and love throughout my life. Also, I would like to thank my beloved wife, Jinmee, and son, Jeong Hyuk, without whose love, patience, and faith in me, I wouldn't have make it through. I also owe an inexpressible debt to my brother, sister, and their spouses for their consideration and support.

Finally, my special thanks goes to parents-in-law and my friends who encouraged and provided support.

Table of Contents

1. Introduction

1.1 Motivation

This research effort begins by considering (minimization) optimization problems having convex, but not necessarily differentiable, objective functions. If some constraints are involved in any such problem, we assume that the constrained feasible region is simple so that the projection operation of finding the closest feasible solution can be easily performed. Problems of this type are called Nondifferentiable Optimization (NDO) problems, and have received a great deal of attention over the last two decades, motivated by many applications in science, engineering, and economics, as well as by the fact that they arise in subproblems for several optimization algorithms. Examples of such contexts include min-max problems, piecewise approximations, multiobjective programming problems, systems of inequalities, dynamic programming problems, and stochastic optimization problems. In mathematical programming approaches, NDO's arise in the contexts of Dantzig-Wolfe decomposition, large-scale discrete optimization, Lagrangian duality, and exact pen-

alty function methods, to cite a few. For such NDO problems, classical optimization methods might fail to find an optimal solution. As an example, Wolfe (1975) presents an instance for which a standard application of a quasi-Newton method converges to a non-optimal kink.

Most of the theoretical background for NDO's can be traced to convex analysis and the study of directional derivatives. Rockafellar (1970) provides a majority of the necessary mathematical background, while Demyanov and Vasilev (1985) and Shor (1985) cover most of the optimality theory and algorithms for NDO problems. Also, Shor (1983), Polyak and Maine (1984), and Zowe (1985) present some algorithmic insights for different types of approaches for such problems.

There are three typical questions that we have to consider for NDO algorithms.

i) How can we find the direction of motion ?

ii) How can we choose a suitable step length ?

iii) Is there any implementable stopping rule ?

Of course, these questions are quite routine for every optimization algorithm. But in NDO, their answers are based on a different set of theoretical issues.

For a convex (nondifferentiable) function, it is well known (see Bazaraa, Sherali, and Shetty, 1993) that for any $\bar{\mathbf{x}}$, there exists a vector $\mathbf{g}$ called a *subgradient* of f at $\bar{\mathbf{x}}$, that satisfies

$$f(\mathbf{x}) - f(\bar{\mathbf{x}}) \geq \mathbf{g}^t(\mathbf{x} - \bar{\mathbf{x}}) \quad \text{for all} \quad \mathbf{x}$$

The collection of all such subgradients, that is, the set

$$\partial f(\bar{\mathbf{x}}) = \{\mathbf{g} : f(\mathbf{x}) - f(\bar{\mathbf{x}}) \geq \mathbf{g}^t(\mathbf{x} - \bar{\mathbf{x}}) \quad \text{for all} \quad \mathbf{x}\}$$

is called the *subdifferential* of f at $\bar{\mathbf{x}}$. Since any convex function (defined on the n-dimensional Euclidean space E^n) is differentiable almost everywhere (see Shor, 1985), the subdifferential of f at $\bar{\mathbf{x}}$ can also be defined as

$$\partial f(\bar{\mathbf{x}}) = \text{Conv}\{\mathbf{g} : \mathbf{g} = \lim_{k \to \infty} \nabla f(\mathbf{x}_k), \lim_{k \to \infty} \mathbf{x}_k = \bar{\mathbf{x}}, \ \mathbf{x}_k \neq \bar{\mathbf{x}} \quad \text{for all} \quad k\}$$

where $\nabla f(\mathbf{x}_k)$ is the gradient of f at $\mathbf{x}_k$ for each k , and where Conv$\{\cdot\}$ is the convex hull of the set $\{\cdot\}$. Note that if f is differentiable at $\bar{\mathbf{x}}$, then $\partial f(\bar{\mathbf{x}}) = \{\nabla f(\bar{\mathbf{x}})\}$ is a singleton, and is the gradient of f at $\bar{\mathbf{x}}$.

It is also well known (see Bazaraa et al., 1993) that the directional derivative $f(\bar{\mathbf{x}}; \mathbf{d})$ of f at $\bar{\mathbf{x}}$ with respect to a direction $\mathbf{d}$ exists for any convex function f, and is given by

$$f'(\bar{\mathbf{x}}; \mathbf{d}) = \lim_{\lambda \to 0^+} \frac{f(\bar{\mathbf{x}} + \lambda \mathbf{d}) - f(\bar{\mathbf{x}})}{\lambda} = \sup\{\mathbf{d}^t\mathbf{g} : \mathbf{g} \in \partial f(\bar{\mathbf{x}})\}$$

Note that a vector $\bar{\mathbf{d}}$ is called a steepest descent direction of f at $\bar{\mathbf{x}}$ if $f'(\bar{\mathbf{x}}; \bar{\mathbf{d}}) = \text{minimum}\{f'(\bar{\mathbf{x}}; \mathbf{d}) : \|\mathbf{d}\| \leq 1\}$. Also, it is known that $\mathbf{d}$ is a descent direction if and only if $f'(\bar{\mathbf{x}}; \mathbf{d}) < 0$, that is, if and only if $\mathbf{d}^t\mathbf{g}$ is negative and bounded away from zero for all $\mathbf{g} \in \partial f(\bar{\mathbf{x}})$. Thus, if f is not differentiable at $\bar{\mathbf{x}}$, then for any $\bar{\mathbf{g}} \in \partial f(\bar{\mathbf{x}})$, the *anti-subgradient* $-\bar{\mathbf{g}}$ is not necessarily a descent direction in general, since we do not have $\bar{\mathbf{g}}^t\mathbf{g} \geq \varepsilon > 0$ for all $\mathbf{g} \in \partial f(\bar{\mathbf{x}})$, for some $\varepsilon > 0$. Moreover, the subdifferential $\partial f(\bar{\mathbf{x}})$ is not available in general, and hence we do not have any global information of f at a nondifferentiable point $\bar{\mathbf{x}}$. Hence, strategies for finding the direction of motion for NDO involve different theoretical issues from the differentiable cases, and are not

so simple. One straightforward approach is to adopt an arbitrary anti-subgradient direction of motion, and to prescribe some suitable step-size along this direction, and reiterate. In this case, as we mentioned earlier, the direction of motion given by an anti-subgradient not only is a blind direction, but such a process may painfully zig-zag toward an optimum, if at all it is optimally convergent. Accompanying such a strategy is the difficulty of choosing a suitable step-length because we cannot conduct any usual line search. Note that if $\mathbf{0} \in \partial f(\bar{\mathbf{x}})$, then $\bar{\mathbf{x}}$ is an optimal solution, given that f is convex. Thus, we can use $\|\mathbf{g}\| < \varepsilon$ as a stopping rule, where $\varepsilon > 0$ is sufficiently small. However, this inequality might not be satisfied at all, even at an optimal solution because of the arbitrariness of the selected subgradient. Therefore, we need some implementable stopping rules. With these observations, we present some new strategies and corresponding theories for nondifferentiable optimization problems.

For the second part of this research, we consider Lagrangian dual (LD) problems which belong to a very important area of application of NDO techniques. It has been proven empirically that subgradient based approaches are the most promising in practice among several alternative solution methods for LD's. However, there is still a strong need for developing efficient and practical NDO algorithms for solving arbitrary Lagrangian dual problems. Here, we present a new method for solving any given Lagrangian dual problem, independent of any special problem structure. Our motivation is to provide an alternative to linear programming solvers for obtaining lower bounds for discrete optimization problems, in the form of an effective NDO procedure that can be applied to a LD of the given problem.

Note that when Lagrangian duality is used for obtaining a lower bound on the given primal problem, we might also be interested in finding a primal optimal solution. In

particular, if the subgradient method is used for solving the underlying LD problem, we cannot obtain primal feasible solutions in general. 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. In this spirit, we present convergence theorems that 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. In addition, we propose a new method for recovering a primal optimal solution from a dual optimal solution under some relaxed assumptions.

1.2 Executive Summary of Various Proposed Algorithms

A variable target value method for subgradient optimization algorithms

Subgradient methods are invariably used for solving nondifferentiable optimization (NDO) problems. In the so-called *pure subgradient method*, the direction of motion is taken as an anti-subgradient direction and some prescribed step-length rules are employed instead of the 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, with the gap of this distance from optimality eventually approaching zero. Typically, at each iteration, 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 the direction of motion can generate a zig-zagging path, resulting in a slow convergence rate.

As just mentioned, 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. Several step-length rules and some test results are available in the literature. Among them, the following three rules are the most popular and widely used in practice and also guarantee convergence in the sense of decreasing the Euclidean distance to optimality. Here, $\mathbf{x}_k$ is the current iterate, $\mathbf{g}_k$ is a subgradient evaluated at $\mathbf{x}_k$, and λ_k is the prescribed step length.

$$\lambda_k = h_k, \quad \text{where} \quad h_k \geq 0, \lim_{k \to \infty} h_k = 0, \quad \text{and} \quad \sum_{k=1}^{\infty} h_k = \infty \tag{1.1a}$$

$$\lambda_k = h_k / \|\mathbf{g}_k\|, \quad \text{where} \quad h_k \geq 0, \lim_{k \to \infty} h_k = 0, \quad \text{and} \quad \sum_{k=1}^{\infty} h_k = \infty \tag{1.1b}$$

$$\lambda_k = \beta_k \frac{f(\mathbf{x}_k) - w}{\|\mathbf{g}_k\|^2}, \quad \text{where} \quad 0 < \varepsilon_1 \leq \beta_k \leq \varepsilon_2 < 2, \quad \text{and } w \text{ is a target value} \tag{1.1c}$$

Rule (1.1a) has been used by Held and Karp (1971) in their well known 1-tree Lagrangian relaxation of the Traveling Salesman Problem. Earlier than this, Ermolev (1966) analyzed the general convergence behavior of this rule. Polyak (1967) provides a general convergence result using rule (1.1b), and Shor (1968) employs this rule in conjunction with generalized gradient concepts. Shor (1985) and Goffin (1977) study corresponding convergence rates. Notwithstanding their clean convergence

properties, the main problem with these rules lies in the slow convergence rate they produce in practice.

Rule (1.1c) can be viewed as three different rules dependent on the choice of the target value w. If w is an upper bound estimate, that is, $w \geq f^* \equiv f(\mathbf{x}^*)$ where $\mathbf{x}^*$ is an optimal solution and f^* is the optimal objective function value, then either finitely a solution $\mathbf{x}_k$ with $f_k \equiv f(\mathbf{x}_k) \leq w$ is obtained, or, infinite convergence to the target value occurs at a geometric rate (see Polyak, 1969). Agmon (1954), Eremin (1968), Motzkin and Schoenberg (1954), and Oettli (1972) use the exact target value $w = f^*$ in their analysis. Also, the geometric convergence of this method has been proven in Motzkin and Schoenberg (1954) and Oettli (1972). But, these convergence results are not so useful in practice, since they require some information about the optimal objective value. Hence, it is impractical to use such a strategy directly, and this must be coordinated with some varying target value technique. The lower estimate case, that is, $w \leq f^*$, has been used in Eremin (1968), Held, Wolfe, and Crowder (1974), and Ali and Kennington (1986). In these works, a lower bound w on the optimal objective value is computed explicitly in the algorithmic process using duality. Polyak (1969) provides the original convergence theorem and, later, Allen, Helgason, Kennington, and Shetty (1987) generalized this result by relaxing the condition $0 < \varepsilon_1 \leq \beta_k \leq \varepsilon_2 < 2$ to $0 < \beta_k \leq \varepsilon_2 < 2$ and $\sum_{k=1}^{\infty} \beta_k = \infty$.

There are several research studies that conduct computational experiments on using these rules and conjecture some superiority of one rule over another in practice. But no direct comparisons and no practical guidelines for selecting parameters have been made for general nondifferentiable optimization problems. Only recommen-

dations for some specially structured problems have been proposed without any guarantee of superiority (see Polyak, 1969, and Held et al., 1974).

Since the optimal objective value is not known, any (conjugate) subgradient algorithm with the step-length rule (1.1c), in general, cannot be guaranteed to generate a sequence of iterates that converges to an optimal solution. In this regard, the variable target value method is one approach to not only overcome the blindness in the choice of the target value, but also to accelerate convergence. Bazaraa and Sherali (1981) present a variable target rule for solving Lagrangian duals of discrete problems. Their target value is computed by a convex combination of a fixed upper bound (which is available via a primal feasible solution) and the incumbent dual solution. Along with this, they present a convex combination weighting rule as a function of the iterations. Recently, Kim, Ahn, and Cho (1991) present a similar rule for minimizing strongly convex functions. However, their method requires an initial lower bound $L_1 < f^*$, and also an estimate $R \geq \|\mathbf{x}_1 - \mathbf{x}^*\|$. Therefore, these two strategies are not practically implementable for general nondifferentiable optimization problems.

Strictly speaking, there is no practical step-length rule that assumes no knowledge whatsoever of the optimal objective value. Therefore, in this research, we present a new variable target value method, under the strict assumption that the only known bounds on the optimal value are $\pm\infty$. This new, nonmonotone, Variable Target Value Algorithm (VTVM) updates target values whenever it is necessary by increasing or decreasing the old target value in an outer loop, depending on the information obtained in the process of the algorithm. Accordingly, in each inner loop, Algorithm VTVM generates a sequence of iterates in a manner such that the sequence of incumbent solutions converges to a (near)-optimal solution. Furthermore, this

method affords a reasonable stopping criterion, and generates a sequence of solutions that guarantee ε-optimality for any given $\varepsilon > 0$, without any knowledge of the optimal objective value.

Deflected Subgradient Methods

No matter what step-length rule is implemented, the pure subgradient method has a limit on its computational performance, simply due to the direction of motion used. As with the steepest descent direction for the differentiable case, the anti-subgradient direction for the nondifferentiable case also results in a zig-zagging phenomenon that might manifest itself at any stage of the subgradient algorithm, causing 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. A conjugate gradient direction for the differentiable case is computed by combining the current anti-gradient with the previous direction, while forcing some conjugacy requirement (see Hestenes and Stiefel, 1952, and Fletcher and Reeves, 1959). However, for the nondifferentiable case, this concept is used only as a strategy to deflect the subgradient, since conjugacy may no longer be meaningful.

Accordingly, the direction of motion $\mathbf{d}_k$ at $\mathbf{x}_k$ is computed as

$$\mathbf{d}_k = -\mathbf{g}_k + \psi_k \mathbf{d}_{k-1} \tag{1.2}$$

where $\psi_k \geq 0$ is a deflection parameter, $\mathbf{g}_k$ is a subgradient of f at $\mathbf{x}_k$, and $\mathbf{d}_{k-1}$ is the previous direction, with $\mathbf{d}_0 \equiv \mathbf{0}$. Then, the new iterate is computed by

$$\mathbf{x}_{k+1} = P_X[\mathbf{x}_k + \lambda_k \mathbf{d}_k] \tag{1.3}$$

where λ_k is a suitable step-length and $P_X(\cdot)$ is a projection operation onto the set X.

There are well-known tested subgradient deflection algorithms of the type (1.2), namely, 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 known to be superior to the pure subgradient method in practice along with a specially designed step-length selection rule. However, MGT does not provide a sufficient rate of improvement, even though it continues to iteratively move closer to the optimal solution and is superior to the pure subgradient algorithm theoretically, because its direction of motion frequently turns out to be selected as the anti-subgradient direction itself, especially at the final stages of the procedure. Sherali and Ulular (1989) demonstrate the superiority of their scheme to Camerini et al.'s procedure and particularly, to pure subgradient based approaches. Despite its computational superiority (better rate of improvement in the objective function value), ADS merely bisects the angle between the anti-subgradient and the previous direction, regardless of any local information. More generally speaking, the conjugate subgradient direction can be viewed as a nonnegative linear combination of a collection of subgradients obtained thus far at any stage of the procedure. In this way, Kim and Ahn (1991) prove their convergence theorem using the step-length rule (1.1a) and (1.1b) along with ε-subgradient concepts under the assumption that $\varepsilon \to 0$.

However, dominant deflection strategies and general convergence behavior analyses are not as yet available. Here, we provide general convergence theorems in which the only assumption on the deflection parameter is nonnegativity, under the case of step-length rule (1.1c) for both, upper and lower bound target values. Since these theorems permit considerably more freedom in selecting the deflection parameters under suitable assumptions on the step-length, we propose a new deflection strategy based on some desirable motivation. This new strategy, which we call an Optimally Deflected Subgradient Algorithm (ODSA), generates the direction of motion by rotating the anti-subgradient optimally toward the direction leading to an estimated (artificial) optimal solution, where the latter is updated in the algorithmic process using target values.

The Algorithms ODSA and VTVM will be tested on various classes of NDO problems and a set of parameter values that promise good performance behavior will be recommended. The results from the preliminary tests on some standard test problems and some strongly convex problems show that the new strategies perform well in comparison with existing methods.

Memoryless space dilatation methods

As an alternate approach for NDO, bundle type methods have been proposed by Mifflin (1977), Lemarechal (1978), and Kiwiel (1981), in which the direction of motion is obtained from the convex hull of previous subgradients, known as (subgradient) *bundles*. Unlike the choice of the step-length in subgradient based algorithms, these

bundle methods involve an inexact line search that produces either an improved solution (serious step) or a trial solution (null step). In either case, a new subgradient is computed and added to the existing bundle to find the modified direction of motion. Lemarechal (1980) presents the concepts of this line search. Kiwiel (1985, 1989) and Lemarechal (1985, 1986) introduce several different strategies to construct the bundles. Also, Kiwiel (1990, 1991) presents improved strategies for such methods under the title of proximal bundle methods. A major difference between bundle and ordinary subgradient methods is that the bundle type methods generate a sequence of iterates for which the objective function values are monotone decreasing, while the subgradient based methods produce a sequence of iterates whose Euclidean distances to the optimal solution are monotone decreasing. For this reason, the bundle type methods are called "descent methods." One difficulty with the bundle methods is that they require the solution of a quadratic subproblem at each iteration for finding the direction of motion, and this can be quite expensive. Indeed, it proves to be impractical for larger sized problems.

In contrast with the subgradient based methods and with the above bundle type methods, the variable metric methods have different algorithmic and theoretical characteristics. For the differentiable case, we can obtain a deflected gradient by transforming the metric based on the Hessian matrix (or its approximation). Similarly, for NDO, a subgradient can be deflected by premultiplying with a suitable matrix. Shor (1970a, b) presents a space dilatation procedure that employs an analogue space transformation along the gradient direction. Goffin (1980) presents convergence results for several different choices of transformation parameters. Of noteworthy value due to its computational performance, Shor and Zhurbenko (1971) and Shor (1975) propose a space dilatation algorithm along the difference of two succes-

sive subgradients, called the *r-algorithm*. Assuming that the selected anti-subgradient is almost perpendicular to the direction toward optimality, this r-algorithm is designed to reduce the orthogonal component of the subgradient, with the intent of alleviating the zig-zagging phenomenon. In practice, the computational performance appears good, but the effort is expensive because of the matrix updating procedure. Moreover, its theory is complicated, and it loses the simplicity of subgradient based algorithms that have made the latter so popular. However, by adopting the idea of memoryless updates in quasi-Newton methods (see Nazareth, 1979), we can hope to overcome some of its disadvantages. This can be done by computing the space transformation operator at each iteration by updating the identity matrix instead of the previous approximation. In this case, it will be shown that the memoryless update turns out to be a convex combination of two successive subgradients, leading to a combination of space dilatation and space reduction operations. Assuming that the angle between the anti-subgradient and the direction to the optimal solution is almost orthogonal, the original space dilatation algorithm resolves this situation theoretically, attempting to avoid the zig-zagging behavior. But if this assumption does not hold at a certain point, the space dilatation might produce a worse direction than that of the pure subgradient method. In addition to the memoryless update scheme, we will incorporate such considerations in designing our Memoryless Space Dilatation and Reduction Algorithm (MSDRA).

We have observed that the memoryless update scheme can lighten the computational burden of the original r-algorithm. However, the precise role of the space transformation parameter in practice has yet to be studied for our algorithm. In other words, Algorithm MSDRA needs to determine when the dilatation or the reduction of space is appropriate.

Recovering primal solutions in Lagrangian dual subgradient methods

There are several issues we have to consider in LD approaches. Among them, dualization strategies, solution methods, and primal recovery techniques are the most important factors. In discrete problems, the quality of the lower bound obtained by solving LD problems depends on the dualization schemes because of the ubiquitous presence of a duality gap. These relationships are well explained in Fisher (1981), Parker and Rardin (1988), Sherali and Myers (1988), and Nemhauser and Wolsey (1989). In particular, Sherali and Myers (1988) present some computational results under different dualization strategies.

For our discussion, let us assume that a Lagrangian dual problem formulation has already been selected for a given primal problem, and that subgradient based methods are employed to solve it. We turn our attention to the task of finding a primal optimal solution from the dual subgradient procedure.

Since the dual problem only provides a lower bound on the primal optimal objective value, we need some techniques for recovering a primal solution, other than using an expensive projection operation within the dual procedure. It is known that the subgradient methods are quite promising for solving the dual problem itself but, in general, the optimal solution of the subproblem is not feasible to the primal problem. In contrast, for the cutting plane methods, an optimal dual solution to the master problem (MP) can be used as a set of convex combination weights applied to the optimal solutions of the Lagrangian subproblems in order to find a primal solution (see Bazaraa et al., 1993). For dual subgradient methods, a similar but weaker result

can be deduced. Shor (1985) presents a convex combination weighting rule as a function of the step-lengths in a dual subgradient method, that leads the sequence of convex combinations of the optimal LD subproblem solutions to converge to a primal solution. Also, Larsson and Liu (1988) use an average weighting scheme along with a step-length rule that is a special case of that of Shor's, and show that the same result holds in this case. In this research, we present a convergence theorem that generalizes these previous results, and provides a more flexible choice of admissible step-length rules and convex combination weighting strategies. Moreover, this result permits one to examine more computationally promising step-length rules than the ones considered heretofore.

Our primal convergence result is provided by the dual convergence theory along with a suitable choice of step-lengths and convex combination weights. However, similar to Shor's results, our assumptions are still quite strong so that it is a little doubtful if primal solutions would be robustly obtained in practice. In fact, there are several direction finding strategies other than the pure-subgradient method and there are several well-known step rules that perform well computationally, but for which no primal convergence results are known. In this respect, we also present primal convergence theorems for general deflected subgradient methods.

A hybrid primal-dual subgradient algorithm

Lagrangian Dual (LD) methods are widely used approaches for solving specially structured constrained problems by dualizing complicating constraints in order to obtain easier (than the original) subproblems. These subproblems provide a lower

bound on the original (minimization) problem value, and an ascent scheme attempts to find a greatest lower bound. Also, for linear programming relaxations of discrete problems, Lagrangian dual techniques are often used to obtain bounds for branch-and-bound methods since these linear programming relaxations frequently turn out to be highly degenerate so that usual LP solvers, such as the simplex method, experience great difficulties in solving them. The sparsity and repetitive structures due to reformulation techniques also pose a problem for interior point methods.

Starting with the saddle point optimality conditions of Kuhn and Tucker (1951), the Lagrangian duality concept was first adopted by Everett (1963) as an algorithmic scheme to solve constrained problems as an infinite sequence of unconstrained problems. Falk (1967) presents the use of Lagrangian duality for nonlinear programming. Also for discrete or combinatorial problems, Lagrangian duality has received a great deal of attention after the frontier work of Geoffrion (1970, 1971, 1974) under the name of *Lagrangian Relaxation*. In particular, the work of Held and Karp (1970, 1971) provided the impetus for applying LD approaches for discrete problems in conjunction with subgradient methods.

There are several strategies for solving Lagrangian dual problems. These solution methods can be divided into two large classes, namely, cutting plane methods and subgradient based methods. Other than these, several heuristic multiplier adjustment approaches have been proposed by Erlenkotter (1978), Fisher et al. (1986), and Fisher and Kedia (1990). However, the latter methods are specially designed for each different class of problems by exploiting their specific structures, and hence may not be applicable to general problems. On the other hand, the subgradient based methods, introduced by Held and Karp (1971) and Held, Wolfe, and Crowder (1974), are the

most popular, quite promising, and easy to implement in practice. However, for problems lacking any special structure, these methods can stall far from optimality. Thus, in this research, we propose a hybrid strategy that uses not only well-known subgradient based NDO methods but also the newly developed algorithms, in order to assist the convergence process. To this end, we divide the solution process into three stages.

In the first stage, the algorithm adopts the subgradient based methods, namely MGT, ADS, and ODSA in turn. For the step-length rule, we augment the variable target value method as follows. This procedure contains two major loops, say, the outer and the inner loop. At the beginning of each outer loop, the algorithm selects a target value and a step-length parameter value β. At this point, a deflection strategy (among MGT, ADS, and ODSA) is also selected. Then, the inner loop generates a sequence of iterates by using the selected deflection strategy and the target value, suitably reducing the β parameter value as necessary. If either the algorithm fails to improve for some preassigned consecutive runs, or the incumbent function value reaches an ε-neighborhood of the given target for a given $\varepsilon > 0$, then the algorithm terminates the current outer loop. Note that the new target value for each outer loop is computed by either increasing or decreasing the current target value (at the end of the previous outer loop), depending on the algorithmic progress, and that the step-length parameter β is preassigned as a function of the outer loop counter. The algorithm for the first stage is terminated if either a near-optimal solution is achieved or the inner loop counter reaches the preassigned maximum number of iterations, and the overall algorithm moves to the second stage.

In the second stage of the algorithm, the dual procedure remains the same as in the first stage, while the algorithm updates the primal solution by averaging the subproblem solutions. This last process has been discussed earlier in the primal solution recovery scheme.

Since the primal solution obtained in the previous dual procedure may not be feasible due to the maximum limit placed on the iterations, we may need to further refine the primal solution. The final stage of the proposed algorithm adopts a subgradient deflection strategy on a penalty function formulation that, in theory, guarantees both near-feasibility and near-optimality.

2. Optimially Deflected Subgradient Method and a Variable Target Value Method

2.1 Introduction

Let us consider the problem of minimizing a convex, but not necessarily differentiable, function $f(\mathbf{x})$ over a closed, convex subset X of E_n. Because of the inherent nondifferentiability, classical methods that use gradients or Hessian matrices (or some approximations thereof) to find the direction of motion, may fail to find an optimal solution. Wolfe (1975) presents such an example in which a quasi-Newton method generates a sequence that converges to a non-optimal point. Subgradient methods, on the other hand, are appropriate approaches for such nondifferentiable optimization (NDO) problems. In the, so-called, *pure-subgradient method*, the direction of motion is taken as the anti-subgradient direction and some prescribed step-length rules are employed instead of the usual line search in order to generate a sequence of iterates. However, to assure convergence, a suitable step-length should

be taken so that the iterates get progressively closer in Euclidean distance to an optimal solution, with this distance from optimality eventually going to zero. Typically, at each iteration, the decrement in the Euclidean distance to an optimal solution is relatively smaller than the step-length, and moreover, such a process using an anti-subgradient as the direction of motion can generate a zig-zagging path, resulting in a slow convergence behavior. With this motivation, our research addresses two main concerns for subgradient based algorithms, namely, the choice of a *direction of motion* and the *step-length rule*, in designing a theoretically convergent and a practically effective and implementable procedure.

First, let us comment on step-length rules. It is well 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. Several step-length rules and some test results are available in the literature. Among them, the following three rules given in (2.1) below are the most popular and widely used in practice, and also enjoy convergence properties. Here, $\mathbf{x}_k$ is the current iterate, $\mathbf{d}_k$ is the direction of motion at $\mathbf{x}_k$, and λ_k is the prescribed step-length. Note that although we have written the following rules using a general $\mathbf{d}_k$ for the sake of convenience in future reference, these rules were originally proposed for the pure-subgradient method wherein $\mathbf{d}_k$ was taken as $-\mathbf{g}_k$, the anti-subgradient of f at $\mathbf{x}_k$.

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

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

$$\lambda_k = \beta_k \frac{f(\mathbf{x}_k) - w}{\|\mathbf{d}_k\|^2}, \quad \text{where} \quad 0 < \varepsilon_1 \leq \beta_k \leq \varepsilon_2 < 2, \quad \text{and } w \text{ is a target value} \qquad (2.1c)$$

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. Earlier than this, Ermolev (1966) analyzed the general convergence behavior of this rule. Polyak (1967) provides the general convergence arguments for the rule (2.1b), and Shor (1968) uses this in conjunction with generalized gradient concepts. Goffin (1977) and Shor (1985) study corresponding convergence rates. Notwithstanding their clean convergence properties, the main problem with the rules (2.1a, b) 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. If $w \geq f^* \equiv f(\mathbf{x}^*)$, where $\mathbf{x}^*$ is an optimal solution and f^* is the optimal objective function value, then either finitely a solution $\mathbf{x}_k$ with $f_k \equiv f(\mathbf{x}_k) \leq w$ is obtained, or else, infinite convergence to the target value occurs at a geometric rate (see Polyak, 1969). Agmon (1954), Eremin (1968), Motzkin and Schoenberg (1954), and Oettli (1972) use the exact target value $w = f^*$ in their analyses, and so, the prescribed procedure is not directly implementable, but must be coordinated with some varying target value technique. The lower estimate case, that is, $w \leq f^*$, has been considered by Eremin (1968), Held et al. (1974), and Ali and Kennington (1986). In these papers, a lower bound w on the optimal objective value is assumed to be known, perhaps obtained via duality concepts. Polyak (1969) provides the original convergence theorem, and later, Allen et al. (1987) generalize this result by relaxing the condition $0 < \varepsilon_1 \leq \beta_k \leq \varepsilon_2 < 2$ to $0 < \beta_k \leq \varepsilon_2 < 2$, but requiring that $\sum_{k=1}^{\infty} \beta_k = \infty$.

There are several research studies that conduct computational experiments using variants of step-length rule (2.1c), making recommendations for some specially structured problems (see Polyak, 1969, and Held et al., 1974). 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.

Strictly speaking, there is no practical subgradient step-length rule based procedure that assumes no knowledge whatsoever of the optimal objective value, and so, which can be used for general nondifferentiable optimization problems. Therefore, in this research, we present a new variable target value method, under the assumption that the only known bounds on the optimal value are $\pm\infty$. This new, non-monotone, *Variable Target Value Algorithm (VTVM)* updates target values whenever it is necessary by increasing or decreasing the old target value in an outer loop, depending on the information obtained during the process of an inner loop of the algorithm. The algorithm is proven to generate a sequence of incumbent solutions that converges to a (near)-optimal solution, and is shown to be computationally effective.

The second crucial point to note is that no matter what step-length rule is implemented, the computational performance of the pure-subgradient method is limited simply due to the direction of motion used. As with the steepest descent direction for the differentiable case, the anti-subgradient direction for the nondifferentiable case also results in a zig-zagging phenomenon that might manifest itself at any stage of the subgradient algorithm, causing the procedure to crawl toward optimality. As a tool to overcome this difficulty, a conjugate subgradient concept has been introduced by

imitating the conjugate gradient methods for the differentiable case. A conjugate gradient direction for the differentiable case is computed by combining the current anti-gradient with the previous direction, while forcing some conjugacy requirement (see Hestenes and Stiefel, 1952, and Fletcher and Reeves, 1959). However, for the nondifferentiable case, this concept is used only as a strategy to deflect the subgradient, since conjugacy may no longer be meaningful. Accordingly, the direction of motion $\mathbf{d}_k$ at $\mathbf{x}_k$ is computed as

$$\mathbf{d}_k = -\mathbf{g}_k + \psi_k \mathbf{d}_{k-1} \tag{2.2}$$

where $\psi_k \geq 0$ is a deflection parameter, $\mathbf{g}_k$ is a subgradient of f at $\mathbf{x}_k$, and $\mathbf{d}_{k-1}$ is the previous direction with $\mathbf{d}_0 \equiv \mathbf{0}$. Then, the new iterate is computed according to

$$\mathbf{x}_{k+1} = P_X[\mathbf{x}_k + \lambda_k \mathbf{d}_k] \tag{2.3}$$

where λ_k is a suitable step-length and $P_X(\cdot)$ denotes the projection operation onto the set X.

There are some well-known, and extensively tested, subgradient deflection algorithms of the type (2.2), namely, the *Modified Gradient Technique (MGT)* proposed by Camerini et al. (1975) and the *Average Direction Strategy (ADS)* of Sherali and Ulular (1989). These methods have been shown to be superior to the pure-subgradient method in practice along with a specially designed step-length selection rule of the type (2.1c). However, the direction of motion generated by MGT frequently turns out to be selected as simply the anti-subgradient direction itself, especially at the final stages of the procedure, thereby inhibiting the rate of convergence. Sherali and Ulular (1989) demonstrate the superiority of their scheme to Camerini et al.'s procedure, and particularly, to pure-subgradient based approaches. Despite its computa-

tional superiority, ADS merely bisects the angle between the anti-subgradient and the previous direction, regardless of any local information.

The above cited literature concentrates on specific strategies, while a general convergence behavior analysis is not as yet available. Kim and Ahn (1991) view the direction generated by conjugate subgradient procedure as a nonnegative linear combination of the collection of anti-subgradients obtained over the previous iterations, and prove a related convergence theorem, but using the step-length rules (2.1a) and (2.1b) along with ε-subgradient concepts under the assumption that $\varepsilon \to 0$. On the other hand, in this paper, we provide general convergence theorems for ordinary conjugate subgradient algorithms using the attractive step-length rule (2.1c), in which the only condition imposed on the deflection parameter is nonnegativity. Since these theorems permit considerably more freedom in selecting the deflection parameters under suitable assumptions on the step-length rule, we design a new deflection strategy based on some desirable motivation. This new strategy, which we call an *Optimally Deflected Subgradient Algorithm (ODSA)*, generates the direction of motion by rotating the anti-subgradient optimally toward the direction leading to an estimated (artificial) optimal solution, where the latter is updated in the algorithmic process using target values.

The remainder of this chapter is organized as follows. In Section 2, we present convergence theorems for general deflected subgradient algorithms using the step-length rule (2.1c) for both upper and lower bound target value cases. Based on these theorems, we present a new deflection strategy, Algorithm ODSA, in Section 3. Section 4 describes a new variable target value method, Algorithm VTVM, which assumes no *a priori* knowledge on any bounds on the optimal objective function value,

and we establish its convergence. Finally, in Section 5, we present results on some computational experiments along with recommendations for selecting deflection and step-length parameter values.

2.2 Convergence Theorems for Fixed Target Values

The literature contains several convergence results for the step-length rule (2.1c) used in conjunction with both the pure-subgradient method and some specific deflected subgradient methods. Polyak(1969) provides the original convergence theorems for the pure-subgradient method using a fixed target value under the assumptions

$$0 < \varepsilon_1 \leq \beta_k \leq \varepsilon_2 < 2 \tag{2.4}$$

and $\|\mathbf{g}_k\| < M$ for all k, for some M. Later, Allen et al. (1987) used the assumptions $0 < \beta_k \leq \varepsilon_2 < 2$ and $\sum_{k=1}^{\infty} \beta_k = \infty$ instead of (2.4), and generalized Polyak's results for the pure-subgradient method.

Camerini et al. (1975) provide a similar convergence theorem for their subgradient deflection strategy, MGT. According to their convergence theorem, for a given fixed target value $w > f^*$ and adopting a step-length λ_k along the deflected subgradient direction, where

$$\varepsilon \frac{f_k - w}{\|\mathbf{d}_k\|^2} \leq \lambda_k \leq \frac{f_k - w}{\|\mathbf{d}_k\|^2}$$

for some $0 < \varepsilon < 1$, the algorithm generates a sequence $\{\mathbf{x}_k\}$ for which either there exists a $\hat{k}$ such that $\mathbf{x}_{\hat{k}} \in \{\mathbf{x} : f(\mathbf{x}) \leq w\}$, or $\{\mathbf{x}_k\} \to \bar{\mathbf{x}}$ for some $\bar{\mathbf{x}}$ belonging to the boundary of this level set. Later, Sherali and Ulular (1989) proved a primal-dual convergence theorem for their subgradient deflection strategy ADS used in conjunction with a proposed primal-dual algorithm. In their algorithm, the step-lengths of the form (2.1c) are computed by using the difference between the current primal and dual function values, in lieu of using target values w.

In the present section, we provide convergence results for general deflected subgradient algorithms wherein the deflection parameter simply satisfies $\psi_k \geq 0$ for all k. Our results turn out to be similar to those presented by Allen et al. for the pure-subgradient algorithm, except that the choice of ε_2 in (2.4) is somewhat more restricted. Therefore, these theorems not only extend Polyak's results to deflected subgradient methods, but also generalize Camerini et al.'s and Sherali and Ulular's convergence theorems for fixed (both upper and lower bounding) target values.

To begin our discussion, suppose that the algorithm (2.2) and (2.3) used in conjunction with the step-length rule (2.1c) generates a sequence of iterates $\{\mathbf{x}_k\}$ for some deflection parameters $\psi_k \geq 0$ for all k, and for some fixed target value w. Assume that $\|\mathbf{g}_k\| > 0$ for all k, for otherwise, if $\mathbf{g}_k = \mathbf{0}$ for any k, then $\mathbf{x}_k$ is an optimal solution. Consider the following results.

Lemma 2.1 If $w \geq f^*$ and $0 < \beta_k \leq 1$ for all k, then we have

$$\mathbf{d}_{k-1}^t(\mathbf{x}^* - \mathbf{x}_k) \geq 0 \quad \text{for all } k \tag{2.5}$$

Proof Note that this is trivially true for $k = 1$, since $\mathbf{d}_0 = \mathbf{0}$. By induction, consider any $k \geq 2$, and assume that $\mathbf{d}_{k-2}^t(\mathbf{x}^* - \mathbf{x}_{k-1}) \geq 0$. Using the definition of $\mathbf{d}_{k-1}$ and λ_k, and the Cauchy-Schwarz inequality, along with the convexity of f, we have the following string:

$$\begin{aligned}
\mathbf{d}_{k-1}^t(\mathbf{x}^* - \mathbf{x}_k) &= \mathbf{d}_{k-1}^t(\mathbf{x}^* - \mathbf{x}_{k-1}) + \mathbf{d}_{k-1}^t(\mathbf{x}_{k-1} - \mathbf{x}_k) \\
&= -\mathbf{g}_{k-1}^t(\mathbf{x}^* - \mathbf{x}_{k-1}) + \psi_{k-1}\mathbf{d}_{k-2}^t(\mathbf{x}^* - \mathbf{x}_{k-1}) + \mathbf{d}_{k-1}^t(\mathbf{x}_{k-1} - \mathbf{x}_k) \\
&\geq f(\mathbf{x}_{k-1}) - f(\mathbf{x}^*) + \mathbf{d}_{k-1}^t(\mathbf{x}_{k-1} - \mathbf{x}_k) \\
&\geq \beta_{k-1}(f_{k-1} - w) - \|\mathbf{d}_{k-1}\| \, \|\mathbf{x}_k - \mathbf{x}_{k-1}\| \\
&= \beta_{k-1}(f_{k-1} - w) - \|\mathbf{d}_{k-1}\| \, \|P_X(\mathbf{x}_{k-1} + \lambda_{k-1}\mathbf{d}_{k-1}) - \mathbf{x}_{k-1}\|
\end{aligned}$$

Now, since $\mathbf{x}_{k-1} \in X$, we have

$$\begin{aligned}
\mathbf{d}_{k-1}^t(\mathbf{x}^* - \mathbf{x}_k) &\geq \beta_{k-1}(f_{k-1} - w) - \|\mathbf{d}_{k-1}\|\|\mathbf{x}_{k-1} + \lambda_{k-1}\mathbf{d}_{k-1} - \mathbf{x}_{k-1}\| \\
&= \beta_{k-1}(f_{k-1} - w) - \lambda_{k-1}\|\mathbf{d}_{k-1}\|^2 \\
&= \beta_{k-1}(f_{k-1} - w) - \beta_{k-1}(f_{k-1} - w) = 0 \qquad (2.6)
\end{aligned}$$

Hence, the assertion (2.5) holds for all k by mathematical induction and this completes the proof. ■

Corollary 2.1 Instead of the assumptions $w \geq f^*$ and $0 < \beta_k \leq 1$, for all k, if w and β_k are chosen to satisfy the inequality

$$f_k - f^* \geq \beta_k(f_k - w) \quad \text{for all} \quad k\,,$$

where $f_k \equiv f(\mathbf{x}_k)$, then the assertion of Lemma 2.1 continues to hold.

Proof Evident from the string of deductions in Lemma 2.1. ∎

Theorem 2.1 Suppose that $w \geq f^*$, $0 < \varepsilon_1 \leq \beta_k \leq 1$ for some $\varepsilon_1 > 0$, and that there exists a $M > 0$ such that $\|\mathbf{d}_k\| < M$ for all k. Then, we have either (i) there exists a k such that $f_k \leq w$, or (ii) $f_k > w$ for all k and $\lim_{k\to\infty} f_k = w$.

Proof Clearly, either $f_k \leq w$ for some k, or else $f_k > w$ for all k. Assuming the latter, it is sufficient to show that $\lim_{k\to\infty} f_k = w$. We now have,

$$\|\mathbf{x}^* - \mathbf{x}_{k+1}\|^2 = \|\mathbf{x}^* - P_X(\mathbf{x}_k + \lambda_k \mathbf{d}_k)\|^2 \leq \|\mathbf{x}^* - \mathbf{x}_k - \lambda_k \mathbf{d}_k\|^2$$

$$= \|\mathbf{x}^* - \mathbf{x}_k\|^2 + \lambda_k^2 \|\mathbf{d}_k\|^2 - 2\lambda_k \mathbf{d}_k^t(\mathbf{x}^* - \mathbf{x}_k)$$

$$= \|\mathbf{x}^* - \mathbf{x}_k\|^2 + \lambda_k^2 \|\mathbf{d}_k\|^2 + 2\lambda_k \mathbf{g}_k^t(\mathbf{x}^* - \mathbf{x}_k) - 2\lambda_k \psi_k \mathbf{d}_{k-1}^t(\mathbf{x}^* - \mathbf{x}_k)$$

$$\leq \|\mathbf{x}^* - \mathbf{x}_k\|^2 + \lambda_k^2 \|\mathbf{d}_k\|^2 + 2\lambda_k(f^* - f_k) \tag{2.7}$$

The last inequality holds from (2.5) and the definition of a subgradient. Now,

$$\lambda_k(f^* - f_k) \leq \lambda_k(w - f_k) = -\beta_k \frac{(f_k - w)^2}{\|\mathbf{d}_k\|^2}$$

Thus, using (2.1c), we have

$$\|\mathbf{x}^* - \mathbf{x}_{k+1}\|^2 \leq \|\mathbf{x}^* - \mathbf{x}_k\|^2 + \beta_k^2 \frac{(f_k - w)^2}{\|\mathbf{d}_k\|^2} - 2\beta_k \frac{(f_k - w)^2}{\|\mathbf{d}_k\|^2}$$

$$= \|\mathbf{x}^* - \mathbf{x}_k\|^2 + \beta_k(\beta_k - 2) \frac{(f_k - w)^2}{\|\mathbf{d}_k\|^2}$$

$$< \|\mathbf{x}^* - \mathbf{x}_k\|^2$$

Hence, $\{\|\mathbf{x}^* - \mathbf{x}_k\|^2\}$ is a bounded monotone decreasing sequence and is therefore convergent. Consequently, we have,

$$\beta_k(\beta_k - 2) \frac{(f_k - w)^2}{\|\mathbf{d}_k\|^2} \to 0 \quad \text{as} \quad k \to \infty$$

Since $\|\mathbf{d}_k\|$ is bounded and $\beta_k(\beta_k - 2)$ is bounded away from 0 for all k, we have $\lim_{k \to \infty} f_k = w$, and this completes the proof. ■

Remark 2.1 The assertion of the above theorem can be written to claim that for any given $\varepsilon > 0$, in a finite number of iterations, we can find a k such that f_k is within ε of w, that is,

$$f_k \leq w + \varepsilon. \tag{2.8}$$

The foregoing choice of the target as an upper bound on f^*, yet less than any known incumbent value, is impractical since the optimal objective value is not known in general. This behooves us to consider the case when the target w is chosen to be a lower bound on f^*, which is often realizable in many situations. The following theorem points out the limiting behavior for this case.

Theorem 2.2 Suppose that $w < f^*$ and that there exists a $M > 0$ such that $\|\mathbf{d}_k\| \leq M$ for all k. Given any $\varepsilon > 0$, if β_k is selected as

$$\beta_k = \frac{\varepsilon}{f_k - w} > 0 \tag{2.9}$$

for all k, then there exists a k such that

$$f_k \leq f^* + \varepsilon \tag{2.10}$$

Proof On the contrary, suppose that there is no such k satisfying (2.10), that is,

$$f_k - f^* > \varepsilon = \beta_k(f_k - w) \quad \text{for all} \quad k. \tag{2.11}$$

Thus, by Corollary 2.1, we have $\mathbf{d}_{k-1}^t(\mathbf{x}^* - \mathbf{x}_k) \geq 0$ for all k, and so, the string leading to (2.7) of Theorem 2.1 holds true. Therefore, using (2.1c) in (2.7), along with (2.9) and (2.11), we have

$$\|\mathbf{x}^* - \mathbf{x}_k\|^2 - \|\mathbf{x}^* - \mathbf{x}_{k+1}\|^2 \geq \lambda_k\left[-\beta_k(f_k - w) + 2(f_k - f^*)\right] > \lambda_k\varepsilon \geq \frac{\varepsilon^2}{\|\mathbf{d}_k\|^2} \geq \frac{\varepsilon^2}{M^2}$$

since $\lambda_k = \varepsilon/\|\mathbf{d}_k\|^2$. Hence, by taking telescopic sums, we have

$$\|\mathbf{x}^* - \mathbf{x}_1\|^2 \geq \|\mathbf{x}^* - \mathbf{x}_1\|^2 - \|\mathbf{x}^* - \mathbf{x}_{t+1}\|^2 > \frac{t\varepsilon^2}{M^2} \quad \text{for all} \quad t \geq 1.$$

But this leads to $\|\mathbf{x}^* - \mathbf{x}_1\|^2 > \infty$ as $t \to \infty$, a contradiction, and hence, the proof is complete. ■

Note that the parameter choice (2.9) for the fixed target value case of Theorem 2.2 can lead to short step sizes, given a small convergence tolerance $\varepsilon > 0$. Fortunately, in

the variable target value method presented in Section 4, we are able to circumvent this choice and continue to use (2.1c) as for Theorem 2.1, in this case as well, and yet establish convergence to optimality. More importantly, note that the foregoing results permit considerable flexibility in the choice of the deflection parameter ψ. We exploit this flexibility in the next section, and prescribe a new deflection strategy motivated by optimality considerations.

2.3 An Optimally Deflected Subgradient Algorithm

In this section, motivated by Theorems 2.1 and 2.2, we prescribe a value of $\psi_k \geq 0$ to be used in (2.2), (2.3) under the step-length rule (2.1c), with the intent of obtaining desirable search directions that tend to avoid the zig-zagging phenomenon inherent in pure-subgradient methods.

Toward this end, at iteration k of some algorithmic process, suppose that a target value w is given such that $f_k > w \geq f^*$, so that $L_w = \{\mathbf{x}_w : f(\mathbf{x}_w) \leq w\} \neq \phi$. (In particular, note that $\mathbf{x}^* \in L_w$, where $f^* = f(\mathbf{x}^*)$, and that the following derivation holds just as well with $\mathbf{x}_w \equiv \mathbf{x}^*$.) Effectively, we want to find a ψ_k that makes the angle θ between $\mathbf{d}_k$ and $(\mathbf{x}_w - \mathbf{x}_k)$ as small as possible for some $\mathbf{x}_w \in L_w$. This can be written as the following optimization problem:

$$\underset{\psi_k \geq 0}{\text{Sup}} \left[\cos\ \theta = \frac{\mathbf{d}_k^t(\mathbf{x}_w - \mathbf{x}_k)}{\|\mathbf{d}_k\|\|\mathbf{x}_w - \mathbf{x}_k\|} \right]$$

For simplicity, let us drop subscripts and define

$$\psi = \psi_k,\ \bar{\mathbf{d}} = \mathbf{d}_{k-1},\ \mathbf{g} = \mathbf{g}_k,\ \ r = -\mathbf{g}_k^t(\mathbf{x}_w - \mathbf{x}_k) \geq f_k - f(\mathbf{x}_w),\ \text{ and } s = \bar{\mathbf{d}}^t(\mathbf{x}_w - \mathbf{x}_k).$$

Then, $r \geq 0$ since $f_k \geq w \geq f(\mathbf{x}_w)$, and we have $\mathbf{d}_k^t(\mathbf{x}_w - \mathbf{x}_k) = r + \psi s$. Moreover, the foregoing optimization problem for determining ψ becomes

$$\operatorname*{Sup}_{\psi \geq 0} \frac{r + s\psi}{\| -\mathbf{g} + \psi\bar{\mathbf{d}}\| \|\mathbf{x}_w - \mathbf{x}_k\|}$$

This is equivalent to the problem

$$\operatorname*{Sup}_{\psi \geq 0} \left[\Phi(\psi) \equiv \frac{r + s\psi}{\| -\mathbf{g} + \psi\bar{\mathbf{d}}\|} \right] \tag{2.12}$$

since $\|\mathbf{x}_w - \mathbf{x}_k\|$ is independent of the choice of ψ, for any assumed $\mathbf{x}_w \in L_w$. Note that

$$\Phi'(\psi) = \frac{\left[-(\bar{\mathbf{d}}^t\mathbf{g})s - \|\bar{\mathbf{d}}\|^2 r \right]\psi + \|\mathbf{g}\|^2 s + (\bar{\mathbf{d}}^t\mathbf{g})r}{\| -\mathbf{g} + \psi\bar{\mathbf{d}}\|^3}$$

and so,

$$\Phi'(\bar{\psi}) = 0 \Leftrightarrow \bar{\psi} = \frac{(\bar{\mathbf{d}}^t\mathbf{g})r + \|\mathbf{g}\|^2 s}{(\bar{\mathbf{d}}^t\mathbf{g})s + \|\bar{\mathbf{d}}\|^2 r} \tag{2.13}$$

Thus, if $\bar{\psi} > 0$, then we choose the supremum as the largest of the values

$$\Phi(0) = \frac{r}{\|\mathbf{g}\|},\quad \lim_{\psi \to \infty} \Phi(\psi) = \frac{s}{\|\bar{\mathbf{d}}\|},\quad \text{and} \quad \Phi(\bar{\psi}). \tag{2.14}$$

On the other hand, if $\bar{\psi} < 0$, then we do not have to consider $\Phi(\bar{\psi})$, and we need to compare only the first two values in (2.14). If the value $s/\|\bar{\mathbf{d}}\|$ is the largest in (2.14) for either case, we merely take the old direction of motion as the new direction. (This

can be caused by having taken too small a step-length at the previous iteration.) Otherwise, we set $\psi^* = 0$ if the maximum value is attained by $r/\|\mathbf{g}\|$, and $\psi^* = \overline{\psi}$ if $\Phi(\overline{\psi})$ gives the largest value when $\overline{\psi} > 0$.

To actually implement this prescribed choice of ψ^*, we need to estimate r and s which are not available in practice. Indeed, these values can be iteratively updated by a suitable algorithmic design. Note that, for each k, adopting appropriate subscripts k,

$$r_k \equiv -\mathbf{g}_k^t(\mathbf{x}_w - \mathbf{x}_k) \geq f_k - f(\mathbf{x}_w) \geq f_k - w \tag{2.15}$$

and, since $r_{k-1} + \psi_{k-1}s_{k-1} \equiv \mathbf{d}_k^t(\mathbf{x}_w - \mathbf{x}_{k-1})$, we can write

$$s_k = \mathbf{d}_{k-1}^t(\mathbf{x}_w - \mathbf{x}_k) = r_{k-1} + \psi_{k-1}s_{k-1} - \mathbf{d}_{k-1}^t(\mathbf{x}_k - \mathbf{x}_{k-1}) \tag{2.16}$$

From (2.15) and (2.16), and motivated by the role of condition (2.5), we can derive a systematic scheme for updating r_k and s_k as follows. We compute r_k according to

$$r_k = \mu_k(f_k - w) \quad \text{for all} \quad k \geq 1 \tag{2.17}$$

for some $\mu_k \geq 1$. (Computationally, we have determined that the choice $\mu_k = 1 + 0.5e^{(1-k)}$ performs well.) Furthermore, we compute s_k by letting $s_1 = 0$, and for all $k \geq 2$, we compute

$$s_k = \text{maximum } \{r_{k-j} + \psi_{k-j}s_{k-j} - \mathbf{d}_{k-j}^t(\mathbf{x}_k - \mathbf{x}_{k-j}),\ 0\} \tag{2.18}$$

where $j \geq 1$ is the smallest integer for which a finite value for ψ_{k-j} was previously obtained. For example, suppose that the direction of motion was taken as the previous one at the $(k-1)$-st iteration; in other words, suppose that $s_{k-1}/\|\mathbf{d}_{k-2}\|$ was the

largest value in (2.14), so that we adopted $\mathbf{d}_{k-1} = \mathbf{d}_{k-2}$, and suppose that $\psi_{k-2} < \infty$. Then in order to avoid using $\psi_{k-1} \to \infty$ in (2.16), we may treat the last two consecutive iterations as one iteration because their directions of motion are the same. Hence, the subscripts $(k-1)$'s in (2.16) are changed to $(k-2)$'s in (2.18) for the computation of s_k, in addition to enforcing $s_k \geq 0$ as motivated by (2.3) and the definition of s_k.

Now, the algorithm can be summarized as follows. In this algorithm, we assume that a fixed target value is given as an upper bound on f^*, and we will later relax this assumption in Section 4. Also, we assume as before that $\|\mathbf{d}_k\| \leq M$ for all k, and for some M.

Optimally Deflected Subgradient Algorithm (ODSA)

Initialization Choose a starting solution $\mathbf{x}_1$ and evaluate f_1 and $\mathbf{g}_1$. Set $\mathbf{d}_0 = \mathbf{0}$.

Step 1 Find $\psi_k = \psi^*$ as in (2.13) and (2.14) using (2.17) and (2.18) to estimate r_k and s_k. Compute the direction of motion according to

$$\mathbf{d}_k = -\mathbf{g}_k + \psi_k \mathbf{d}_{k-1}$$

Step 2 Find the new iterate

$$\mathbf{x}_{k+1} = P_X[\mathbf{x}_k + \lambda_k \mathbf{d}_k]$$

where the step-length λ_k is given by

$$\lambda_k = \beta_k \frac{f_k - w}{\|\mathbf{d}_k\|^2} \quad \text{and} \quad 0 < \varepsilon_1 \leq \beta_k \leq 1 \quad \text{for all} \quad k$$

If any suitable termination criterion holds, then stop the algorithmic process. Note that several termination rules such as (i) $f_k \leq w$, (ii) $k = K$ for some preassigned number of iterations K, or (iii) $\|\mathbf{g}_k\| \leq \varepsilon_3$ for some sufficiently small $\varepsilon_3 > 0$ can be employed here. Otherwise, increment k by one and return to Step 1.

Algorithm ODSA uses a fixed target value for computing the deflection parameter and its convergence is implied by Theorem 2.1. Note here that if $\psi_k \to \infty$ in (2.14), then we put $\psi_k = \overline{M}$ for some sufficiently large $\overline{M} > 0$ so that the direction of motion still has the form of $\mathbf{d}_k = -\mathbf{g}_k + \psi_k \mathbf{d}_{k-1}$ with $\psi_k \geq 0$ for all k. Moreover, observe that the convergence property of Theorem 2.2 also holds whenever the target value is given as a lower bound on f^*. Note that in this case, although the last inequality in (2.15) does not hold, we still have $r_k > 0$ by (2.17) since $f_k > w$. In fact, because $f_k - w > f_k - f(\mathbf{x}_w)$ in this case, and by (2.15) we want $r_k \geq f_k - f(\mathbf{x}_w)$, the choice of r_k in (2.17) automatically provides for some of this required compensation.

As mentioned earlier, and similar to the pure-subgradient algorithm, we need some varying target value strategy for implementing this scheme in practice, when no known target value that is satisfactorily close to f^* is available *a priori*. Toward this end, we provide a variable target value method that can be used along with any conjugate subgradient based direction strategy, including that of ODSA.

2.4 A Variable Target Value Method

Since the optimal objective value is usually not known, any (conjugate) subgradient algorithm employing the step-length rule (2.1c) for a fixed target value, in general, cannot be guaranteed to generate a sequence of iterates that converges to an optimum or even a near-optimum. In this regard, the variable target value method is one way to not only recover from the blindness in the choice of the target value, but also, to accelerate convergence. Bazaraa and Sherali (1981) present a variable target rule for solving Lagrangian duals of discrete problems. Their target value is computed by using a convex combination of a fixed upper bound (which is available via a primal feasible solution) and the incumbent dual solution, based on a convex combination weighting rule stated as a function of the iterations. Recently, Kim et al. (1991) present a similar rule for minimizing strongly convex functions, given an initial lower bound $L_1 < f^*$. Also, for general convex functions, their method requires an initial lower bound on f^*, plus an upper bound estimate on $\|\mathbf{x}_1 - \mathbf{x}^*\|$. Therefore, these existing algorithms are not practically implementable for general nondifferentiable optimization problems for which reasonable bounds of this type might not be available *a priori*. In this paper, without using any information about the optimal value and the solution space, we propose a new variable target value method that generates a (nonmonotone) sequence of target values in order to control the step-length. Furthermore, this method affords a reasonable stopping criterion, and generates a sequence of solutions that guarantees ε-optimality for any given $\varepsilon > 0$.

Suppose that at some *outer loop iteration* ℓ and an *inner loop iteration* k, we have a current solution $\mathbf{x}_k$, a direction of motion $\mathbf{d}_k$, a target value w_ℓ, and a best known ob-

jective value z_k. If $w_\ell \geq f^*$, then from either the convergence theorems of Polyak (1967, 1969) and Allen et al. (1987) using the pure subgradient algorithm, or from Theorem 2.1 for subgradient deflection algorithms, we can find a j (within the inner loop iterations) such that $f_{k+j} \leq w_\ell + \varepsilon$ for some $\varepsilon > 0$, where $f(\mathbf{x}_k)$ is denoted by f_k for all k. When this occurs, the target value is decreased in an outer loop update. On the other hand, there might be a case when the algorithm fails to hit this ε-neighborhood of the target value over some reasonable, preassigned, number of iterations, even if the target satisfies the assumptions of Theorem 2.1. (Note that this will necessarily be the case if the chosen target is actually a (strict) lower bound on f^*.) In these cases, the target value will be increased in an outer loop update. Since this strategy controls the step-length by tracking the best obtained objective value, it is more practical than other methods that change the step-length by revising the parameter β alone. Moreover, this scheme assumes no known information about optimality, and it can be used with any direction finding strategy. The following algorithmic statement formalizes the above concepts.

Variable Target Value Algorithm (VTVM)

Initialization Select parameters ε_0, ε, $\sigma^{(1)}$, $\sigma^{(2)}$, $\gamma^{(1)}$, $\gamma^{(2)}$, $\beta^{(1)}$, and $\beta^{(2)}$ as recommended below, along with termination tolerances $\bar{\tau}$ and $k_{\max}$, which respectively represent the maximum number of consecutive target increases permitted, and a limit on the number of iterations. Select a starting solution $\mathbf{x}_0 \in X$, compute its objective function value f_0, and let the initial direction of motion be $\mathbf{d}_0 = -\mathbf{g}_0$, where $\mathbf{g}_0$ is a subgradient of f at $\mathbf{x}_0$. If $\|\mathbf{g}_0\| < \varepsilon_0$, then stop with $\mathbf{x}_0$ as a near-optimal solution. Otherwise, set $\mathbf{x}^* = \mathbf{x}_0$ and $\mathbf{g}^* = \mathbf{g}_0$, and record $z_0 = f_0$ as the best known objective function value. Initialize the outer loop counter $\ell = 1$, and compute the initial target value $w_1 = \max\{\text{LB},$

$f_0 - \|\mathbf{d}_0\|^2/2\}$, where LB is any known lower bound on f^*, being taken as $-\infty$ if no such lower bound is available. Also, select the initial acceptance tolerance for measuring proximity to the present target value as $\varepsilon_1 = (\sigma^{(1)} + \sigma^{(2)})(f_0 - w_1)$. Set the counter of consecutive target increases to $\tau = 0$, the consecutive failure counter to $\gamma = 0$, and the inner loop counter to $k = 0$. Put $\Delta = 0$, where Δ measures accumulated improvements within the inner loop iterations.

Step 1(a) (Outer Loop Parameter Updates) Compute $\sigma_\ell = \sigma^{(1)} + \sigma^{(2)}e^{(1-\ell)}$, $\bar{\gamma}_\ell = \gamma^{(1)} + \gamma^{(2)}e^{(1-\ell)}$, and $\beta_\ell = \beta^{(1)} + \beta^{(2)}e^{(1-\ell)}$.

Step 1(b) (Inner Loop Main Iteration) If $k = k_{\max}$, stop. Else, compute the step-length

$$\lambda_k = \beta_\ell \frac{f_k - w_\ell}{\|\mathbf{d}_k\|^2}$$

Find the new iterate $\mathbf{x}_{k+1} = P_X[\mathbf{x}_k + \lambda_k \mathbf{d}_k]$, evaluate its objective function value f_{k+1}, and find a subgradient $\mathbf{g}_{k+1}$ of f at $\mathbf{x}_{k+1}$. If $\|\mathbf{g}_{k+1}\| < \varepsilon_0$, terminate the algorithm with $\mathbf{x}_{k+1}$ as a (near) optimal solution. Otherwise, compute the new direction of motion $\mathbf{d}_{k+1}$ at $\mathbf{x}_{k+1}$ via any desired strategy. If $f_{k+1} < z_k$, update $\Delta \leftarrow \Delta + (z_k - f_{k+1})$, and go to Step 2(a). Otherwise, go to Step 3(a).

Step 2(a) (Improvement Case) Put $\gamma = 0$, $z_{k+1} = f_{k+1}$, and update $\mathbf{x}^* = \mathbf{x}_{k+1}$ and $\mathbf{g}^* = \mathbf{g}_{k+1}$. If $z_{k+1} \leq w_\ell + \varepsilon_\ell$, then go to Step 2(b). Otherwise, increment k by one, and return to Step 1(b).

Step 2(b) (Outer Loop Update; Decrease Target Value and Adjust Acceptance Tolerance) Let $\eta = 0.5 + 0.5e^{-\ell/10}$ and compute a new target value as

$$w_{\ell+1} = (z_{k+1} - \varepsilon_\ell) - \eta\Delta$$

and let

$$\varepsilon_{\ell+1} = \max\{(z_{k+1} - w_{\ell+1})\sigma_\ell,\ \varepsilon\}$$

Put $\tau = 0$ and $\Delta = 0$, increment ℓ and k by one, and return to Step 1(a).

Step 3(a) (Failure Case) Put $z_{k+1} = z_k$, and increment γ by one. If $\gamma \geq \bar{\gamma}_\ell$, go to Step 3(b). Otherwise, increment k by one, and return to Step 1(b).

Step 3(b) (Outer Loop Update; Increase Target Value and Adjust Acceptance Tolerance) Compute a new target value as

$$w_{\ell+1} = \frac{(z_{k+1} - \varepsilon_\ell) + w_\ell}{2}$$

and let

$$\varepsilon_{\ell+1} = \max\{(z_{k+1} - w_{\ell+1})\sigma_\ell,\ \varepsilon\}$$

Increment τ by one. If $\tau = \bar{\tau}$, terminate the algorithm. Otherwise, put $\gamma = 0$ and $\Delta = 0$, increment ℓ and k by one, and return to Step 1(a).

Recommended parameter values

Case 1 $k_{\max} \in [200,\ 1000]$ and $\bar{\tau} = \infty$

$\varepsilon_0 = 10^{-6}$ [tolerance on subgradient norm]

$\varepsilon = 10^{-1}$ [overall convergence tolerance]

$(\sigma^{(1)}, \sigma^{(2)}) = (0.1,\ 0.5)$ or $(0.15,\ 0)$ [acceptance tolerance adjustment parameter]

$(\gamma^{(1)}, \gamma^{(2)}) = (30$ to $50,\ 10)$ [consecutive failure tolerance parameter]

$(\beta^{(1)}, \beta^{(2)}) = (0.25,\ 0.75)$ [step-length factor parameter]

Case 2 $k_{max} = \infty$ and $\bar{\tau} = 15$

$\varepsilon_0 = 10^{-6}$ [tolerance on subgradient norm]

$\varepsilon = 10^{-1}$ [overall convergence tolerance]

$(\sigma^{(1)}, \sigma^{(2)}) = (0.1,\ 0.5)$ or $(0.15,\ 0)$ [acceptance tolerance adjustment parameter]

$(\gamma^{(1)}, \gamma^{(2)}) = (30,\ 10)$ [consecutive failure tolerance parameter]

$(\beta^{(1)}, \beta^{(2)}) = (0.25,\ 0.75)$ [step-length factor parameter]

Remark 2.2 Note that a restarting technique is often an important computational ingredient of subgradient procedures (see Held et al., 1974, Bazaraa and Sherali, 1981, and Sherali and Ulular, 1989). In the same spirit, for VTVM, whenever the target needs to be increased at Step 3(b) due to $\bar{\gamma}_\ell$ consecutive failures, we can restart the algorithm using the best iterate obtained thus far. Specifically, we set $\mathbf{x}_{k+1} = \mathbf{x}^*$ and $\mathbf{g}_{k+1} = \mathbf{g}^*$. Moreover, if a subgradient deflection strategy is in use, then an anti-subgradient (pure-subgradient) direction can be employed at each restart, that is, $\mathbf{d}_{k+1} = -\mathbf{g}^*$.

We now establish the convergence bahavior of the foregoing algorithm.

Lemma 2.2 Consider any conjugate subgradient algorithm in which the direction of motion is of the form (2.2) with the step-length rule (2.1c). Suppose that the target values, denoted w_k, vary with each inner iteration k, but it turns out that there exist values $\underline{w}$ and $\bar{w}$ such that

$$f^* \leq \underline{w} < w_k < \overline{w} \leq f_k \quad \text{for all} \quad k.$$

If $0 < \varepsilon_1 \leq \beta_k \leq 1$ and $\|\mathbf{d}_k\| < M$ for all k, and for some $M > 0$, then we must have both $\{w_k\}$ and $\{f_k\}$ tending to $\overline{w}$ in the limit as $k \to \infty$.

Proof Let us examine the modifications of Lemma 2.1 and Theorem 2.1 under the assumption that w_k varies in the range $f^* \leq \underline{w} < w_k < \overline{w}$, rather than remains fixed at some $w \geq f^*$. By using $w = \underline{w}$ in the first appearance of w in Lemma 2.1, the final statement (2.6) changes to

$$\beta_{k-1}(f_{k-1} - \underline{w}) - \beta_{k-1}(f_{k-1} - w_{k-1}) = \beta_{k-1}(w_{k-1} - \underline{w}) > 0$$

Hence, the assertion of Lemma 2.1 holds. Now, in the proof of Theorem 2.1, we get as before that

$$\lim_{k \to \infty} \frac{(f_k - w_k)^2}{\|\mathbf{d}_k\|^2} = 0$$

Under the hypothesis of the present lemma, this can happen only if both $\{f_k\} \to \overline{w}$ and $\{w_k\} \to \overline{w}$, and this completes the proof. ■

Now, observe that $\{z_k\}$ is a monotone nonincreasing sequence that is bounded below by f^*, and is hence a convergent sequence. Let $\bar{z}$ be the limit of this sequence, and consider the following result.

Lemma 2.3 Suppose that Algorithm VTVM is operated in conjunction with the conjugate subgradient algorithm of the form (2.2). If $k_{\max} = \bar{\tau} = \infty$, $\varepsilon_\ell = \varepsilon$ for all ℓ, and if

$\bar{z} - \varepsilon - f^* \geq \delta > 0$ for some δ satisfying $0 < \delta < \varepsilon$, then there exists an outer iteration $\tilde{\ell}$ such that

$$\bar{z} - \varepsilon - \delta < w_\ell < \bar{z} - \varepsilon + \delta \quad \text{for all} \quad \ell \geq \tilde{\ell}.$$

Proof Since $\{z_k\} \to \bar{z}$, there exists a K_2 such that $z_k < \bar{z} + \delta$ for all $k \geq K_2$. Let us examine the algorithmic process once we have $k \geq K_2$. If $w_\ell \leq \bar{z} - \varepsilon - \delta$ for some outer loop ℓ, then the algorithm should fail to attain ε-neighborhood proximity of the target value without increasing it, since $w_\ell + \varepsilon \leq z_k - \delta < z_k$ for all k. Thus, by successively increasing the target value, we can find an outer iteration $\tilde{\ell}$, such that $w_{\tilde{\ell}} > \bar{z} - \varepsilon - \delta$. On the other hand, for any outer iteration ℓ, if we have $w_\ell \geq \bar{z} - \varepsilon + \delta$, then when $(k+1) \geq K_2$, we will have a success iteration since $z_{k+1} < \bar{z} + \delta \leq w_\ell + \varepsilon$, whence the corresponding decrease in the target value will yield

$$w_{\ell+1} = z_{k+1} - \varepsilon - \eta\Delta < (\bar{z} + \delta) - \varepsilon - \eta\Delta < \bar{z} - \varepsilon + \delta$$

Moreover, once $k \geq K_2$ at the beginning of some outer loop, any further decrease in target value must also satisfy

$$w_{\ell+1} = z_{k+1} - \varepsilon - \eta\Delta > z_{k+1} - \varepsilon - \Delta > z_{k+1} - \varepsilon - \delta \geq \bar{z} - \varepsilon - \delta$$

since then $\Delta < \delta$. Therefore, we can find an inner iteration $\tilde{k} \geq K_2$ during an outer iteration $\tilde{\ell}$ such that

$$\bar{z} - \varepsilon - \delta < w_{\tilde{\ell}} < \bar{z} - \varepsilon + \delta \tag{2.19}$$

Now, it remains to show that (2.19) holds for all $\ell \geq \tilde{\ell}$. Let $j(1)$ be the first iteration after $\tilde{k}$ at which the target value needs to be changed at the next outer loop, so that either

$$w_{\tilde{\ell}+1} = (z_{\tilde{k}+j(1)} - \varepsilon) - \eta\Delta \tag{2.20a}$$

in the case of a success, or

$$w_{\tilde{\ell}+1} = \frac{(z_{\tilde{k}+j(1)} - \varepsilon) + w_{\tilde{\ell}}}{2} \tag{2.20b}$$

in the case of a failure. If (2.20a) occurs, we have

$$\bar{z} - \varepsilon + \delta > w_{\tilde{\ell}} > w_{\tilde{\ell}+1} = (z_{\tilde{k}+j(1)} - \varepsilon) - \eta\Delta \geq \bar{z} - \varepsilon - \delta\,,$$

and if (2.20b) occurs, we have

$$\bar{z} - \varepsilon - \delta < w_{\tilde{\ell}} < w_{\tilde{\ell}+1} = \frac{(z_{\tilde{k}+j(1)} - \varepsilon) + w_{\tilde{\ell}}}{2} < \frac{(\bar{z} + \delta) - \varepsilon + (\bar{z} - \varepsilon + \delta)}{2} = \bar{z} - \varepsilon + \delta$$

Hence, in either case, $w_{\tilde{\ell}+1}$ continues to satisfy (2.19). Applying the foregoing argument inductively, we get that (2.19) holds for all $\ell \geq \tilde{\ell}$ and this completes the proof.

■

Theorem 2.3 Suppose that we run Algorithm VTVM in conjunction with a conjugate subgradient strategy of the form (2.2), and with step-length rule (2.1c), where $0 < \varepsilon_1 \leq \beta_k \leq 1$ for all k. If $k_{\max} = \bar{\tau} = \infty$, $\varepsilon_\ell = \varepsilon$ for all ℓ, and $\|\mathbf{d}_k\| < M$ for all k, and for some $M > 0$, then we have,

$$\bar{z} - \varepsilon \leq f^*$$

Proof Assume on the contrary that $\bar{z} - \varepsilon > f^*$. Then, we can choose a δ satisfying $\varepsilon > \delta > 0$ such that

$$\bar{z} - \varepsilon - f^* \geq \delta > 0$$

By Lemma 2.3, we can find an outer iteration $\tilde{\ell}$ such that

$$\underline{w} \equiv \bar{z} - \varepsilon - \delta < w_\ell < \bar{z} - \varepsilon + \delta \equiv \overline{w} \quad \text{for all} \quad \ell \geq \tilde{\ell}$$

By Lemma 2.2, since $f^* \leq \underline{w} < w_\ell < \overline{w} < \bar{z} \leq f_k$ for all k and ℓ sufficiently large, we get $\{f_k\} \to \overline{w}$. But this is a contradiction because $\overline{w} < \bar{z}$ since $\varepsilon > \delta$, and so the proof is complete. ■

Therefore, if Algorithm VTVM is implemented under the assumptions of Theorem 2.3 in conjunction with any (deflected) subgradient method that satisfies the assumptions of Theorem 2.1, it will generate a sequence of incumbent objective function values that converge to an ε-neighborhood of the optimal objective function value. It is noteworthy that this algorithm theoretically permits the use of the promising step-length rule (2.1c), instead of using (2.1a) or (2.1b) which are known to be poor step-length rules in practice.

The following result establishes the convergence property of Algorithm VTVM using variable acceptance tolerances ε_ℓ, as prescribed by the stated procedure. This is important from a computational efficiency viewpoint, since a small, fixed, acceptance tolerance of $\varepsilon_\ell = \varepsilon$, can lead to a slow progress during the initial stages of the algorithm. In fact, we found that the adaptive scheme in which ε_ℓ is permitted to vary as in the algorithmic statement of VTVM, yielded a superior and more stable computa-

tional performance than using a fixed value of ε, while the latter choice tended to generate undesirable oscillations in the target value.

Theorem 2.4 Suppose that Algorithm VTVM is operated in conjunction with a conjugate subgradient strategy of the form (2.2), and with step-length rule (2.1c), where $0 < \varepsilon_1 \leq \beta_k \leq 1$ for all k. If $k_{\max} = \bar{\tau} = \infty$, $\sigma^{(1)} < 1/3$, and if $\|\mathbf{d}_k\| < M$ for all k, for some $M > 0$, then we have $\bar{z} - \varepsilon \leq f^*$.

Proof As in the proof of Theorem 2.3, let us assume on the contrary that $\bar{z} - \varepsilon - f^* \geq \delta > 0$ for some δ satisfying $0 < \delta < \varepsilon$. We will show that there exists then, an outer iteration $\tilde{\ell}$ such that

$$\bar{z} - \varepsilon - \delta < w_\ell < \bar{z} - \varepsilon + \delta \quad \text{for all} \quad \ell \geq \tilde{\ell} \tag{2.21}$$

which would hence yield a contradiction by Lemma 2.2 as in the proof of Theorem 2.3.

Toward this end, similar to the proof of Lemma 2.3, let us examine the algorithmic process once we have $k \geq K_2$ and $\ell \geq L$ for a sufficiently large L so that $z_k < \bar{z} + \delta$ for all $k \geq K_2$ and $\sigma_\ell \leq 1/3$ for all $\ell \geq L$. (Note that L exists since the outer loop counter ℓ is incremented at least every $\bar{\gamma}$ iterations.) If $w_\ell \leq \bar{z} - \varepsilon - \delta$ for some outer loop ℓ, then let us consider two cases, namely, either a success or a failure during this outer loop. In the case of a success, we must have $\varepsilon_\ell \geq \varepsilon$, and then $\varepsilon_{\ell+1}$ is either set at ε, or else, for the present iteration index k, we set

$$\varepsilon_{\ell+1} = (z_{k+1} - w_{\ell+1})\sigma_\ell = \sigma_\ell(\varepsilon_\ell + \eta\Delta) \leq \sigma_\ell(\varepsilon_\ell + \delta) < \sigma_\ell(\varepsilon_\ell + \varepsilon) \leq \frac{2}{3}\varepsilon_\ell \tag{2.22}$$

In the case of a failure, for some iteration index k, we have at Step 3b that $z_{k+1} - \varepsilon_\ell > w_\ell$, where, assuming that $\varepsilon_\ell > \varepsilon$, we have for some $k' < k$,

$$\varepsilon_\ell = (z_{k'+1} - w_\ell)\sigma_{\ell-1} \geq \sigma_\ell(z_{k+1} - w_\ell) \tag{2.23}$$

Moreover, at this iteration k, we set $w_{\ell+1} = [(z_{k+1} - \varepsilon_\ell) + w_\ell]/2$ and update $\varepsilon_{\ell+1} = \max\{(z_{k+1} - w_{\ell+1})\sigma_\ell, \varepsilon\}$. Hence, either $\varepsilon_{\ell+1} = \varepsilon$, or else

$$\varepsilon_{\ell+1} = (z_{k+1} - w_{\ell+1})\sigma_\ell = \left[z_{k+1} - \frac{(z_{k+1} - \varepsilon_\ell) + w_\ell}{2}\right]\sigma_\ell = \frac{\varepsilon_\ell \sigma_\ell}{2} + \frac{(z_{k+1} - w_\ell)\sigma_\ell}{2}$$

Thus, by (2.23), we have

$$\varepsilon_{\ell+1} \leq \frac{\varepsilon_\ell}{6} + \frac{\varepsilon_\ell}{2} = \frac{2}{3}\varepsilon_\ell$$

Hence, if the target value w_ℓ remains less than or equal to $\bar{z} - \varepsilon - \delta$, ε_ℓ decreases at a geometric rate and will ultimately become ε, and from the foregoing arguments, will remain at ε while the procedure goes through a series of failure, ultimately making $w_\ell > \bar{z} - \varepsilon - \delta$.

After this occurs, in case we ever obtain $w_{\ell+1} > \bar{z} - \varepsilon + \delta$ at some outer loop update performed at some iteration $k + 1$, then

$$(z_{k+1} - w_{\ell+1})\sigma_\ell < [(\bar{z} + \delta) - (\bar{z} - \varepsilon + \delta)]\sigma_\ell = \varepsilon\sigma_\ell < \varepsilon$$

and so, $\varepsilon_{\ell+1} \equiv \varepsilon$. But this means as in the proof of Lemma 2.3 that for some outer iteration $\tilde{\ell} \geq L$, we will have

$$\bar{z} - \varepsilon - \delta < w_{\tilde{\ell}} < \bar{z} - \varepsilon + \delta \tag{2.24}$$

Now, let us show that if w_ℓ satisfies (2.21), then $\varepsilon_\ell \equiv \varepsilon$ for all $\ell \geq \tilde{\ell}$. This follows because once any revised $w_{\ell+1}$ falls in the range defined by (2.24) or (2.21) at some iteration $k+1$, we then have

$$(z_{k+1} - w_{\ell+1})\sigma_\ell < [(\bar{z} + \delta) - (\bar{z} - \varepsilon - \delta)]\sigma_\ell = (\varepsilon + 2\delta)\sigma_\ell < \frac{\varepsilon + 2\varepsilon}{3} = \varepsilon$$

Hence, by the prescribed update formula, we will have $\varepsilon_{\ell+1} = \varepsilon$. Therefore, by the proof of Lemma 2.3, (2.21) continues to hold for all $\ell \geq \tilde{\ell}$, and this completes the proof. ■

2.5 Implementation and Computational Experience

For computational testing purposes, we attempted to solve two groups of test problems. The first group includes some well-known standard (convex) test problems such as the dual transportation problem *(TR48)*, the dual assignment problem *(A48)*, and the problem of finding the minimum of a maximum of several quadratic functions *(MXQD)* (see Lemarechal, 1982, Lemarechal and Mifflin, 1977, and Kiwiel, 1985). Table 2.1 gives the sizes of the standard test problems in the first group and their optimal objective values.

Table 2.1. Minimization test problems and their optimal objective values.

	TR48	A48	MXQD [1]
n	48	48	10
f^*	-638565	-9870	-0.8414

(1): f is the maximum of five convex quadratic functions.

The second group of test problems involves the minimization of some strongly convex functions that have been used by Kim et al. (1991). The strongly convex functions, denoted by SCi, $i = 1,..., 10$, have the following form:

$$f(\mathbf{x}) = \max_{i \in I}\{\mathbf{a}_i^t\mathbf{x} + b_i\} + \sum_{j=1}^{n}(x_j - c_j)^2$$

Here, the vectors $\{\mathbf{a}_i : i \in I\}$ and the scalars $\{b_i : i \in I\}$ are randomly generated on $[-1, 1]$, and the quantities $\{c_j : j = 1,..., n\}$ are randomly generated on $[-2, 2]$. We ran Kim et al.'s algorithm using their recommended parameters (see Kim et al., 1991). Table 2.2 presents the sizes of the problems and the results obtained after 10,000 iterations of Kim et al.'s algorithm. Here, $f(\mathbf{x}_{best})$ denotes the best objective function value (upper bound) obtained, LB denotes the obtained lower bound via their algorithm, and OPT(%) indicates the percentage optimality computed via the formula

$$\text{OPT}(\%) = \left[1 - \frac{f(\mathbf{x}_{best}) - \text{LB}}{\text{LB}}\right] \times 100$$

Despite the gap evident in the OPT(%) column of Table 2.2, let us use the lower bounds LB as our estimates of the optimal value to evaluate the performance of the other algorithms tested later on, in comparison with Kim et al.'s procedure.

All the algorithms tested were coded in FORTRAN VS II, and run on an IBM 3090, Model 300E computer. Computational experiments were conducted for studying the effect of the two stopping criteria on Algorithm VTVM. First, among the two stopping criteria, namely k_{max} and $\bar{\tau}$, we set the maximum number of iterations k_{max} at various

finite values, while we put $\bar{\tau} = \infty$. Similarly, for the second set of runs, we fixed the maximum number of consecutive target increases $\bar{\tau}$ at various finite values and kept $k_{max} = \infty$. In these tests, we also ran Algorithm VTVM along with several direction finding strategies such as pure-subgradient (PURE), MGT, ADS, and ODSA. Restarting strategies were used in conjunction with all tested procedures (see Remark 2.2).

Table 2.2. Strongly Convex Test Problems

Problem	(||, n)	$f(x_{best})$	LB	OPT(%)
SC1	(50, 30)	8.7525	8.5822	98.02
SC2	(70, 50)	8.8229	8.4637	95.76
SC3	(100, 50)	9.9488	9.7779	98.25
SC4	(100, 100)	15.5873	14.8571	97.51
SC5	(150, 50)	8.9133	8.7091	97.66
SC6	(150, 100)	12.5979	12.1270	96.12
SC7	(200, 50)	9.0951	8.8133	96.80
SC8	(200, 100)	14.7125	14.1078	95.71
SC9	(200, 150)	17.4146	16.7968	96.32
SC10	(200, 200)	23.4783	22.5860	96.02

In Tables 2.3-2.6, we present the best solutions obtained after 1000 and 2000 iterations for the first group of test problems given in Table 2.1, and after 200 iterations for the second group of test problems given in Table 2.2. The parameters $(\gamma^{(1)}, \gamma^{(2)}, \sigma^{(1)}, \sigma^{(2)})$ used in these tests are (50, 10, 0.1, 0.5) and (15, 10, 0.15, 0) for the first and the second group of test problems, respectively. Since the first group of problems are relatively harder than the second group of problems, we used a different

level of k_{max}, and accordingly, different $\gamma^{(1)}$ and $\sigma^{(1)}$ values above. However, we fixed $(\beta^{(1)}, \beta^{(2)}) = (0.25, 0.75)$ throughout the tests. In the tables, ITR(CPU) denotes the iteration number at which the best recorded solution was found within the corresponding iteration limit, which coincides with the number of function evaluations, with the cpu time in seconds given in parenthesis.

Table 2.3 presents computational results for VTVM when operated in conjunction with various deflection strategies for the first group of problems. We observe that Algorithm VTVM performs well for all the strategies by attaining at least 99.96%, 99.99%, and 95.70% of optimality for TR48, A48, and MXQD, respectively. Moreover, we see that ADS and ODSA perform better than PURE and MGT. In particular, for Problem MXQD, ADS and ODSA achieved better than 98% of optimality. Also, note that all the subgradient deflection strategies enjoy a superiority over the pure-subgradient method by accelerating tail-end convergence. (See the difference between 1000 and 2000 runs for Problem MXQD).

Table 2.3. Computational results for VTVM when $k_{max} = 1000$ and 2000.

PROB	PURE		MGT		ADS		ODSA	
	$f(x_{best})$	ITR(CPU)	$f(x_{best})$	ITR(CPU)	$f(x_{best})$	ITR(CPU)	$f(x_{best})$	ITR(CPU)
TR48	-638295.34 -638448.37	989(2.50) 1947(5.05)	-638411.87 -638419.87	972(2.50) 1968(5.09)	-638445.30 -638483.89	961(2.43) 1993(5.11)	-638462.20 -638470.23	990(2.69) 1984(5.26)
A48(*)	-9869.28	377(1.09)	-9869.07	442(1.27)	-9869.18	432(1.24)	-9869.29	401(1.22)
MXQD	7.0076 -0.8052	689(0.55) 1953(1.58)	67.54 -0.8223	937(0.75) 1975(1.61)	61.92 -0.8309	932(0.74) 1975(1.65)	49.29 -0.8317	965(0.80) 1967(1.63)

- the optimal values are -638565, -9870, and -0.8414, respectively, for TR48, A48, and MXQD.
- parameters are $(k_{max}, \bar{\tau}) = (1000 \text{ and } 2000, \infty)$ and $(\gamma^{(1)}, \gamma^{(2)}, \sigma^{(1)}, \sigma^{(2)}, \beta^{(1)}, \beta^{(2)}) = (50, 10, 0.1, 0.5, 0.25, 0.75)$.
- (*) Each algorithm achieved 99.99% of optimality in less than 500 iterations.

For the next set of comparisons, we have chosen strongly convex test problems as used by Kim et al. (1991), since these problems admit a lower bound computation on the optimal objective value that can be used for algorithms that require such a bound as either an initial or a fixed target value. In this comparison, we observed the computational performance of the proposed algorithm VTVM, along with that of the blocking-halving (BH) method of Sherali and Ulular (1989), the variable target value method (KIM) of Kim et al. (1991), and with the results obtained using various combinations of these rules in conjunction with the three subgradient deflection strategies MGT, ADS, and ODSA as tested previously. Throughout the test runs, we set 200 as the maximum number of iterations, and used the initial LB (see Kim et al., 1991) as the fixed lower bound for BH and as the initial target value for VTVM. (For BH, we used 3 as the number of blocks, and we set the maximum number of iterations per block to 75, and the maximum number of consecutive failures before restarting to 15 - see Sherali and Ulular, 1991 for details.) For Algorithm VTVM, we fixed $(\gamma^{(1)}, \gamma^{(2)}\ \sigma^{(1)}, \sigma^{(2)}) = (15, 10, 0.15, 0)$.

Table 2.4 presents results for the different step-length rules used by the foregoing methods in conjunction with the pure-subgradient algorithm (no deflections). We observe that Algorithm VTVM is quite competitive for these particular test problems. Actually, we found that the given initial LB's did not affect the performance of VTVM, while they are necessarily required for implementing KIM and BH. Moreover, BH and VTVM perform significantly better than Kim et al.'s algorithm, and so, we will present further results using the two former methods.

Table 2.5 gives test results for the three deflected subgradient strategies used in conjunction with BH and VTVM, respectively. From these runs, we observe that all

three deflection strategies perform almost equally, attaining at an average (for the BH and VTVM runs taken together) 95.12%, 95.07%, and 95.04% of optimality for MGT, ADS, and ODSA, respectively. Quite surprisingly, the pure-subgradient method gives fairly good results (95.39% optimality achieved at an average over the BH and VTVM runs). Evidently, the relatively well behaved test functions do not disfavor the pure-subgradient in this case. If we view the results in terms of comparing VTVM and BH, we observe that VTVM performs very competitively (actually, achieving solutions that are 0.16% closer to optimality than does BH). In summary, throughout the above experiments, we found that VTVM performs well in comparison, even though known lower bounds were assumed to be available for the competing methods.

Table 2.6 summarizes the computational results presented in Tables 2.4-2.5.

Table 2.4. Computational results using pure-subgradient directions with various step-length rules and with $k_{max} = 200$.

	KIM			BH			VTVM		
	$f(x_{best})$	ITR(CPU)	OPT(%)	$f(x_{best})$	ITR(CPU)	OPT(%)	$f(x_{best})$	ITR(CPU)	OPT(%)
SC1	9.4100	186(0.32)	90.35	8.7232	182(0.30)	98.36	8.7368	86(0.14)	98.20
SC2	9.6906	189(0.73)	85.50	8.8401	198(0.74)	95.55	8.9321	200(0.74)	94.47
SC3	10.3628	132(1.05)	94.02	9.9243	162(0.84)	98.50	10.0375	199(1.03)	97.34
SC4	16.6451	127(2.07)	87.97	15.6380	200(2.08)	94.74	15.8223	189(1.98)	93.50
SC5	9.7732	162(1.54)	87.78	9.1007	199(1.55)	95.50	9.0702	199(1.57)	95.85
SC6	13.5342	74(3.10)	88.40	12.7525	178(2.73)	94.84	12.6907	199(3.07)	95.35
SC7	9.3655	200(2.01)	93.73	9.1397	191(1.93)	96.29	9.5028	200(2.03)	92.18
SC8	16.0777	170(4.04)	86.04	14.8428	173(3.50)	94.79	14.6386	155(2.99)	96.24
SC9	21.6099	75(6.04)	71.35	18.0173	181(5.59)	92.73	17.9199	199(6.11)	93.31
SC10	25.6703	160(8.29)	86.34	24.0524	200(8.21)	93.51	23.3631	118(7.97)	96.56

- parameters for VTVM are $(k_{max}, \bar{\tau}) = (200, \infty)$ and $(\gamma^{(1)}, \gamma^{(2)}, \sigma^{(1)}, \sigma^{(2)}, \beta^{(1)}, \beta^{(2)}) = (15, 10, 0.15, 0, 0.25, 0.75)$.

Table 2.5. Computational results for BH and VTVM with various deflection strategies.

	MGT+BH			ADS+BH			ODSA+BH		
	$f(x_{best})$	ITR(CPU)	OPT(%)	$f(x_{best})$	ITR(CPU)	OPT(%)	$f(x_{best})$	ITR(CPU)	OPT(%)
SC1	8.7208	188(0.31)	98.39	8.7149	200(0.33)	98.45	8.7511	197(0.34)	98.03
SC2	8.8690	197(0.75)	95.21	8.9909	178(0.67)	93.77	9.1205	181(0.70)	92.24
SC3	9.9824	155(0.82)	97.91	9.9836	192(1.02)	97.90	10.1039	192(1.03)	96.67
SC4	15.6152	184(1.92)	94.90	15.5516	179(1.87)	95.33	15.5443	199(2.15)	95.37
SC5	9.0145	200(1.56)	96.49	9.0244	195(1.52)	98.38	9.0858	198(1.57)	95.67
SC6	12.9105	195(3.04)	93.54	13.0021	174(2.70)	92.78	12.7174	180(2.85)	95.13
SC7	9.3151	190(1.93)	94.31	9.2593	191(1.94)	94.94	9.4016	173(1.79)	93.32
SC8	14.8234	152(3.08)	94.93	14.8961	200(4.04)	94.41	14.9207	177(3.65)	94.24
SC9	18.1793	198(5.84)	91.77	18.4778	195(6.00)	89.99	18.07921	193(6.07)	92.37
SC10	23.8020	150(6.20)	94.62	24.2753	145(5.97)	92.52	23.8653	173(6.82)	94.34
	MGT+VTVM			ADS+VTVM			ODSA+VTVM		
SC1	8.7446	188(0.30)	98.11	8.7993	140(0.23)	97.47	8.7217	192(0.35)	98.37
SC2	8.9329	186(0.68)	94.46	8.9244	198(0.74)	95.84	8.9165	200(0.77)	94.65
SC3	10.0465	175(0.90)	97.25	10.0814	186(0.92)	96.90	10.0707	182(0.97)	97.01
SC4	15.8748	190(1.99)	93.15	15.6006	196(2.13)	95.00	15.5650	199(2.20)	95.24
SC5	9.0574	200(1.53)	96.00	9.1358	188(1.44)	95.10	9.1576	198(1.57)	94.85
SC6	12.7846	200(3.07)	94.56	13.3257	192(2.88)	90.12	13.1772	184(2.80)	91.34
SC7	9.7575	200(2.02)	89.29	9.1698	173(1.85)	95.95	9.2483	197(2.04)	95.06
SC8	14.7808	195(3.95)	95.23	14.9281	195(3.96)	94.19	14.7501	163(4.86)	95.45
SC9	17.4853	185(5.73)	95.90	17.5060	200(6.89)	95.78	17.5414	187(6.82)	95.57
SC10	23.4217	177(8.45)	96.30	23.4403	199(9.05)	96.22	23.5177	152(8.76)	95.87

- parameters for BH are $k_{max} = 200$, number of blocks $= 3$, number of iterations per block $= 75$, failure reset frequency $= 15$
- parameters for VTVM are $(k_{max}, \bar{\tau}) = (200, \infty)$ and $(\gamma^{(1)}, \gamma^{(2)}, \sigma^{(1)}, \sigma^{(2)}, \beta^{(1)}, \beta^{(2)}) = (15, 10, 0.15, 0, 0.25, 0.75)$.

Table 2.6. Summary of computational results (average % optimality) for the strongly convex test problems.

	$(\sigma^{(1)}, \sigma^{(2)})$	PURE	MGT	ADS	ODSA	AVERAGE
BH	N/A	95.48	95.21	94.87	94.74	95.07
VTVM	(0.1, 0.5)	91.18	90.52	89.18	94.32	91.30
	(0.15, 0)	95.30	95.03	95.26	95.34	95.23
	(0.2, 0)	95.56	92.81	92.83	94.28	93.87
	(0.25, 0)	94.53	91.24	92.62	94.60	93.25

Thus far, we have used a finite value of k_{max} as a stopping rule. In general, it is more suitable to hold k_{max} relatively high, and use some consecutive failure measure as an alternative termination criterion. Therefore, we now present some computational results when a finite $\bar{\tau}$ is used as a stopping rule, but with $k_{max} = \infty$. Table 2.7 presents results obtained when $\bar{\tau} = 15$ and 30 are used for the first group of test problems, along with $\gamma^{(1)} = 30$ and 50, and $\bar{\tau} = 15$ is used along with $\gamma^{(1)} = 15$ for the second group of test problems. For these computations, we used $(\gamma^{(2)}, \sigma^{(1)}, \sigma^{(2)}) = (10, 0.1, 0.5)$ throughout. (Due to the relative difficulty of the first group of test problems, a higher value of $\bar{\tau} = 30$ was needed to attain near optimality, while $\bar{\tau} = 15$ was good enough for the second group of problems.) From these experiments, we observe that VTVM performs uniformly well throughout both the groups of test problems. Moreover, we find that strategy ADS + VTVM performs slightly better than the other combinations, although all combinations perform reasonably well. Also, while the parameter choice for $(\sigma^{(1)}, \sigma^{(2)}) = (0.1, 0.5)$ gave the worst results, relatively speaking, when we set $k_{max} = 200$ and $\bar{\tau} = \infty$, this choice of parameters yields the best objective function values when we use $k_{max} = \infty$ and $\bar{\tau} = 15$. On the other hand, the parameter value choices of $(\sigma^{(1)}, \sigma^{(2)}) = (0.15, 0)$, $(0.2, 0)$, and $(0.25, 0)$ attain roughly the same % of

optimality as that achieved in a 200 iteration limit. Based on these experiments, we suggest the parameter choices as prescribed following the statement of Algorithm VTVM in Section 4.

Table 2.7. Computational results for VTVM when $k_{max} = \infty$, $\bar{\tau} < \infty$

PROB	$\gamma^{(1)}$	$\bar{\tau}$	PURE		MGT		ADS		ODSA	
			$f(x_{best})$	ITR(CPU)	$f(x_{best})$	ITR(CPU)	$f(x_{best})$	ITR(CPU)	$f(x_{best})$	ITR(CPU)
TR48	30	15	-635533.76	2224(5.68)	-637286.85	2824(7.30)	-638376.19	1286(3.24)	-638243.22	1253(3.33)
	30	30	-635533.82	2796(7.06)	-637286.85	2824(7.32)	-638376.19	1486(3.84)	-638243.26	1543(4.12)
	50	15	-637025.76	3087(7.91)	-638387.90	1858(4.79)	-638387.74	1667(4.26)	-638245.86	1708(4.54)
	50	30	-637025.12	4556(11.69)	-638387.93	2292(5.59)	-638397.76	1874(4.83)	-638245.91	2149(5.68)
A48	30	15	-9869.99	4805(12.71)	-9869.99	3838(10.23)	-9869.98	7889(21.03)	-9869.96	3832(10.32)
	30	30	-9869.99	5293(13.90)	-9869.99	2828(10.24)	-9869.98	7889(21.03)	-9869.9 6	3832(10.37)
*	50	15	-9869.99	4805(13.05)	-9869.99	4173(11.10)	-9869.99	8288(21.98)	-9869.99	3832(10.29)
MXQD	30	15	64.8185	415(0.34)	151.4769	515(0.42)	214.1453	501(0.14)	301.3940	506(0.42)
	30	30	-0.8321	1487(1.21)	-0.8381	1423(1.16)	-0.8365	1199(0.97)	-0.8336	2175(1.81)
	50	15	7.0076	689(0.56)	67.5431	937(0.77)	214.1453	741(0.62)	90.3253	910(0.77)
	50	30	-0.8351	2425(1.99)	-0.8369	2013(1.66)	-0.8309	1975(1.65)	-0.8317	1967(1.)6)
SC1	15	15	8.7200	428(0.70)	8.7179	210(0.35)	8.7457	344(0.95)	8.7490	282(0.63)
SC2	15	15	8.7594	485(1.83)	8.8209	344(1.28)	8.8399	326(1.26)	8.7976	319(1.27)
SC3	15	15	9.9215	612(3.21)	9.9879	380(2.01)	9.9629	348(1.96)	9.9741	222(1.12)
SC4	15	15	15.4419	503(5.31)	15.4950	396(4.32)	15.5594	298(3.44)	15.5000	185(2.77)
SC5	15	15	8.8931	450(3.49)	8.9613	362(2.87)	8.9981	324(2.69)	8.9330	341(2.77)
SC6	15	15	12.5316	388(6.03)	12.5980	383(6.21)	12.6677	393(6.23)	12.6111	316(5.71)
SC7	15	15	9.0503	593(6.05)	9.1132	307(3.21)	9.1200	336(3.95)	9.0945	388(4.14)
SC8	15	15	14.5859	383(7.84)	14.6257	405(8.51)	14.6870	250(6.58)	14.6858	210(5.55)
SC9	15	15	17.3209	468(14.50)	17.5000	362(11.55)	17.4209	340(11.16)	18.2992	185(6.00)
SC10	15	15	23.3242	(18.81)	23.4801	260(11.17)	23.6151	299(10.68)	23.8238	218(10.95)

- parameter values: $(\gamma^{(2)}, \sigma^{(1)}, \sigma^{(2)}, \beta^{(1)}, \beta^{(2)}) = (10, 0.15, 0, 0.25, 0.75)$.
- (*): All algorithms attain 99.99% of optimality when $\gamma^{(1)} = 50$ and $\bar{\tau} = 15$.

In conclusion, we remark that the new variable target value method VTVM not only enjoys global convergence property whether any information about the optimal objective value is available or not, but it also appears to perform well computationally under reasonable stopping criteria without the need for much fine tuning of parameter values. Also, we note that the new deflection strategy ODSA is quite competitive with other popular deflection strategies.

3. Memoryless Space Dilation and Reduction Algorithm

3.1 Introduction

Consider the nondifferentiable optimization (NDO) problem of minimizing a convex, though not necessarily differentiable, function f over E_n. (The case of $\mathbf{x}$ being restricted to lie in some convex subset X of E_n is treated subsequently.) This problem has received a great deal of attention over the past two decades, and thus, several new methods have been proposed. Since nondifferentiability causes serious difficulties when the classical methods designed for differentiable problems are used, most algorithms for NDO use subgradients in some modified, related fashion for finding a direction of motion, and usually prescribe in closed-form a suitable step size to take along the direction.

Depending on the particular strategy used for finding the direction of motion, algorithms for NDO can be categorized as follows. The *pure-subgradient algorithm*, which is an analogue of the steepest descent method for differentiable problems, uses an anti-subgradient as the direction of motion. On the other hand, the *deflected subgradient algorithms*, also called *conjugate subgradient algorithms*, imitate conjugate gradient methods. As with the steepest descent direction for the differentiable case, the anti-subgradient direction for the nondifferentiable case can result in a zigzagging phenomenon that might manifest itself at any stage of the subgradient algorithm, causing the procedure to crawl toward optimality. As a tool to overcome this difficulty, a conjugate subgradient concept has been introduced in which the direction of motion is computed by combining the current anti-subgradient with the previous direction. Note that while some conjugacy requirement (see Fletcher and Reeves, 1959, and Hestenes and Stiefel, 1952) is enforced for the differentiable case, for the nondifferentiable case, this concept is used only as a strategy to deflect the anti-subgradient. The *Modified Gradient Technique (MGT)* of Camerini et al. (1975) and the *Average Direction Strategy (ADS)* of Sherali and Ulular (1989) are two well-known and extensively tested deflection strategies.

As an alternate approach for NDO, *bundle type methods* have been proposed by Mifflin (1977) and Lemarechal (1978) in which the direction of motion is obtained via the convex hull of a set of previously generated subgradients, known as (subgradient) bundles. Unlike the choice of the step-length in subgradient based algorithms, these bundle methods involve an inexact line search that produces either an improved solution (serious step) or a trial solution that is rejected (null step). In either case, a new subgradient is computed and added to the existing bundle to find a modified direction of motion. Lemarechal (1980) presents the concepts of this approach.

Kiwiel (1985, 1989) and Lemarechal (1986) introduce several different strategies to construct such bundles. Also, Kiwiel (1990, 1991) presents improved strategies for such methods under the title of *proximal bundle methods*. A major difference between bundle and ordinary subgradient methods is that, unlike the latter, the bundle type methods generate a sequence of iterates for which the objective function values are monotone decreasing. For this reason, the bundle type methods are classified as "descent methods." However, one difficulty of the bundle methods is that they require the solution of a quadratic subproblem at each iteration for finding the direction of motion, and this can be quite expensive. Indeed, it proves to be impractical for larger sized problems.

In contrast with the subgradient based methods and with the above bundle type methods, the *variable metric methods* have different algorithmic and theoretical characteristics. For the differentiable case, we can obtain a deflected gradient by using a transformed metric based on the Hessian matrix (or its approximation). Similarly, for NDO, a subgradient can be deflected by premultiplying it with a suitable matrix. Shor (1970a, 1970b) presents a *space dilation (dilatation) procedure* that employs an analogue space transformation along the gradient direction. Goffin (1977) presents convergence results for several different choices of transformation parameters. Of noteworthy value due to its computational performance, Shor and Zhurbenko (1971) and Shor and Shabashova (1972) propose a space dilation algorithm, dilating the space along the difference of two successive subgradients, called the *r-algorithm*. Assuming that the selected anti-subgradient is almost perpendicular to the direction toward optimality, this r-algorithm is designed to reduce the orthogonal component of the subgradient with respect to the optimal direction, with the intent of alleviating the zig-zagging phenomenon. In practice, the computational

performance appears good, but the method is expensive because of the matrix storage and updating requirements. Moreover, its theory is complicated, and it loses the simplicity of subgradient based algorithms that have made the latter so popular. However, by adopting the idea of memoryless updates in quasi-Newton methods (1979), we can hope to overcome some of its disadvantages. This can be done by computing the space transformation operator at each iteration by updating the identity matrix instead of the previous approximation. Naturally, no matrix storage is required. Moreover, in this case, with a proper choice of parameters that admits convergence, it will be shown that the memoryless update turns out to be a convex combination of two successive subgradients, leading to a combination of space dilation and space reduction operations.

The remainder of this chapter is organized as follows. In Section 2, the proposed memoryless update scheme is presented. Section 3 presents strategies for selecting values for the space transformation parameter based on relationships with deflected subgradient methods, and establishes related convergence properties. As a result, we are able to imbed this update scheme within a variable target value method that guarantees convergence without any *a priori* information on the optimal objective function value. Finally, in Section 4, we present some computational results and comparisons, using standard dual transportation and assignment problems.

3.2 Memoryless Space Dilation and Reduction Strategy

Shor (1970a, 1970b) introduces the space dilation operator which can be represented as

$$\mathbf{R}_\alpha(\mathbf{r}) = \mathbf{I} + (\alpha - 1)\mathbf{r}\mathbf{r}^t$$

for some parameter $0 < \alpha < 1$ and some dilation vector $\mathbf{r}$. Note that under the transformation $\mathbf{x} = \mathbf{R}_\alpha(\mathbf{r})\mathbf{y}$, so that

$$\mathbf{y} = \mathbf{R}_\alpha(\mathbf{r})^{-1}\mathbf{x} = \left[\mathbf{I} + (\frac{1}{\alpha} - 1)\mathbf{r}\mathbf{r}^t\right]\mathbf{x},$$

the vector $\mathbf{y}$ is obtained by stretching the vector $\mathbf{x}$ along the direction $\mathbf{r}$, depending on the value of α. The original algorithm uses the (sub)gradient as a dilation vector. Subsequently, in Shor and Zhurbenko (1971) and Shor and Shabashova (1972), a dilation scheme along the direction of the difference between two consecutive subgradients has been proposed, and this is widely recognized as being among the most efficient methods for nondifferentiable optimization problems. In this procedure, at each iteration k, given a subgradient $\mathbf{g}_k$, the previous subgradient $\mathbf{g}_{k-1}$, and the previous transformation operator $\mathbf{B}_{k-1}$, a new transformation operator $\mathbf{B}_k$ is obtained via the following update scheme:

$$\mathbf{B}_k = \mathbf{B}_{k-1}\mathbf{R}_{\alpha_k}(\mathbf{r}_k') \quad \text{where} \quad \mathbf{r}_k' = \frac{\mathbf{B}_{k-1}^t(\mathbf{g}_k - \mathbf{g}_{k-1})}{\|\mathbf{B}_{k-1}^t(\mathbf{g}_k - \mathbf{g}_{k-1})\|}.$$

Using the space transformation $\mathbf{x} = \mathbf{B}_k\mathbf{y}$, so that $\mathbf{B}_k^t\mathbf{g}_k$ is a subgradient of $F(\mathbf{y}) \equiv f(\mathbf{B}_k\mathbf{y})$ under this transformation, a step-length of λ_k (presented in Shor, 1985) along the

anti-subgradient in the transformed space yields $\mathbf{y}_{k+1} = \mathbf{y}_k - \lambda_k \mathbf{B}_k^t \mathbf{g}_k$. In the original $\mathbf{x}$ space, this yields the new iterate

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \mathbf{B}_k \mathbf{B}_k^t \mathbf{g}_k$$

Later, Skokov (1974) showed that one can reduce the number of arithmetic operations in the computations by using the symmetric matrix $\mathbf{H}_k = \mathbf{B}_k \mathbf{B}_k^t$, and directly updating this equivalently via the following rank-one update scheme, where $\mathbf{q}_{k-1} \equiv \mathbf{g}_k - \mathbf{g}_{k-1}$.

$$\mathbf{H}_k = \mathbf{H}_{k-1} - (1 - \alpha_k^2) \frac{\mathbf{H}_{k-1} \mathbf{q}_{k-1} \mathbf{q}_{k-1}^t \mathbf{H}_{k-1}}{\mathbf{q}_{k-1}^t \mathbf{H}_{k-1} \mathbf{q}_{k-1}} \tag{3.1}$$

Accordingly, we would then have,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_k \mathbf{H}_k \mathbf{g}_k$$

This *r-algorithm*, as it is called, was shown to be quite promising for various practical problems (see Shor, 1985). However, for large sized problems, the matrix updating process becomes very expensive. By imitating the memoryless quasi-Newton algorithm for the differentiable case, we propose to alleviate some of the computational difficulties associated with the r-algorithm for larger sized problems. (See Bazaraa et al, 1993, and Luenburger, 1984) for a discussion on memoryless quasi-Newton algorithms.)

Toward this end, assuming that $\|\mathbf{g}_k - \mathbf{g}_{k-1}\| \neq 0$, let us define

$$\mathbf{r}_k = \frac{\mathbf{q}_{k-1}}{\|\mathbf{q}_{k-1}\|} \equiv \frac{\mathbf{g}_k - \mathbf{g}_{k-1}}{\|\mathbf{g}_k - \mathbf{g}_{k-1}\|} \tag{3.2}$$

to be the normalized difference vector of two successive subgradients. For each k, by replacing $\mathbf{H}_{k-1}$ (or the previous transformation matrix $\mathbf{B}_{k-1}$) with the identity matrix in (3.1), we have

$$\mathbf{H}_k = \mathbf{I} - (1 - \alpha_k^2)\frac{\mathbf{q}_{k-1}\mathbf{q}_{k-1}^t}{\mathbf{q}_{k-1}^t\mathbf{q}_{k-1}} = \mathbf{I} - (1 - \alpha_k^2)\mathbf{r}_k\mathbf{r}_k^t \tag{3.3}$$

Accordingly, the direction of motion is computed as

$$\mathbf{d}_k = \mathbf{H}_k(-\mathbf{g}_k) = -\mathbf{g}_k + (1 - \alpha_k^2)(\mathbf{g}_k^t\mathbf{r}_k)\mathbf{r}_k \tag{3.4a}$$

and the new iterate is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k\mathbf{d}_k \tag{3.4b}$$

for some step-length λ_k. To gain some insight into the resulting prescribed direction, let us define

$$s_k = (1 - \alpha_k^2)\frac{\mathbf{g}_k^t\mathbf{r}_k}{\|\mathbf{g}_k - \mathbf{g}_{k-1}\|} \tag{3.5}$$

Then, the direction of motion can be written as

$$\mathbf{d}_k = -\mathbf{g}_k + s_k(\mathbf{g}_k - \mathbf{g}_{k-1}) = -[(1 - s_k)\mathbf{g}_k + s_k\mathbf{g}_{k-1}] \tag{3.6}$$

which is an affine combination of the present and previous anti-subgradients. Note that in Shor's original algorithm, α_k is chosen to lie in (0, 1) so that it dilates the space as mentioned above. But in here, we relax this requirement, permitting $\alpha_k > 1$ if necessary, so that at each iteration, the transformation can either dilate or shrink the space along the direction **r**. The motivation is to maintain $0 \leq s_k < 1$ so that the direction (3.6) is similar to that obtained via a conjugate subgradient deflection strategy

in which the previous direction is substituted by the previous anti-subgradient. As shown in the following section, this admits related convergence properties. As an aside, note in this connection that if $\alpha_k = 1$, then no space transformation is performed and we obtain $s_k = 0$, so that we simply adopt the pure-subgradient direction at this iteration.

3.3 The Algorithm and its Convergence

Before we present the proposed algorithm and examine its convergence, let us consider the issue of selecting a step-length rule. Note that for generalized subgradient methods, we can write

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \sum_{j \in J_k} t_{kj}\mathbf{g}_j, \tag{3.7}$$

where $J_k \subseteq \{1,2,..., k\}$ and $t_{kj} \geq 0$ for all j and k; that is, the direction of motion is some nonnegative linear combination of the previously obtained anti-subgradients. The convergence of this procedure has been proven by Kim and Ahn (1991) using the ε-subgradient concept of Nurminski and Zhelikhovski (1977) and Demyanov and Vasilev (1985), when the step-length is given by the usual rule satisfying $\lambda_k \geq 0$, $\lim_{k \to \infty} \lambda_k = 0$, and $\sum_{k=1}^{\infty} \lambda_k = \infty$. Since the direction of motion $\sum_{j \in J_k} t_{kj}\mathbf{g}_j$ is an ε_k-subgradient of f at $\mathbf{x}_k$ for some ε_k for each k, Kim and Ahn require $\lim_{k \to \infty} \varepsilon_k = 0$ for their convergence theorem to hold. Using this same divergent series step-length rule, we can establish

the convergence of the procedure (3.4) under a suitable assumption. To see this, let us define

$$v_k = f_k - f_{k-1} - \mathbf{g}_{k-1}^t(\mathbf{x}_k - \mathbf{x}_{k-1}) \geq 0 \tag{3.8}$$

where $f_t \equiv f(\mathbf{x}_t)$ for all t, and consider the following lemma.

Lemma 3.1 Suppose that the direction of motion $\mathbf{d}_k$ is given by (3.4a) or (3.6). If $0 \leq s_k < 1$ in (3.5) for all k, then

$$f(\mathbf{x}) - f_k \geq -\mathbf{d}_k^t(\mathbf{x} - \mathbf{x}_k) - s_k v_k$$

for all $\mathbf{x} \in E_n$, that is, $-\mathbf{d}_k$ is a $(s_k v_k)$-subgradient of f at $\mathbf{x}_k$.

Proof Using the fact that $\mathbf{g}_k$ is a subgradient of f at $\mathbf{x}_k$, and that $\mathbf{g}_{k-1}$ is a v_k-subgradient of f at $\mathbf{x}_k$, we have for all $\mathbf{x} \in E_n$,

$$\begin{aligned}
&-\mathbf{d}_k^t(\mathbf{x} - \mathbf{x}_k) - s_k v_k \\
&= (1 - s_k)\mathbf{g}_k^t(\mathbf{x} - \mathbf{x}_k) + s_k \mathbf{g}_{k-1}^t(\mathbf{x} - \mathbf{x}_k) - s_k v_k \\
&\leq (1 - s_k)[f(\mathbf{x}) - f_k] + s_k[f(\mathbf{x}) - f_k + v_k] - s_k v_k \\
&= f(\mathbf{x}) - f_k
\end{aligned}$$

This completes the proof. ■

Thus, according to the results of Kim and Ahn (1991) if $\lim_{k \to \infty} s_k v_k = 0$, then the procedure (3.4), used along with a divergent series step-length rule, will generate a sequence of solutions that converges to an optimal solution. Note that the assumptions

$0 \leq s_k < 1$ and $\lim_{k \to \infty} s_k v_k = 0$ can be established by choosing $\{\alpha_k\} \to 1$ suitably, so that $0 \leq s_k < 1$ for all k and $\lim_{k \to \infty} s_k = 0$, provided that $\{v_k\}$ is bounded. However, this strategy loses the merit of using a variable metric method, since $\{\alpha_k\} \to 1$ eventually leads the algorithm to a pure-subgradient method. Likewise, for the generalized subgradient methods characterized by (3.7), since t_{kj} implicitly involves both the affine combination weights associated with the subgradients and the step-lengths, we cannot control the affine combination weights independently, and thus, we might lose the advantage of freely selecting a desired direction. Considering these weaknesses, that are further aggravated by the poor computational behavior of a divergent series step-length rule, we devise a new algorithm that employs (3.4) along with the more popular step-length rule

$$\lambda_k = \beta_k \frac{f_k - w}{\|\mathbf{d}_k\|^2} \tag{3.9}$$

where w is a target value and $\beta_k > 0$ is a suitable parameter. Note that among the various step-length rules for subgradient algorithms (see Allen et al, 1987, and Goffin, 1980), this rule is known to be quite promising in practice.

From now on, we consider a constrained problem of minimizing $f(\mathbf{x})$ over some convex subset X of E_n, so that the proposed algorithm can be used for solving a wider class of problems. Hence, a new iterate is computed by

$$\mathbf{x}_{k+1} = P_X(\mathbf{x}_k + \lambda_k \mathbf{d}_k) \tag{3.10}$$

instead of (3.4b), where $P_X(\cdot)$ denotes a projection operation onto the set X. The following result establishes the principal convergence property of the proposed algorithm.

Theorem 3.1 Suppose that $w \geq f^* \equiv f(\mathbf{x}^*)$, where $\mathbf{x}^*$ is an optimal solution to the problem of minimizing $f(\mathbf{x})$ over $\mathbf{x} \in X$, and that the procedure (3.4a) and (3.10) using the step-length rule (3.9) is run while f_k remains greater than the target value w. Moreover, suppose that α_k and β_k used in (3.5) and (3.9), respectively, are chosen such that

$$0 \leq s_k < 1 \quad \text{and} \quad 0 < \varepsilon_1 \leq \beta_k \leq \varepsilon_{2k} \equiv 1 - \frac{s_k v_k}{f_k - w} \leq 1 \tag{3.11}$$

holds for all k for some $\varepsilon_1 > 0$, where v_k is given by (3.8). Assume that $\|\mathbf{g}_k\| < M$ for all k, for some $M > 0$. Then, either (i) for some k we will have $f_{k+1} \leq w$, or else, we will have (ii) $f_k > w$ for all k and $\lim_{k \to \infty} f_k = w$.

Proof Clearly, either $f_{k+1} \leq w$ for some k, or else $f_k > w$ for all k. Assuming the latter, we have

$$\begin{aligned}
\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 &= \|P_X(\mathbf{x}_k + \lambda_k \mathbf{d}_k) - \mathbf{x}^*\|^2 \\
&\leq \|\mathbf{x}_k + \lambda_k \mathbf{d}_k - \mathbf{x}^*\|^2 \\
&= \|\mathbf{x}_k - \mathbf{x}^*\|^2 + \lambda_k^2 \|\mathbf{d}_k\|^2 + 2\lambda_k \mathbf{d}_k^t (\mathbf{x}_k - \mathbf{x}^*) \\
&= \|\mathbf{x}_k - \mathbf{x}^*\|^2 + \lambda_k^2 \|\mathbf{d}_k\|^2 - 2\lambda_k R_k
\end{aligned} \tag{3.12}$$

where

$$R_k = (1 - s_k)\mathbf{g}_k^t (\mathbf{x}_k - \mathbf{x}^*) + s_k \mathbf{g}_{k-1}^t (\mathbf{x}_k - \mathbf{x}^*)$$

Since $0 \leq s_k < 1$, we have

$$R_k = (1 - s_k)\mathbf{g}_k^t(\mathbf{x}_k - \mathbf{x}^*) + s_k\mathbf{g}_{k-1}^t(\mathbf{x}_{k-1} - \mathbf{x}^*) + s_k\mathbf{g}_{k-1}^t(\mathbf{x}_k - \mathbf{x}_{k-1})$$

$$\geq (1 - s_k)(f_k - f^*) + s_k(f_{k-1} - f^*) + s_k\mathbf{g}_{k-1}^t(\mathbf{x}_k - \mathbf{x}_{k-1})$$

$$= (1 - s_k)(f_k - f^*) + s_k(f_{k-1} - f^*) + s_k(f_k - f_{k-1}) - s_k v_k$$

$$= f_k - f^* - s_k v_k$$

$$\geq f_k - w - s_k v_k$$

$$= (f_k - w)(1 - \frac{s_k v_k}{f_k - w})$$

$$\geq \beta_k(f_k - w)$$

Hence,

$$\lambda_k^2\|\mathbf{d}_k\|^2 - 2\lambda_k R_k \leq \lambda_k^2\|\mathbf{d}_k\|^2 - 2\lambda_k\beta_k(f_k - w) = -\beta_k^2\frac{(f_k - w)^2}{\|\mathbf{d}_k\|^2} < 0$$

Therefore, from (3.12), we have that $\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 < \|\mathbf{x}_k - \mathbf{x}^*\|^2$ for all k, that is, $\{\|\mathbf{x}_k - \mathbf{x}^*\|\}$ is a nonnegative, monotone decreasing sequence, and so, we must have,

$$\beta_k^2\frac{(f_k - w)^2}{\|\mathbf{d}_k\|^2} \to 0 \quad \text{as} \quad k \to \infty$$

Since $\|\mathbf{d}_k\| \leq (1 - s_k)\|\mathbf{g}_k\| + s_k\|\mathbf{g}_{k-1}\| < M$ by the boundedness of the subgradient norms, and β_k is bounded away from 0, we have that $\lim_{k\to\infty} f_k = w$. This completes the proof.■

The above theorem establishes convergence of the procedure (3.4a) and (3.10) used in conjunction with the step-length rule (3.9), given an upper bound on the optimal objective function value w that is used as a *fixed* target value. On the other hand, the following corollary addresses the convergence behavior when the upper bounding target values vary.

Corollary 3.1 Suppose that for each k, the target value w is permitted to vary, and is taken as some w_k satisfying $f_k > w_k \geq f^*$ for all k. If α_k and β_k are selected to satisfy (3.11) with w replaced by w_k, and if $\|\mathbf{g}_k\| < M$ for all k, for some M, then, for any given $\delta_1 > 0$, there exists a K_1 such that

$$f_k - w_k < \delta_1 \quad \text{for all} \quad k \geq K_1 \tag{3.13}$$

Proof Following the proof of Theorem 3.1 for the case $f_k > w_k$ for all k, we have

$$\lim_{k \to \infty} \beta_k^2 \frac{(f_k - w_k)^2}{\|\mathbf{d}_k\|^2} = 0$$

which gives $\lim_{k \to \infty}(f_k - w_k) = 0$. The assertion therefore holds true, and the proof is complete. ■

For convergence to hold in Theorem 3.1 and Corollary 3.1, we require the condition (3.11). To avoid the case of taking an anti-subgradient as the direction of motion, that is, adopting the weak direction $\mathbf{d}_k = -\mathbf{g}_k$, we may wish to restrict the choice of s_k in (3.11) to $s_k > 0$ as far as possible.

Now, to construct a procedure for selecting α_k that defines s_k via (3.5), let us first assume that for all k,

$$\|\mathbf{g}_k - \mathbf{g}_{k-1}\| \neq \mathbf{0} \quad \text{and} \quad \mathbf{g}_k^t \mathbf{r}_k \neq 0 \tag{3.14}$$

Accordingly, define

$$a_k = 1 - \frac{\|\mathbf{g}_k - \mathbf{g}_{k-1}\|}{\mathbf{g}_k^t \mathbf{r}_k} \quad \text{and} \quad \bar{a}_k = \max\{0, a_k\} \tag{3.15}$$

and consider the following results.

Lemma 3.2 For each k such that (3.14) holds, if α_k is chosen to satisfy

$$\begin{cases} \bar{a}_k < \alpha_k^2 < 1 & \text{if} \quad \mathbf{g}_k^t \mathbf{r}_k > 0 \\ 1 < \alpha_k^2 < \bar{a}_k & \text{if} \quad \mathbf{g}_k^t \mathbf{r}_k < 0 \end{cases} \tag{3.16}$$

then, $0 < s_k < 1$.

Proof If $\mathbf{g}_k^t \mathbf{r}_k > 0$, then $\alpha_k^2 < 1$ implies that $s_k > 0$ from (3.5). Since $\bar{a}_k < 1$, we have

$$1 > \alpha_k^2 > \bar{a}_k \geq 1 - \frac{\|\mathbf{g}_k - \mathbf{g}_{k-1}\|}{\mathbf{g}_k^t \mathbf{r}_k}$$

which implies that

$$(1 - \alpha_k^2) < \frac{\|\mathbf{g}_k - \mathbf{g}_{k-1}\|}{\mathbf{g}_k^t \mathbf{r}_k}$$

and hence that $0 < s_k < 1$. For the case of $\mathbf{g}_k^t \mathbf{r}_k < 0$, we again have $s_k > 0$ in (3.5) since $\alpha_k^2 > 1$. Since $\bar{a}_k = a_k > 1$, we have

$$\alpha_k^2 < a_k = 1 - \frac{\|\mathbf{g}_k - \mathbf{g}_{k-1}\|}{\mathbf{g}_k^t \mathbf{r}_k}$$

Hence,

$$(1 - \alpha_k^2) > \frac{\|\mathbf{g}_k - \mathbf{g}_{k-1}\|}{\mathbf{g}_k^t \mathbf{r}_k}$$

Therefore, we have $s_k < 1$ since $\mathbf{g}_k^t \mathbf{r}_k < 0$. This completes the proof. ∎

Now, in addition to the condition (3.14), let us assume that for each k, $v_k \neq 0$, and define

$$b_k = 1 - \frac{1}{v_k}(f_k - w_k)(1 - \varepsilon_1)\frac{\|\mathbf{g}_k - \mathbf{g}_{k-1}\|}{\mathbf{g}_k^t \mathbf{r}_k} \quad \text{and} \quad \bar{b}_k = \max\{0, b_k\} \tag{3.17}$$

for any given $0 < \varepsilon_1 < 1$ and for some target value w_k satisfying $f_k > w_k \geq f^*$. Note that if $\mathbf{g}_k^t \mathbf{r}_k > 0$, then $\bar{b}_k < 1$, and if $\mathbf{g}_k^t \mathbf{r}_k < 0$, then $\bar{b}_k = b_k > 1$.

Lemma 3.3 For each k such that $v_k \neq 0$ and (3.14) holds, if $f_k > w_k$ and if α_k is chosen to satisfy

$$\begin{cases} \bar{b}_k < \alpha_k^2 < 1, & \text{if} \quad \mathbf{g}_k^t \mathbf{r}_k > 0 \\ 1 < \alpha_k^2 < \bar{b}_k, & \text{if} \quad \mathbf{g}_k^t \mathbf{r}_k < 0 \end{cases} \tag{3.18}$$

then, we have

$$\varepsilon_{2k} \equiv 1 - \frac{s_k v_k}{f_k - w_k} > \varepsilon_1$$

Proof Observe that when $\mathbf{g}_k^t\mathbf{r}_k$ is of either sign, we have

$$s_k = (1-\alpha_k^2)\frac{\mathbf{g}_k^t\mathbf{r}_k}{\|\mathbf{g}_k-\mathbf{g}_{k-1}\|}$$

$$< (1-\bar{b}_k)\frac{\mathbf{g}_k^t\mathbf{r}_k}{\|\mathbf{g}_k-\mathbf{g}_{k-1}\|}$$

$$\leq (1-b_k)\frac{\mathbf{g}_k^t\mathbf{r}_k}{\|\mathbf{g}_k-\mathbf{g}_{k-1}\|}$$

$$= \frac{1}{v_k}(f_k-w_k)(1-\varepsilon_1)$$

This implies that

$$\frac{s_k v_k}{f_k-w_k} < 1-\varepsilon_1$$

and hence the assertion holds true. ∎

From Lemmas 3.2 and 3.3, combining (3.16) and (3.18), we see that for each k, whenever $v_k \neq 0$ and (3.14) holds, if α_k is chosen to satisfy

$$\max\{\sqrt{\bar{a}_k}, \sqrt{\bar{b}_k}\} < \alpha_k < 1, \quad \text{if} \quad \mathbf{g}_k^t\mathbf{r}_k > 0$$

$$1 < \alpha_k < \min\{\sqrt{\bar{a}_k}, \sqrt{\bar{b}_k}\}, \quad \text{if} \quad \mathbf{g}_k^t\mathbf{r}_k < 0$$

then, we will have, noting (3.11), that

$$0 < s_k < 1 \quad \text{and} \quad \varepsilon_1 < \varepsilon_{2k} \tag{3.19}$$

On the other hand, if (3.14) is violated, we can simply select $s_k = 0$ so that (3.11) holds, thereby supporting Theorem 3.1 and Corollary 3.1 in either case. Moreover, note that while defining b_k, we require that $v_k \neq 0$. However, this requirement is not necessary for the assertions of Theorem 3.1 and Corollary 3.1 to hold, since $\varepsilon_{2k} = 1$ if $v_k = 0$. Hence, to avoid the case of an undefined b_k in (3.17) when $v_k = 0$, we can set $\bar{b}_k = 0$ if $\mathbf{g}_k^t \mathbf{r}_k > 0$, and set $\bar{b}_k = \bar{M}$ for some large number $\bar{M} \geq \bar{a}_k$ if $\mathbf{g}_k^t \mathbf{r}_k < 0$. (Note that we can simply take $\bar{M} = \bar{a}_k$ in this latter case.) Then, under assumption (3.14), the above choice of α_k still guarantees (3.19). To summarize, the routine for finding the direction of motion, along with the prescribed parameters for determining the step-length, is formally stated below.

Memoryless Space Dilation and Reduction Strategy (MSDRS): Direction and Step-Length Parameter Subroutine

Let $\varepsilon_3 > 0$ and $\varepsilon_4 > 0$ be given as sufficiently small tolerances, and let $0 < \varepsilon_1 < 1$ be a chosen step-length parameter.

Step 1 If $\|\mathbf{g}_k - \mathbf{g}_{k-1}\| < \varepsilon_3$, terminate the procedure with $\mathbf{d}_k = -\mathbf{g}_k$, and let $s_k = 0$ and $\varepsilon_{2k} = 1$. Otherwise, compute $\mathbf{r}_k$ as in (3.2). If $\mathbf{g}_k^t \mathbf{r}_k = 0$, terminate the procedure with $\mathbf{d}_k = -\mathbf{g}_k$, and let $s_k = 0$ and $\varepsilon_{2k} = 1$. Otherwise, find a_k and $\bar{a}_k$ as in (3.15), and compute v_k as in (3.8). Proceed to Step 2.

Step 2 If $v_k > \varepsilon_4$, compute b_k and $\bar{b}_k$ as in (3.17) and proceed to Step 3. Otherwise, set

$$\bar{b}_k = \begin{cases} 0, & \text{if } \mathbf{g}_k^t \mathbf{r}_k > 0 \\ \bar{a}_k, & \text{if } \mathbf{g}_k^t \mathbf{r}_k < 0 \end{cases}$$

and proceed to Step 3.

Step 3 Select a suitable combination weight ϕ_k satisfying $0 < \phi_k < 1$, and compute $\bar{\alpha}_k$ as

$$\bar{\alpha}_k = \begin{cases} \max\{\sqrt{\bar{a}_k}, \sqrt{\bar{b}_k}\}, & \text{if} \quad \mathbf{g}_k^t \mathbf{r}_k > 0 \\ \min\{\sqrt{\bar{a}_k}, \sqrt{\bar{b}_k}\}, & \text{if} \quad \mathbf{g}_k^t \mathbf{r}_k < 0 \end{cases} \tag{3.20}$$

determine the transformation parameter α_k as

$$\alpha_k = \phi_k + (1 - \phi_k)\bar{\alpha}_k, \tag{3.21}$$

and proceed to Step 4.

Step 4 Compute a direction of motion $\mathbf{d}_k$ as

$$\mathbf{d}_k = -\mathbf{g}_k + (1 - \alpha_k^2)(\mathbf{g}_k^t \mathbf{r}_k)\mathbf{r}_k$$

or equivalently, as

$$\mathbf{d}_k = -[(1 - s_k)\mathbf{g}_k + s_k \mathbf{g}_{k-1}],$$

where s_k is defined as

$$s_k = (1 - \alpha_k^2)\frac{\mathbf{g}_k^t \mathbf{r}_k}{\|\mathbf{g}_k - \mathbf{g}_{k-1}\|}, \quad 0 < s_k < 1$$

Compute

$$\varepsilon_{2k} = 1 - \frac{s_k v_k}{f_k - w},$$

and terminate the procedure.

The overall algorithm for which Theorem 3.1 and Corollary 3.1 hold true can then be stated as follows. At the k-th iteration of the prescribed procedure, assume that we have a current subgradient $\mathbf{g}_k$ and a previous subgradient $\mathbf{g}_{k-1}$ (when $k \geq 2$). Then the direction of motion $\mathbf{d}_k$ can be computed by the foregoing subroutine MSDRS when $k \geq 2$, with $\mathbf{d}_1$ taken simply as $-\mathbf{g}_1$. Computing the step-length by (3.9), where the parameter β_k is chosen to satisfy (3.11), and where s_k is given by the subroutine MSDRS, the new iterate is determined via (3.10), and the process is repeated.

We remark here that the result in Theorem 3.1 requires the target value to be chosen as an upper bound on the optimal objective value. On the other hand, when the target value is a lower bound on the optimal value, an analogous result to that of Allen et al. (1987) for the pure-subgradient strategy is elusive. However, by imbedding the foregoing scheme within the framework of a variable target value method as described in Chapter 2, we can construct a convergent algorithm. As shown in Chapter 2, since the property of Theorem 3.1 and Corollary 3.1 holds true, this procedure will generate a monotone, non-increasing sequence of incumbent objective values $\{z_k\}$ that converges to an ε-neighborhood of the optimal objective function value, for any chosen $\varepsilon > 0$, without any *a priori* knowledge about the optimal objective function value. Accompanying this scheme is a sequence of target values $\{w_\ell\}$ generated via an outer loop of the procedure, which guides the step-length selection process defined by (3.9) during the inner loop iterations. This inner loop of the procedure, which is stated below, attempts to hit the current target value within a variable tolerance, and depending on whether or not it is successful, it readjusts the target value along with the tolerance in an outer loop update. For the sake of completeness, the specific

details of this algorithm are provided below, suitably modified to accommodate the subroutine MSDRS.

Variable Target Memoryless Space Dilatation/Reduction Algorithm (VT-MSDRS)

Initialization Select values for the parameters ε_0, ε, ε_1, ε_3, ε_4, $\sigma^{(1)}$, $\sigma^{(2)}$, $\gamma^{(1)}$, $\gamma^{(2)}$, $\psi^{(1)} \geq \varepsilon_1$, and $\psi^{(2)}$, and pick termination tolerances $\bar{\tau}$ and k_{max}, which respectively represent the maximum number of consecutive target increases, and a limit on the number of iterations. (See recommended values given below, following the statement of the algorithm.) Select a starting solution $\mathbf{x}_0 \in X$, compute its objective function value f_0, and let the initial direction of motion be $\mathbf{d}_0 = -\mathbf{g}_0$, where $\mathbf{g}_0$ is a subgradient of f at $\mathbf{x}_0$. Set $\varepsilon_{20} = 1$. If $\|\mathbf{g}_0\| < \varepsilon_0$, then stop with $\mathbf{x}_0$ as a near-optimal solution. Otherwise, set $\mathbf{x}^* = \mathbf{x}_0$ and $\mathbf{g}^* = \mathbf{g}_0$, and record $z_0 = f_0$ as the best known objective function value. Initialize the outer loop counter $\ell = 1$, and compute the initial target value $w_1 = \max\{\text{LB}, f_0 - \|\mathbf{d}_0\|^2/2\}$, where LB is any known lower bound on f^*, being taken as $-\infty$ if no such lower bound is available. Also, select the initial acceptance tolerance for measuring proximity to the present target value as $\varepsilon(1) = (\sigma^{(1)} + \sigma^{(2)})(f_0 - w_1)$. Set the counter of consecutive target increases to $\tau = 0$, the consecutive failure counter to $\gamma = 0$, and the inner loop counter to $k = 0$. Put $\Delta = 0$, where Δ measures accumulated improvements within the inner loop iterations.

Step 1(a) (Outer Loop Parameter Updates) Compute $\sigma_\ell = \sigma^{(1)} + \sigma^{(2)}\theta^{(1-\ell)}$, $\bar{\gamma}_\ell = \gamma^{(1)} + \gamma^{(2)}\theta^{(1-\ell)}$, and $\psi_\ell = \psi^{(1)} + \psi^{(2)}\theta^{(1-\ell)}$.

Step 1(b) (Inner Loop Main Iteration) If $k = k_{max}$, stop. Else, compute the step-length

$$\lambda_k = \beta_k \frac{f_k - w_\ell}{\|\mathbf{d}_k\|^2}, \quad \text{where} \quad \beta_k = \text{minimum}\{\varepsilon_{2k}, \psi_\ell\}$$

Find the new iterate $\mathbf{x}_{k+1} = P_X[\mathbf{x}_k + \lambda_k \mathbf{d}_k]$, evaluate the objective function value f_{k+1}, and find a subgradient $\mathbf{g}_{k+1}$ of f at $\mathbf{x}_{k+1}$. If $\|\mathbf{g}_{k+1}\| < \varepsilon_0$, terminate the algorithm with $\mathbf{x}_{k+1}$ as (near) optimal solution. If $f_{k+1} < z_k$, update $\Delta \leftarrow \Delta + (z_k - f_{k+1})$, and go to Step 2(a). Otherwise, go to Step 3(a).

Step 2(a) (Improvement Case) Put $\gamma = 0$, $z_{k+1} = f_{k+1}$, and update $\mathbf{x}^* = \mathbf{x}_{k+1}$ and $\mathbf{g}^* = \mathbf{g}_{k+1}$. If $z_{k+1} \leq w_\ell + \varepsilon(\ell)$, then go to Step 2(b). Otherwise, increment k by one, call subroutine MSDRS to compute the new direction of motion $\mathbf{d}_k$ along with the value of the step-length parameter ε_{2k}, and return to Step 1(b).

Step 2(b) (Outer Loop Update; Decrease Target Value and Adjust Acceptance Tolerance) Let $\eta = 0.5 + 0.5e^{-\ell/10}$ and compute a new target value as

$$w_{\ell+1} = (z_{k+1} - \varepsilon(\ell)) - \eta\Delta$$

and let

$$\varepsilon(\ell+1) = \max\{(z_{k+1} - w_{\ell+1})\sigma_\ell, \ \varepsilon\}$$

Put $\tau = 0$ and $\Delta = 0$, increment ℓ and k by one, call subroutine MSDRS to compute the new direction of motion and the step-length parameter ε_{2k}, and return to Step 1(a).

Step 3(a) (Failure Case) Put $z_{k+1} = z_k$, and increment γ by one. If $\gamma \geq \bar{\gamma}_\ell$, go to Step 3(b). Otherwise, increment k by one, call subroutine MSDRS to compute the new direction of motion and step-length parameter ε_{2k}, and return to Step 1(b).

Step 3(b) (Outer Loop Update; Increase Target Value and Adjust Acceptance Tolerance) Compute a new target value as

$$w_{\ell+1} = \frac{(z_{k+1} - \varepsilon(\ell)) + w_\ell}{2}$$

and let

$$\varepsilon(\ell+1) = \max\{(z_{k+1} - w_{\ell+1})\sigma_\ell,\ \varepsilon\}$$

Increment τ by one. If $\tau = \bar{\tau}$, terminate the algorithm. Otherwise, put $\gamma = 0$ and $\Delta = 0$, increment ℓ and k by one, and reset to $\mathbf{x}_k = \mathbf{x}^*$ and $\mathbf{g}_k = \mathbf{g}^*$. Let the new direction of motion be chosen as the anti-subgradient, that is, $\mathbf{d}_k = -\mathbf{g}_k$, and let $s_k = 0$ and $\varepsilon_{2k} = 1$. Return to Step 1(a).

Recommended Parameter Values $\varepsilon_0 = 10^{-6}$, $\varepsilon = \varepsilon_3 = \varepsilon_4 = 0.1$, $(\gamma^{(1)}, \gamma^{(2)}) = (50, 10)$, $(\sigma^{(1)}, \sigma^{(2)}) = (0.1, 0.5)$, and $\varepsilon_1 = \psi^{(1)}$. (For the values of $k_{\max}$, $\bar{\tau}$, $\psi^{(1)}$, and $\psi^{(2)}$, see Chapter 2 for general selection guidelines. The following section recommends suitable values for the test problems considered herein.)

3.4 Computational Experience

For the purpose of computational testing, we have chosen some dual transportation and dual assignment problems that have been used by Shor (1985) to test his r-algorithm. The transportation problems for n origins and n destinations have the form

$$\text{Minimize}\left\{\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}v_{ij} : \sum_{j=1}^{n} v_{ij} = S_i,\ \ 1 \le i \le n,\ \ \sum_{i=1}^{n} v_{ij} = D_j,\ \ 1 \le j \le n,\ \ v_{ij} \ge 0,\ \ \text{for all}\ \ i,j\right\}$$

where c_{ij} is a per-unit cost for the flow v_{ij} on link (i,j) from origin i to destination j, S_i is the supply at origin i, and D_j is the demand at destination j. The linear programming dual to this problem is given by

$$\text{Maximize}\left\{\sum_{i=1}^{n} x_i S_i + \sum_{j=1}^{n} y_j D_j : x_i + y_j \le c_{ij}\ \ \text{for all}\ \ i,j\right\}$$

where **x** and **y** are the dual vectors to the supply and demand constraints, respectively. For a given **x**, the optimal resulting solution **y** can be found as

$$y_j = \text{minimum}\{c_{ij} - x_i : i = 1,..., n\} \quad \text{for each}\ \ j = 1,...., n$$

Thus, the dual transportation problem, which is our test problem, can be written in its projected form in the **x**-space as

$$\underline{\text{TR(n)}}\text{: Minimize}\left[-\sum_{i=1}^{n} S_i x_i + \sum_{j=1}^{n} D_j \underset{1 \le i \le n}{\text{maximum}}(x_i - c_{ij})\right]$$

The primal transportation problems were generated via the linear programming problem generation technique described in Rosen and Suzuki (1965) and Sherali anf Myers (1988). The assignment problems, denoted by AS(n), were generated similarly, but with $S_i = 1$ for all i and $D_j = 1$ for all j.

All the algorithms tested were coded in FORTRAN VS II, and run on an IBM 3090, Model 300E computer. For the computations, we have used the parameters $(k_{max}, \bar{\tau}) = (2000, 30)$, along with

$$\psi^{(1)} = \text{minimum}\{0.1,\ 10^{-[n/50]}\} \quad \text{and} \quad \psi^{(2)} = \frac{\psi^{(1)}}{10}$$

where $[n/50]$ denotes the greatest integer less than or equal to $n/50$. Also, we fixed $\phi_k = 0.5$ in Equation (3.21). Other parameter values were selected as recommended at the end of the algorithmic statement in Section 3.

Tables 3.1 and 3.2 present computational results for Algorithm VT-MSDRS on the dual transportation and assignment problems, respectively. In the tables, f^* denotes the (known) optimal objective function value, $f(\mathbf{x}_{best})$ denotes the best objective function value obtained, ITR(CPU) denotes the iteration number at which the best recorded solution was found within the corresponding iteration limit (which coincides with the number of functional evaluations), with the cpu time in seconds given in parenthesis. Also, OPT(%) represents the percentage optimality achieved, computed via the formula

$$\text{OPT}(\%) = \left| \frac{f(\mathbf{x}_{best})}{f^*} \right| \times 100$$

The results indicate that Algorithm VT-MSDRS performs uniformly well on the test problems, without the need for any fine tuning of parameters for different problems. Note that these parameter choices were also found to perform consistently well on several other types of test problems solved in Chapter 2, using other subgradient deflection strategies. Furthermore, we found that the algorithm VT-MSDRS was rel-

atively insensitive to the choice of the convex combination weight ϕ_k within the interval [1/2, 1). For both TR(n) and AS(n), we can see that the problems become harder when n grows, with the loss in accuracy for the assignment problems using the same iteration limit being slightly more pronounced than for the transportation problems.

For the sake of comparison, we also present some test results for the r-algorithm in Tables 3 and 4. For these runs, we attempted to use two prescribed step-length rules, namely (i) $\lambda_k = h_k/\|\mathbf{d}_k\|$ and (ii) $\lambda_k = h_k$, where for both rules, $h_k = h_0\delta^{(k-1)}$ for some $h_0 > 0$ and $0 < \delta < 1$ (see Shor and Shabashova, 1972, and Lemarechal, 1982). After several trial runs, we found that the rule (i) could be easily controlled and that it generated better solutions. Also, for a fixed $\delta = 0.95$, we found that the algorithm performed best whenever $6.0 \leq h_0 \leq 7.5$. The results in Table 3.3 are obtained under rule (i) when $h_0 = 7$, $\delta = 0.95$, the dilation parameter $\alpha = 3$, and the maximum iteration limit $k_{max} = 2000$.

As we can see from Tables 3.3 and 3.4, the CPU time grows exponentially with an increase in problem dimension. Note that when $n \geq 150$, the r-algorithm takes more than 1500 CPU seconds to solve, and thus we do not present these results. Moreover, by comparing Tables 3.1 and 3.3 with Tables 3.2 and 3.4, respectively, the superiority of Algorithm VT-MSDRS over the r-algorithm with respect to both accuracy and computational effort is clearly evident. Therefore, the proposed memoryless update not only reduces the computational burden (storage plus cpu time), but also improves the computational performance when embedded within the recommended variable target value method.

To summarize, we have presented in this chapter a memoryless variant of Shor's popular r-algorithm that employs a space dilation or reduction in the direction of the difference between two successive subgradient directions. For this variant, we have described a choice of parameter values for which we have established convergence of the procedure under the use of a desirable step-length rule along with variable upper bounding target values. This convergence property permitted us to embed the proposed scheme within the variable target value method of Chapter 2, which assumes no *a priori* knowledge of the optimal objective function value. Computational results have shown a dramatic improvement of the proposed scheme over Shor's r-algorithm.

Table 3.1. Computational results for VT-MSDRS on dual transportation problems

Problem	f^*	$f(x_{best})$	ITR(CPU)	OPT(%)
TR(20)	-1771	-1770.83	1792(1.08)	99.99
TR(30)	-3511	-3510.73	1523(1.93)	99.99
TR(50)	-5154	-5138.92	1995(6.10)	99.71
TR(80)	-8147	-8035.00	1993(14.58)	98.63
TR(100)	-9519	-9515.11	1997(23.33)	99.97
TR(120)	-11170	-11160.59	2000(31.85)	99.92
TR(150)	-13664	-13646.05	2000(48.93)	99.87
TR(180)	-16523	-16493.04	1998(69.55)	99.82
TR(200)	-19315	-19098.10	1972(82.76)	98.88

Table 3.2. Computational results for VT-MSDRS on dual assignment problems

Problem	f^*	$f(x_{best})$	ITR(CPU)	OPT(%)
AS(20)	-124	-123.99	1200(0.74)	99.99
AS(30)	-210	-209.98	1318(1.61)	99.99
AS(50)	-422	-421.96	1807(5.69)	99.99
AS(80)	-748	-747.93	1732(12.63)	99.99
AS(100)	-948	-947.93	1735(19.28)	99.99
AS(120)	-1100	-1099.82	1981(31.11)	99.98
AS(150)	-1416	-1323.45	1985(48.05)	93.46
AS(180)	-1672	-1552.97	1982(68.39)	92.88
AS(200)	-1856	-1697.24	1997(84.73)	91.45

Table 3.3. Computational results for the r-algorithm on dual transportation problems

Problem	f^*	$f(x_{best})$	ITR(CPU)	OPT(%)
TR(20)	-1771	-1759.90	78(1.25)	99.36
TR(30)	-3511	-3414.50	66(3.34)	97.25
TR(50)	-5154	-4991.70	2000(43.86)	96.86
TR(80)	-8147	-7617.51	236(158.04)	93.50
TR(100)	-9519	-8764.68	280(364.89)	92.08
TR(120)	-11170	-9710.49	2000(621.52)	86.92

Table 3.4. Computational results for the r-algorithm on dual assignment problems

Problem	f^*	$f(x_{best})$	ITR(CPU)	OPT(%)
TR(20)	-124	-122.90	85(1.40)	99.12
TR(30)	-210	-203.44	96(4.98)	96.88
TR(50)	-422	-408.57	2000(62.41)	96.82
TR(80)	-748	-695.27	311(203.84)	92.95
TR(100)	-948	-854.27	2000(391.81)	90.11
TR(120)	-1100	-935.89	2000(675.09)	85.08

4. Primal Recovery Theorems for Lagrangian Dual Subgradient Methods

4.1 Introduction

Lagrangian Dual (LD) methods are widely used approaches for solving specially structured constrained problems through the process of dualizing complicating constraints in order to obtain easier (than the original) subproblems. These subproblems give a lower bound on the original (minimization) problem value, and an ascent scheme attempts to find a greatest lower bound. For linear programming relaxations of discrete problems, Lagrangian dual techniques can be quite useful in obtaining bounds within the context of branch-and-bound methods, since these linear programming relaxations frequently turn out to be highly degenerate, and so LP solvers such as the simplex method experience great difficulties in solving them. However, a number of packages such as XMP, MINTO, CPLEX, MPS-MIP, and OSL, all still use simplex routines for solving these LP relaxations. On the other hand, specialized al-

gorithms such as those derived by Adams and Sherali (1992) for mixed-integer bilinear programming problems have amply demonstrated the relative advantage of using Lagrangian dual based branch-and-bound approaches.

Starting with the saddle point optimality conditions of Kuhn and Tucker (1951), the Lagrangian duality concept was first adopted by Everett (1963) as an algorithmic scheme for solving constrained problems as an infinite sequence of unconstrained problems. Falk (1967) presents the use of Lagrangian duality for nonlinear programming. For discrete or combinatorial problems, Lagrangian duality has received a great deal of attention after the frontier work of Geoffrion (1970, 1971, 1974) under the name of Lagrangian Relaxation. In particular, the work of Held and Karp (1970, 1971) on traveling salesman problems provided the impetus for employing Lagrangian dual approaches for discrete problems in conjunction with subgradient methods.

There are several issues that we need to consider in Lagrangian dual approaches. Among them, dualization strategies, solution methods, and primal recovery techniques are the most important ones. In discrete problems, the quality of the lower bound obtained by solving Lagrangian dual problems depends on the dualization strategy because of the ubiquitous presence of a duality gap. These relationships are well explained in Fisher (1981), Parker and Rardin (1988), Sherali and Myers (1988), and Nemhauser and Wolsey (1989). In particular, Sherali and Myers (1988) present some computational results under different dualization strategies.

There are several strategies for solving Lagrangian dual problems. These solution methods can be divided into two large classes, namely, cutting plane methods and subgradient based methods. In related approaches, several heuristic multiplier ad-

justment techniques have been proposed by Erlenkotter (1978), Fisher, Jaikumar, and Van Wassenhove (1986), and Fisher and Kedia (1990). However, each of these methods are designed for specific problems by exploiting their special structures, and hence are not applicable to general problems. On the other hand, the subgradient based methods, introduced by Held and Karp (1971) and Held, Wolfe, and Crowder (1974), are the most popular, quite promising, and easy to implement in practice. However, for problems lacking any special structure, these methods can stall far from optimality.

In our discussion, we assume that a Lagrangian dual problem formulation has already been selected for a given primal problem, and that some subgradient based method is to be used to solve it. We turn our attention to the task of finding a primal optimal solution from the dual subgradient procedure. To this end, let us assume that a primal problem is given as

P Minimize $\mathbf{cx}$

subject to $\mathbf{Ax} \leq \mathbf{b}$

$\mathbf{x} \in X$

where X is a nonempty polytope in E_n. Here, we can also assume that X has some special structure that can be advantageously exploited, or that the underlying linear programming problem is relatively easy to solve over the set X. The Lagrangian dual for Problem P can be written as follows:

LD Maximize $\theta(\boldsymbol{\pi})$

subject to $\boldsymbol{\pi} \geq \mathbf{0}$

where $\theta(\boldsymbol{\pi})$ is evaluated via the Lagrangian subproblem

LS(π) $\quad \theta(\pi) = \text{minimum } \{\mathbf{cx} + \pi^t(\mathbf{Ax} - \mathbf{b}) : \mathbf{x} \in X\}$

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 a given dual solution, $\pi \geq \mathbf{0}$, let us define

$$X_\pi = \{\mathbf{x}_\pi : \mathbf{x}_\pi \text{ minimizes } \mathbf{cx} + \pi^t(\mathbf{Ax} - \mathbf{b}) \text{ over } X\} \subseteq X$$

as the set of optimal solutions for the subproblem LS(π). Then the subdifferential $\partial\theta(\pi)$ of θ at π is given by $\{\mathbf{Ax}_\pi - \mathbf{b} : \mathbf{x}_\pi \in X_\pi\}$. Note that given π_k at some iteration k, a (pure) subgradient method would use a subgradient $\mathbf{g}_k$ of θ at π_k as the direction of motion, and hence generate a new iterate

$$(\pi_{k+1})_i = \text{maximum } \{0, \ (\pi_k + \lambda_k \mathbf{g}_k)_i\} \tag{4.1}$$

for each component i, where λ_k is a suitable step-length.

Since the dual problem only provides a lower bound on the primal optimal objective value, we need some device for recovering a primal solution, other than using expensive projection operations within the dual procedure. While the subgradient methods are quite promising for solving the dual problem itself, in general, the optimal solutions to the subproblems need not be feasible to the primal problem, even for the subproblem corresponding to an optimal dual solution. In contrast, for the cutting plane methods, an optimal dual solution to the master problem (MP) can be used as a set of convex combination weights applied to the optimal solutions to the corresponding Lagrangian subproblems in order to find a primal optimal solution (see Bazaraa, Sherali, and Shetty, 1993). However, we can indeed deduce a similar result

from the dual subgradient methods. Shor (1985) presents a convex combination weighting rule as a function of the step-lengths in a dual subgradient method that leads the sequence of convex combinations of the optimal Lagrangian subproblem solutions to converge to a primal optimal solution. Also, Larsson and Liu (1989) use an average weighting scheme along with a particular step-length rule that is a member of Shor's class of step sizes, and exhibit a similar result. In this effort, we present a convergence theorem that generalizes these previous results, and provides a more flexible choice of admissible step-length rules and convex combination weighting strategies. Moreover, this result permits one to examine more computationally promising step-length rules than the ones considered heretofore.

This chapter is organized as follows. In Section 2, we provide generalized primal convergence theorems along with some specific step-length rules and convex combination weighting rules. Also, these results are extended for the case of using deflected subgradient methods.

4.2 Primal Convergence Theorems for Dual Subgradient Methods

At the k-th iteration of the dual subgradient method (4.1), let us define

$$\mathbf{x}_k = \sum_{j=1}^{k} \mu_j^k \mathbf{x}_{\pi_j} \quad \text{where} \quad \mu_j^k \geq 0 \quad \text{for} \quad 1 \leq j \leq k \quad \text{and} \quad \sum_{j=1}^{k} \mu_j^k = 1 \tag{4.2}$$

Hence, each $\mathbf{x}_k$ is a convex combination of optimal solutions to the Lagrangian subproblems LS(π_j) for $j = 1,..., k$. Shor(1985) presents a primal convergence theorem using a step-length rule given by a divergent series satisfying

$$\lambda_k \geq 0, \quad \lim_{k\to\infty} \lambda_k = 0, \quad \text{and} \quad \sum_{k=1}^{\infty} \lambda_k = \infty, \tag{4.3}$$

and a convex combination weighting rule given by

$$\mu_j^k = \frac{\lambda_j}{\sum_{j=1}^{k} \lambda_j} \quad \text{for all} \quad 1 \leq j \leq k \tag{4.4}$$

Note here that the step-length rule (4.3) guarantees dual convergence (see Ermolev, 1966, Held and Karp, 1974, and Shor, 1985). Following this, Larsson and Liu (1989) provided a similar result to that of Shor's by using the average weighting rule

$$\mu_j^k = \frac{1}{k} \quad \text{for all} \quad 1 \leq j \leq k \tag{4.5}$$

under a special step-length choice $\lambda_k = a/(b + ck)$ for the dual subgradient method, where $a, c > 0$ and $b \geq 0$ are some numbers. The average weighting rule (4.5) seems more reasonable than that of (4.4), because Shor's rule (4.4) assigns larger weights to the earlier solutions, since $\lim_{k\to\infty} \lambda_k = 0$. However, Larsson and Liu's step-length rule is unpopular and untested for general problems; in other words, no computational performance for general Lagrangian dual problem has been reported. (Their step-length rule has been used in their primal-dual heuristic algorithm for multicommodity flow problems.) Moreover, it belong to the class of Shor's step-length rule (4.3), as indicated by the following remark.

Remark 4.1 Suppose that $\lambda_k = a/(b + ck)$ for all k, where $a, c > 0$ and $b \geq 0$, then clearly, we have $\lim_{k \to \infty} \lambda_k = 0$. Furthermore, we have

$$\frac{a}{b + ck} \geq \frac{a}{(b + c)k} \quad \text{for all} \quad k$$

Hence,

$$\sum_{k=1}^{\infty} \lambda_k \geq \sum_{k=1}^{\infty} \left(\frac{a}{b + c}\right) \frac{1}{k} = \infty$$

Therefore, the step-length rule given by Larsson and Liu (1989) satisfies (4.3). ■

Considering the weakness of Larsson and Liu's procedure, we provide a primal convergence theorem for a wider class of step-length rules and convex combination weighting rules. To this end, let us define for each k

$$\gamma_{jk} = \frac{\mu_j^k}{\lambda_j} \quad \text{for} \quad j = 1, \ldots, k$$

where λ_j and μ_j^k are the step-lengths and the convex combination weights, respectively, and let

$$\Delta\gamma_k^{\max} = \text{maximum } \{\gamma_{jk} - \gamma_{(j-1)k} : 2 \leq j \leq k\}.$$

Consider the following results.

Theorem 4.1 Assume that the subgradient method (4.1) operated with a suitable step-length rule attains dual convergence to some feasible solution, that is, $\pi_k \to \bar{\pi}$ as $k \to \infty$ for some $\bar{\pi} \geq \mathbf{0}$. If the step-lengths λ_k and the convex combination weights μ_j^k are chosen to satisfy

(i) $\gamma_{jk} \geq \gamma_{(j-1)k}$ for all $j = 2,\ldots, k$, for each k,

(ii) $\Delta\gamma_k^{\max} \to 0$ as $k \to \infty$, and

(iii) $\gamma_{1k} \to 0$ as $k \to \infty$ and either (a) $\gamma_{kk} \to \delta < \infty$ as $k \to \infty$ or (b) $\gamma_{kk} \leq \delta$ for all k large enough, for some $\delta \geq 0$.

Then any accumulation point $\bar{\mathbf{x}}$ of the sequence $\{\mathbf{x}_k\}$ is feasible to Problem P.

Proof By the linearity of the constraints in Problem P and the fact that $\sum_{j=1}^{k} \mu_j^k = 1$, we have

$$\mathbf{A}\mathbf{x}_k - \mathbf{b} = \sum_{j=1}^{k} \mu_j^k [\mathbf{A}\mathbf{x}_{\pi_j} - \mathbf{b}] = \sum_{j=1}^{k} \mu_j^k \mathbf{g}_j$$

where $\mathbf{g}_j$ is a subgradient of θ at π_j. Since $\pi_{j+1} \geq \pi_j + \lambda_j \mathbf{g}_j$ by (4.1), we have

$$\frac{1}{\lambda_j}(\pi_{j+1} - \pi_j) \geq \mathbf{g}_j \quad \text{for all} \quad 1 \leq j \leq k$$

Thus, for all k, we have, noting condition (i),

$$\mathbf{A}\mathbf{x}_k - \mathbf{b} \leq \sum_{j=1}^{k} \mu_j^k \frac{1}{\lambda_j}(\pi_{j+1} - \pi_j)$$

$$= -\frac{\mu_1^k}{\lambda_1}\pi_1 + \sum_{j=2}^{k}\left(\frac{\mu_{j-1}^k}{\lambda_{j-1}} - \frac{\mu_j^k}{\lambda_j}\right)\pi_j + \frac{\mu_k^k}{\lambda_k}\pi_{k+1}$$

$$= -\gamma_{1k}\pi_1 - \sum_{j=2}^{k}(\gamma_{jk} - \gamma_{(j-1)k})\pi_j + \gamma_{kk}\pi_{k+1} \leq \gamma_{kk}\pi_{k+1} \tag{4.5}$$

Since $\{\mathbf{x}_k\}$ is contained in a compact set, there exists a convergent subsequence. Let $\bar{\mathbf{x}}$ be any accumulation point, and let K be the index set for which this subsequence converges to $\bar{\mathbf{x}}$, that is, $\lim_{k \in K, k \to \infty} \mathbf{x}_k = \bar{\mathbf{x}}$. Hence, if $\delta = 0$ in (iii), we have taking limits as $k \to \infty$, $k \in K$ in (4.5), that $\mathbf{A}\bar{\mathbf{x}} \leq \mathbf{b}$. On the other hand, if $\delta > 0$, then from (4.5), we get

$$\mathbf{A}\mathbf{x}_k - \mathbf{b} \leq \gamma_{kk}\pi_{k+1} - \bar{\pi}\sum_{j=2}^{k}(\gamma_{jk} - \gamma_{(j-1)k}) + \sum_{j=2}^{k}(\gamma_{jk} - \gamma_{(j-1)k})(\bar{\pi} - \pi_j)$$

$$= \gamma_{kk}(\pi_{k+1} - \bar{\pi}) + \gamma_{1k}\bar{\pi} + \sum_{j=2}^{k}(\gamma_{jk} - \gamma_{(j-1)k})(\bar{\pi} - \pi_j) \tag{4.7}$$

Let $\mathbf{v}_k$ denote the last term in (4.7), and let us show that $\mathbf{v}_k \to \mathbf{0}$ as $k \to \infty$. Toward this end, given any $\varepsilon > 0$, let $J \geq 2$ be large enough so that $\|\bar{\pi} - \pi_j\| \leq \varepsilon/4\delta$ for all $j > J$. Then, for $k \geq J$ and large enough so that, noting (ii) and (iii), $\Delta\gamma_k^{\max}\sum_{j=2}^{J}\|\bar{\pi} - \pi_j\| \leq \varepsilon/2$ and $\gamma_{kk} \leq 2\delta$, we get by (i) and the triangle inequality that

$$\|\mathbf{v}_k\| \leq \sum_{j=2}^{k}(\gamma_{jk} - \gamma_{(j-1)k})\|\bar{\pi} - \pi_j\|$$

$$\leq \Delta\gamma_k^{\max} \sum_{j=2}^{J} \|\bar{\pi} - \pi_j\| + \frac{\varepsilon}{4\delta} \sum_{j=J+1}^{k} (\gamma_{jk} - \gamma_{(J-1)k})$$

$$\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{4\delta} \gamma_{kk} \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$$

Since $\varepsilon > 0$ was arbitrary, $\|\mathbf{v}_k\| \to 0$, and so $\mathbf{v}_k \to \mathbf{0}$ as $k \to \infty$. Using this in (4.7) along with (iii), we get upon taking limit as $k \to \infty$, $k \in K$, that $\mathbf{A}\bar{\mathbf{x}} \leq \mathbf{b}$. Since $\bar{\mathbf{x}} \in X$, this means that $\bar{\mathbf{x}}$ is feasible to Problem P, and the proof is complete. ■

Theorem 4.2 Assume that the subgradient method (4.1) operated with a suitable step-length rule attains dual convergence to some feasible solution, that is, $\pi_k \to \bar{\pi}$ as $k \to \infty$ for some $\bar{\pi} \geq \mathbf{0}$. Then, there exists a J such that for all $k \geq J$,

$$\partial\theta(\pi_k) \subseteq \partial\theta(\bar{\pi}) \tag{4.8}$$

and

$$(\pi_k)_i > 0 \quad \text{and} \quad (\mathbf{A}\mathbf{x}_{\pi_k} - \mathbf{b})_i = \frac{(\pi_{k+1})_i - (\pi_k)_i}{\lambda_k} \tag{4.9}$$

for any component i with $(\bar{\pi})_i > 0$. Moreover, if the primal iterates are generated for $k \geq J$ by using a set of convex combination weights μ_j^k that satisfy, in combination with the step-lengths, the conditions (i), (ii), and (iii) of Theorem 4.1, then any accumulation point $\bar{\mathbf{x}}$ of the sequence $\{\mathbf{x}_k\}$ along with $\bar{\pi}$ yield a pair of primal and dual optimal solutions, respectively.

Proof Since θ is piecewise linear and concave, there exists a neighborhood $N(\bar{\pi})$ of $\bar{\pi}$ such that $\partial\theta(\pi)\subseteq\partial\theta(\bar{\pi})$ for all $\pi \in N(\bar{\pi})$. Hence, $\pi_k \to \bar{\pi}$ as $k \to \infty$ implies that there exists a J_1 large enough so that $\pi_k \in N(\bar{\pi})$ for all $k \geq J_1$. Thus, we have $\mathbf{A}\mathbf{x}_{\pi_k} - \mathbf{b} \in \partial\theta(\pi_k)\subseteq\partial\theta(\bar{\pi})$ for all $k \geq J_1$. Now, if $(\bar{\pi})_i > 0$ for some component i, then since $\pi_k \to \bar{\pi}$ as $k \to \infty$, there exist a J_2^i large enough so that for $k \geq J_2^i$,

$$(\pi_k)_i > 0 \quad \text{and} \quad (\mathbf{A}\mathbf{x}_{\pi_k} - \mathbf{b})_i = \frac{(\pi_{k+1})_i - (\pi_k)_i}{\lambda_k} \tag{4.10}$$

Let us define

$$J = \text{maximum}\{J_1, \ \underset{i}{\text{maximum}}\{J_2^i : (\bar{\pi})_i > 0\}\}.$$

Then for all $k \geq J$, we have the assertions (4.8) and (4.9) holding true.

Now, if the primal iterates are generated for $k \geq J$, then we have that $\mathbf{x}_k$ is a solution to the Lagrangian subproblem LS($\bar{\pi}$), that is, $\mathbf{x}_k \in X_{\bar{\pi}}$ for all $k \geq J$. Therefore, $\bar{\mathbf{x}}$ defined by the theorem belongs to $X_{\bar{\pi}}$. Moreover, by Theorem 4.1, $\bar{\mathbf{x}}$ is feasible to the primal problem. Hence, by the strong duality theorem, it remains to show that the complementary slackness condition holds true in order to complete the proof. For convenience in notation, let us reset the indices as $J + j \leftarrow j$ for all $j \geq 0$, pretending to restart the dual procedure from $\pi_0 \equiv \pi_J$. If $(\bar{\pi})_i > 0$ for some component i, we have

$$(\mathbf{A}\mathbf{x}_k - \mathbf{b})_i = \sum_{j=1}^{k} \mu_j^k (\mathbf{A}\mathbf{x}_{\pi_j} - \mathbf{b})_i = \sum_{j=1}^{k} \mu_j^k (\mathbf{g}_j)_i = \sum_{j=1}^{k} \frac{\mu_j^k}{\lambda_j} (\pi_{j+1} - \pi_j)_i$$

for all k. Thus, if $\delta = 0$ in (iii), we have by (6) that

$$(\mathbf{A}\mathbf{x}_k - \mathbf{b})_i = -\gamma_{1k}(\pi_1)_i - \sum_{j=2}^{k}(\gamma_{jk} - \gamma_{(j-1)k})(\pi_j)_i + \gamma_{kk}(\pi_{k+1})_i \tag{4.11}$$

Note that the second term in (4.11) can be written as

$$(\bar{\pi})_i(\gamma_{kk} - \gamma_{1k}) + \sum_{j=2}^{k}(\gamma_{jk} - \gamma_{(j-1)k})[(\pi_j)_i - (\bar{\pi})_i] \tag{4.12}$$

Now, we can see that the last term in (4.12) converges to 0 as $k \to \infty$ using the same argument in the proof of Theorem 4.1, and so, by (iii) and since $\delta = 0$, the expression in (4.12) as well as that in (4.11) converge to 0 as $k \to \infty$. Therefore, we have that $(\mathbf{A}\bar{\mathbf{x}} - \mathbf{b})_i = 0$.

On the other hand, if $\delta > 0$ in (iii), we have from (4.11) and (4.12) that

$$(\mathbf{A}\mathbf{x}_k - \mathbf{b})_i = -\gamma_{1k}(\pi_1)_i + \gamma_{kk}(\pi_{k+1} - \bar{\pi})_i + (\bar{\pi})_i\gamma_{1k} + \sum_{j=2}^{k}(\gamma_{jk} - \gamma_{(j-1)k})(\bar{\pi} - \pi_j)_i$$

Again, by the same argument as in the proof of Theorem 4.1, we can show that the last term in the above equation converges to 0. Therefore, by taking limits as $k \to \infty$, $k \in K$, we have $(\mathbf{A}\bar{\mathbf{x}} - \mathbf{b})_i = 0$, and hence the complementary slackness condition holds true. This completes the proof. ■

Remark 4.2 In the above proof, the complementary slackness condition automatically holds if the constraints are given by equalities, that is, if $\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$ in Problem P.

Note that in this case, the feasibility of $\bar{\mathbf{x}}$ to the equality constraints easily follows from the arguments in the proof Theorem 4.1.

The following corollaries show us that Shor's and Larsson and Liu's rules are special cases of Theorem 4.1.

Corollary 4.1 If the step-lengths and convex combination weights are given by (4.3) and (4.4), respectively, then the conditions (i), (ii), and (iii) hold.

Proof Note that $\gamma_{jk} = 1/\sum_{t=1}^{k} \lambda_t$ for all $j = 1,..., k$, for all k. Thus, we have $\gamma_{jk} = \gamma_{(j-1)k}$ for all $j = 2,..., k$ which implies that $\Delta_k^{\max} \equiv 0$ for all k. Also $\sum_{k=1}^{\infty} \lambda_k = \infty$ implies $\gamma_{kk} \to 0$ as $k \to \infty$, and this completes the proof. ■

Corollary 4.2 If the convex combination weights are given by the average weighting rule (4.5) and the step-lengths are given by $a/(b + ck)$ for all k where a, $c > 0$ and $b \geq 0$, then the conditions (i), (ii), (iii) hold.

Proof It is evident from the fact $\gamma_{jk} = (b + cj)/ak$ for $j = 1,..., k$ for all k that $\gamma_{jk} \geq \gamma_{(j-1)k}$ for $j = 2,..., k$, $\gamma_{1k} \to 0$, and $\gamma_{kk} \to c/a > 0$ as $k \to \infty$. Since $\gamma_{jk} - \gamma_{(j-1)k} = c/ak$, we have $\Delta\gamma_k^{\max} = c/ak \to 0$ as $k \to \infty$. This completes the proof. ■

The following corollary presents another special step-length rule that satisfies the hypothesis of Theorem 4.1.

Corollary 4.3 Suppose that a step-length rule satisfying

$$\lambda_{k+1} \leq \lambda_k \quad \text{for all} \quad k \quad \text{and} \quad \lim_{k \to \infty} k\lambda_k = \infty \tag{4.13}$$

attains dual convergence, that is, $\pi_k \to \bar{\pi}$ for some $\bar{\pi} \geq \mathbf{0}$. If the convex combination weights are given by the average weighting rule, $\mu_j^k = 1/k$, for $j = 1,..., k$ for all k, then the assertions of Theorems 4.1 and 4.2 hold true. In particular, if the step-lengths are given by

$$\lambda_k = k^{-\mu}, \quad \text{where} \quad 0 < \mu < 1 \quad \text{for all} \quad k \tag{4.14}$$

then dual convergence is attained and the condition (4.13) holds.

Proof Note that $\gamma_{jk} = 1/(k\lambda_j)$ for $j = 1,... k$ for all k. Thus, the condition (i) holds since $\lambda_j \leq \lambda_{j-1}$. Now, we have

$$\gamma_{jk} - \gamma_{(j-1)k} = \frac{1}{k}\left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{j-1}}\right) = \frac{1}{k}\frac{\lambda_{j-1} - \lambda_j}{\lambda_j \lambda_{j-1}} \leq \frac{1}{k}\frac{\lambda_{j-1}}{\lambda_j \lambda_{j-1}} = \frac{1}{k\lambda_j}$$

which implies that

$$\Delta\gamma_k^{\max} \leq \max_{2 \leq j \leq k} \frac{1}{k\lambda_j} = \frac{1}{k\lambda_k} = \gamma_{kk}$$

Hence, $\gamma_{kk} \to 0$ and $\Delta\gamma_k^{\max} \to 0$ as $k \to \infty$, since $k\lambda_k \to \infty$ as $k \to \infty$, and also, $\gamma_{1k} = 1/(k\lambda_1) \to 0$ as $k \to \infty$. This proves the first part of the corollary. Now, if the step-lengths are given by (4.14), then convergence to a dual optimal solution is obtained since it satisfies (4.3). Moreover, it is easily verified that (4.13) holds true, and this completes the proof. ■

Remark 4.4 To be able to implement the rule in practice, embodied by Theorems 4.1 and 4.2, we should not have to store $\mathbf{x}_{\pi_k}$ for all k in order to find $\mathbf{x}_k$ via $\sum_{j=1}^{k} \mu_j^k \mathbf{x}_{\pi_j}$, as the convex combination weights μ_j^k change, from iteration to iteration. Hence, for example, if we have

$$\phi_k \equiv \frac{\mu_j^{k+1}}{\mu_j^k}$$

independent of j, then we have $\mu_j^{k+1} = \phi_k \mu_j^k$ and the primal iterates can be updated as follows:

$$\mathbf{x}_{k+1} \equiv \sum_{j=1}^{k+1} \mu_j^{k+1} \mathbf{x}_{\pi_j}$$

$$= \sum_{j=1}^{k} \mu_j^{k+1} \mathbf{x}_{\pi_j} + \mu_{k+1}^{k+1} \mathbf{x}_{\pi_{k+1}} = \phi_k \mathbf{x}_k + \mu_{k+1}^{k+1} \mathbf{x}_{\pi_{k+1}}$$

In particular, $\phi_k = (\sum_{j=1}^{k} \lambda_j)/(\sum_{j=1}^{k+1} \lambda_j)$ for Shor's convex combination weighting rule and $\phi_k = k/(k+1)$ for Larsson and Liu's average weighting rule. ■

Thus far, we have deduced primal feasibility when the chosen step-lengths attained dual convergence for some $\bar{\pi}$. In addition to this, a primal-dual convergence result is obtained when the primal iterates are generated after a sufficient number of dual iterations. Now, let us relax the dual convergence assumption and consider the following results.

Theorem 4.3 Suppose that the sequence $\{\pi_k\}$ generated by the subgradient method (4.1) is bounded, not necessarily having an optimal dual solution as even an accumulation point, and suppose further that condition (i) of Theorem 4.1 holds along with the condition that $\gamma_{kk} \to 0$ as $k \to \infty$. Then any accumulation point $\bar{\mathbf{x}}$ of the sequence $\{\mathbf{x}_k\}$ is feasible to Problem P.

Proof Note from (4.5) and the hypothesis of the theorem that $\mathbf{A}\bar{\mathbf{x}} - \mathbf{b} \leq \mathbf{0}$, and since $\bar{\mathbf{x}} \in X$, the proof is complete. ■

In fact, there are several direction finding strategies other than using pure subgradient directions, and several popular step-length rules that perform well computationally, but for which no primal convergence results are known. For example, the step length rule given by

$$\lambda_k = \beta_k \frac{w_k - \theta(\pi_k)}{\|\mathbf{g}_k\|^2} \tag{4.15}$$

where w_k is a target value and $0 < \varepsilon_1 \leq \beta_k \leq \varepsilon_2 < 2$, is the most popular one in practice (see Allen et al., 1987). When rule (4.15) is employed with the pure subgradient method using a fixed upper bound target value, that is $w_k \equiv w \geq \theta^*$ for all k, we have that there exists a subsequence of dual iterates that converges to a dual optimal solution. In this case, let K be the index set of the convergent subsequence such that $\lim_{k \in K,\ k \to \infty} \pi_k = \bar{\pi}$. Then, the subsequence $\{\mathbf{x}_k\}_K$ generated by (4.2) over the set K may not converge. However, for a subsequence $\{\mathbf{x}_k\}_{K_1} \subseteq \{\mathbf{x}_k\}$ indexed by a set $K_1 \subseteq K$ we could have primal convergence of Theorems 4.1 and 4.2 hold true, provided $\delta = 0$. Therefore, we can claim that if the rule (4.15) is used in conjunction with the subgradient

algorithm, then some, but perhaps not all, of the accumulation points of the sequence $\{\mathbf{x}_k\}$ are optimal solutions to the primal Problem P.

4.3 Primal Convergence for Dual Deflected Subgradient Methods.

Thus far, we have deduced some primal convergence results when the pure-subgradient methods were employed for solving the Lagrangian dual problems. It is well known that the pure-subgradient direction could generate zig-zagging phenomenon which leads slow tail-end convergence, and hence several subgradient deflection strategies have been proposed to overcome the weakness of the pure-subgradient methods. In this section, we present some primal convergence theorems for the case of using deflected subgradient methods. The deflected subgradient direction at iteration k, can be written as

$$\mathbf{d}_k = \mathbf{g}_k + \psi_k \mathbf{d}_{k-1}$$

where $\mathbf{g}_k$ is a subgradient evaluated at π_k, $\mathbf{d}_{k-1}$ is the previous direction of motion, and ψ_k is a deflection parameter.

For a moment, let us consider equality constrained problems, that is, the primal problem P is given as minimizing $\mathbf{cx}$ subject to $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \in X$, and consider the following result.

Theorem 4.4 Suppose that the deflection parameter ψ_k, and the convex combination weights μ_j^k have been selected to satisfy

$$\psi_j = \frac{\mu_{j-1}^k}{\mu_j^k} \quad \text{for all} \quad 1 \leq j \leq k \quad \text{for all} \quad k \tag{4.16}$$

Then, we have

$$\mathbf{A}\mathbf{x}_k - \mathbf{b} = \frac{\mu_k^k}{\lambda_k}(\pi_{k+1} - \pi_k) \quad \text{for all} \quad k \tag{4.17}$$

Proof For each k, since $\pi_{k+1} = \pi_k + \lambda_k \mathbf{d}_k$ and $\mathbf{d}_k = \mathbf{g}_k + \psi_k \mathbf{d}_{k-1}$, we have

$$\mathbf{A}\mathbf{x}_k - \mathbf{b} = \sum_{j=1}^{k}\left[\frac{1}{\lambda_j}(\pi_{j+1} - \pi_j) - \psi_j \frac{1}{\lambda_j}(\pi_j - \pi_{j-1})\right]$$

$$= \mu_1^k \frac{1}{\lambda_1}(\pi_2 - \pi_1) + \sum_{j=2}^{k}\mu_j^k\left[\frac{1}{\lambda_j}(\pi_{j+1} - \pi_j) - \psi_j \frac{1}{\lambda_j}(\pi_j - \pi_{j-1})\right]$$

$$= \left(\frac{\mu_2^k}{\lambda_1}\psi_2 - \frac{\mu_1^k}{\lambda_1}\right)\pi_1 + \sum_{j=2}^{k-1}\left(\frac{\mu_{j-1}^k}{\lambda_{j-1}} - \frac{\mu_j^k}{\lambda_j} - \frac{\mu_j^k}{\lambda_{j-1}}\psi_j + \frac{\mu_{j+1}^k}{\lambda_j}\psi_{j+1}\right)\pi_j$$

$$+ \left(\frac{\mu_{k-1}^k}{\lambda_{k-1}} + \frac{\mu_k^k}{\lambda_k} - \frac{\mu_k^k}{\lambda_{k-1}}\psi_k\right)\pi_k + \frac{\mu_k^k}{\lambda_k}\pi_{k+1} \tag{4.18}$$

From (4.16), we have

$$\frac{\mu_2^k}{\lambda_1}\psi_2 - \frac{\mu_1^k}{\lambda_1} = 0$$

and

$$\frac{\mu_{j-1}^k}{\lambda_j} - \frac{\mu_j^k}{\lambda_j} - \frac{\mu_j^k}{\lambda_{j-1}}\psi_j + \frac{\mu_{j+1}^k}{\lambda_j}\psi_{j+1} = 0 \quad \text{for all } j = 2,\ldots, k-1$$

Thus, the first and the second terms of (4.18) become zero. Also, since

$$\frac{\mu_{k-1}^k}{\lambda_{k-1}} + \frac{\mu_k^k}{\lambda_k} - \frac{\mu_k^k}{\lambda_{k-1}}\psi_k = -\frac{\mu_k^k}{\lambda_k}$$

we have

$$\mathbf{A}\mathbf{x}_k - \mathbf{b} = \frac{\mu_k^k}{\lambda_k}(\pi_{k+1} - \pi_k) \tag{4.19}$$

and this completes the proof. ■

The following remarks show that the requirement (4.16) is practically valid.

Remark 4.5 Suppose that $\mu_j^{k+1}/\mu_j^k = \phi_k$ is independent of j for all k and that

$$\frac{\mu_{j-1}^k}{\mu_j^k} = \frac{\mu_{j-1}^{k-1}}{\mu_j^{k-1}} = \cdots \qquad \text{for all } 1 \le j \le k, \quad \text{for all } k$$

Then, the choice of deflection parameter (4.16) holds for all $1 \le j \le k$ for all k. Therefore, Shor's and Larsson and Liu's rules are special cases of Theorem 4.4 above. In fact, since the step-lengths and the convex combination weights of the Shor's rule are given as (4.3) and (4.4), respectively, the deflection parameter becomes

$$\psi_k = \frac{\lambda_{k-1}}{\lambda_k}$$

Also, for the Larsson and Liu's rule, since $\mu_j^k = 1/k$, we have $\psi_k = 1$ for all k, and hence the direction of motion becomes $\mathbf{d}_k = \mathbf{g}_k + \mathbf{d}_{k-1}$.

Remark 4.6 At iteration k of some algorithmic procedure, assume that we have known that $\psi_1 = 0$ and $\psi_2, \ldots, \psi_{k-1}$. Then the convex combination weights μ_j^k and the deflection parameter ψ_k can be obtained as follows. First, choose $\mu_1^k \geq 0$, and find $\mu_j^k = \mu_{j-1}^k / \psi_j$ for $j = 2, \ldots, k-1$. Then, either (i) choose ψ_k according to some deflection rule and compute $\mu_k^k = \mu_{k-1}^k / \psi_k$, or (ii) select $\mu_k^k \geq 0$ independently and compute $\psi_k = \mu_{k-1}^k / \mu_k^k$. Note here that a normalization procedure is necessary to satisfy $\sum_{j=1}^{k} \mu_j^k = 1$.

Theorem 4.5 Suppose that ψ_k and μ_j^k are selected to satisfy (4.16). Then, either

(a) if $\|\mathbf{d}_k\| \leq M$ for some M for all k and λ_k is bounded from above, and if $\gamma_{kk} = \mu_k^k / \lambda_k \to 0$ as $k \to \infty$,

or

(b) if λ_k is chosen to have dual convergence, that is, $\pi_k \to \bar{\pi}$ as $k \to \infty$ for some $\bar{\pi}$, and if $\gamma_{kk} \to \delta \geq 0$ or $\gamma_{kk} \leq \delta$ for all k for some $\delta \geq 0$,

then any accumulation point $\bar{\mathbf{x}}$ is feasible to the primal problem. Moreover, in the latter case, $\bar{\mathbf{x}}$ and $\bar{\pi}$ are a pair of primal and dual optimal solutions.

Proof Since $\pi_{k+1} - \pi_k = \lambda_k \mathbf{d}_k$, we have from (4.19) that whenever (a) holds, $\mathbf{A}\bar{\mathbf{x}} - \mathbf{b} = \mathbf{0}$ by taking limits as $k \to \infty$ and $k \in K$, where $\bar{\mathbf{x}}$ is an accumulation point of $\{\mathbf{x}_k\}$ over the subset K. This shows the assertion holds for (a). Now, when (b) holds, the assertion is trivial from the fact that $\{\pi_{k+1} - \pi_k\} \to \mathbf{0}$ as $k \to \infty$. Moreover, in this case, since the complementary slackness condition automatically holds (equality constrained case),

by the proof of Theorem 4.2 and Remark 4.2, $\bar{\mathbf{x}}$ and $\bar{\pi}$ are a pair of priaml and dual optimal solutions. This completes the proof ■

Now, let us extend the above results to the inequality constrained cases. Suppose that the primal problem is given by

P Minimize $\mathbf{cx}$

subject to $\mathbf{Ax} \leq \mathbf{b}$

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

By introducing slack variables satisfying $\mathbf{Ax} + \mathbf{s} = \mathbf{b}$, $\mathbf{s} \geq \mathbf{0}$, we have whenever $\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$, that

$$\mathbf{s}_i = (\mathbf{b}_i - \mathbf{A}_i\mathbf{x}) < \mathbf{b}_i + \sum_{j : \mathbf{A}_{ij} < 0} |\mathbf{A}_{ij}| \mathbf{u}_j + 1 \equiv \mathbf{U}_i \tag{4.20}$$

for all $i = 1,..., m$, where $\mathbf{s}_i$, and $\mathbf{b}_i$, and $\mathbf{u}_i$ are the i—th components of the corresponding vectors, and $\mathbf{A}_i$ is i—th column of matrix $\mathbf{A}$ and $\mathbf{A}_{ij}$ is the j—th element of the vector $\mathbf{A}_i$. Therefore, we can construct the following equivalent problem.

EP Minimize $\mathbf{cx}$

subject to $\mathbf{Ax} + \mathbf{s} = \mathbf{b}$

$(\mathbf{x}, \mathbf{s}) \in Z = \{(\mathbf{x}, \mathbf{s}) : \mathbf{x} \in X, \mathbf{0} \leq \mathbf{s} \leq \mathbf{U}\}$

We now can apply the procedure of equality constrained case to EP. In this case, given a dual iterate π_k, the Lagrangian subproblem becomes

$$\underline{\text{ELS}(\pi_k)} \qquad \theta(\pi_k) = \text{minimum } \{\mathbf{cx} + \pi_k^t(\mathbf{Ax} - \mathbf{b}) : \mathbf{x} \in X\} + \sum_{i:(\pi_k)_i < 0} \pi_k \mathbf{U}_i \tag{4.21}$$

Let $\mathbf{x}_{\pi_k}$ be an optimal solution to ELS(π_k). Then, a subgradient can be obtained as

$$(\mathbf{g}_k)_i = \begin{cases} \mathbf{A}_i\mathbf{x}_{\pi_i} - \mathbf{b}_i, & \text{if } (\pi_k)_i \geq 0 \\ \mathbf{A}_i\mathbf{x}_{\pi_i} - \mathbf{b}_i + \mathbf{U}_i, & \text{if } (\pi_k)_i < 0 \end{cases} \tag{4.22}$$

Let us define a primal solution of EP as

$$\begin{pmatrix} \mathbf{x}_k \\ \mathbf{s}_k \end{pmatrix} = \sum_{j=1}^{k} \mu_j^k \begin{pmatrix} \mathbf{x}_{\pi_j} \\ \mathbf{s}_{\pi_j} \end{pmatrix}$$

where $\mathbf{s}_{\pi_j}$ is given by

$$(\mathbf{s}_{\pi_j})_i = \begin{cases} 0 & \text{if } (\pi_j)_i \geq 0 \\ \mathbf{U}_i & \text{if } (\pi_j)_i < 0 \end{cases}$$

Then, the previous convergence result holds for EP. Also, note that if $(\bar{\mathbf{x}}, \bar{\mathbf{s}})$ is feasible to EP, then $\bar{\mathbf{s}} < \mathbf{U}$ by virtue of (4.20).

Theorem 4.6 Suppose that for the extended problem EP, we have dual convergence, that is, $\{\pi_k\} \to \bar{\pi}$ as $k \to \infty$ for some $\bar{\pi}$ such that $(\bar{\mathbf{x}}, \bar{\mathbf{s}})$ solves

$$\text{minimize}\{\mathbf{cx} + \bar{\pi}^t(\mathbf{Ax} + \mathbf{s} - \mathbf{b}) : (\mathbf{x}, \mathbf{s}) \in Z\} \tag{4.23}$$

Then, $(\bar{\mathbf{x}}, \bar{\pi})$ is a primal-dual optimal solution of P.

Proof Since $(\bar{\mathbf{x}}, \bar{\mathbf{s}})$ is feasible, we have $\bar{\mathbf{s}} < \mathbf{U}$, and so, we must have $\bar{\pi} \geq \mathbf{0}$, since $(\bar{\mathbf{x}}, \bar{\mathbf{s}})$ solves (4.23). Moreover, if $(\bar{\pi})_i > 0$, then the corresponding $(\bar{\mathbf{s}})_i = 0$. Hence, $\bar{\mathbf{x}}$ solves for minimum$\{\mathbf{cx} + \bar{\pi}^t(\mathbf{Ax} - \mathbf{b}) : \mathbf{x} \in X\}$, and we have $\{\mathbf{A}\bar{\mathbf{x}} \leq \mathbf{b}, \bar{\mathbf{x}} \in X\}$ which means $\mathbf{x}$ is feasible to P. Moreover, we have complementary slackness condition, $(\bar{\pi})_i(\mathbf{A}_i\bar{\mathbf{x}} - \mathbf{b}_i) = 0$ for all i, that is, $(\bar{\pi})_i = 0$, or $(\bar{\pi})_i > 0$ and $(\bar{\mathbf{s}})_i = 0$. Consequently, $(\bar{\mathbf{x}}, \bar{\pi})$ is a primal-dual optimal solution to P and this completes the proof. ■

Note that for the inequality constrained case, the perturbation of subgradient as in (4.23) over the subgradients of the pure equality constrained case, along with the relaxation of nonnegative of π, does the main role of the above theorem.

5. A Hybrid Primal-Dual Subgradient Algorithm

5.1 Introduction

In this chapter, we present a hybrid primal-dual subgradient algorithm (HPDSA) that uses a certain combination of subgradient deflection strategies in conjunction with the variable target value method presented in Chapter 2 to solve the Lagrangian dual problem. Along with this, it adopts the primal convergence theorems of Chapter 4 to recover a primal solution using 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 near feasibility and near optimality.

To present this scheme, as before, let us assume that a primal problem is given as

P Minimize $\mathbf{cx}$

subject to $\mathbf{Ax} \leq \mathbf{b}$

$\mathbf{x} \in X$

where **A** is an $m \times n$ matrix and X is a nonempty polytope in E_n. Here, we assume that X has some special structure that can be advantageously exploited, so that linear programming problems over X, as well as of computing projections onto X are relatively easy to solve. We also assume that the problem has been scaled by dividing each row by its infinite norm ($\|\cdot\|_\infty$). The Lagrangian dual for Problem P can be written as follows:

LD Maximize $\theta(\pi)$

subject to $\pi \geq \mathbf{0}$

where $\theta(\pi)$ is evaluated via the Lagrangian subproblem

LS(π) $\theta(\pi) = \text{minimum } \{\mathbf{cx} + \pi^t(\mathbf{Ax} - \mathbf{b}) : \mathbf{x} \in X\}$

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 a given dual solution, $\pi \geq \mathbf{0}$, let us define

$$X_\pi = \{\mathbf{x}_\pi : \mathbf{x}_\pi \text{ evaluates } \theta(\pi)\} \subseteq X$$

as the set of optimal solutions for the subproblem LS(π). Then the subdifferential $\partial\theta(\pi)$ of θ at π is given by $\{\mathbf{Ax}_\pi - \mathbf{b} : \mathbf{x}_\pi \in X_\pi\}$.

Several methods have been proposed for solving LD such as cutting plane methods, (steepest) ascent methods, subgradient based methods, and heuristic methods (for example, see Fisher, 1981, and Bazaraa, Sherali, and Shetty, 1993). The cutting plane method, which was originally proposed by Kelly (1960), is a row generation algorithm that is equivalent, via duality, to the column generation algorithm of Wolfe (see

Dantzig, 1963), and also to the decomposition method of Dantzig and Wolfe (1961). Moreover, it can be interpreted as a tangential approximation scheme since at each iteration of the algorithm, it generates a new cut that provides an improved piecewise linear concave upper approximation for the objective function θ of LD. Therefore, this method is also called an outer linearization method (see Bazaraa and Goode, 1979, Parker and Rardin, 1988, and Bazaraa et al., 1993). Even if this method generates better approximations at each iteration, the corresponding sequence of objective function values may not be monotonically increasing. On the other hand, the ascent methods find an ascent direction via suitable auxiliary programs and then perform a one-dimensional line search along the obtained direction to find an improved iterate. Bazaraa and Goode (1979) present several variants for finding the (steepest) ascent direction using different normalization constraints. Another class of ascent methods that can be employed for solving Lagrangian dual problems are bundle type methods (see Kiwiel, 1985). While these are theoretically convergent, they are only suitable for solving relatively smaller sized problems.

In contrast to the foregoing approaches, the procedures that are based on a single subgradient evaluation at each iteration are very simple and easy to implement, although they are not strictly ascent methods. Not only for Lagrangian dual problems, but also for general nondifferentiable optimization problems, these subgradient methods usually perform fairly well in practice. Held and Karp (1970, 1971) and Held, Wolfe, and Crowder (1974) present successful implementations of subgradient methods for solving Lagrangian duals of discrete problems. Goffin (1977) and Shor (1985) present the convergence behavior of these subgradient methods. However, a fine tuning of parameter values for these methods is usually required, and even then the

methods can sometimes stall at an objective value far below the optimal value (see Sherali and Myers, 1986).

In related approaches, several heuristic multiplier adjustment techniques have been proposed by Erlenkotter (1978), Fisher, Jaikumar, and Van Wassenhove (1986), and Fisher and Kedia (1990). However, each of these methods are designed for specific problems by exploiting their special structures, and hence are not applicable to general problems. On the other hand, as mentioned earlier, the simplex method can be used for solving LD problems in conjunction with the column generation technique (see Fisher, 1981, and Fisher et al., 1975). Marsten (1975) and Marsten, Hogan, and Blankenship (1975) present the Boxstep method that can be used in the context of the outer linearization method or the Dantzig-Wolfe decomposition method. It has also been suggested that the subgradient methods and the Boxstep method might complement each other, and that the subgradient methods could provide a good starting solution for the Boxstep method. In this way, it has been conjectured that a combination of these two methods might perform well in practice (see Marsten et al., 1975, and Fisher, 1981). However, this strategy is yet to be tested for general LD problems.

The new hybrid algorithm proposed in this chapter consists of three stages. First, the algorithm attempts to solve the Lagrangian dual problem using a combined strategy of three different subgradient deflection methods, namely, ODSA, ADS, and MGT (see Chapter 2 for details of these algorithm) in conjunction with the variable target value method, VTVM, that is presented also in Chapter 2. After a certain number of iterations, or when some other stopping criteria is triggered, such as a maximum allowable limit on the number of consecutive target increases, the algorithm turns to its second stage. In the second stage, the algorithm not only continues to perform the

dual procedure as in the first stage, but also generates a sequence of updated primal solutions using some convex combinations of the Lagrangian subproblem solutions. At the end of this stage, the algorithm evaluates the extent of feasibility and optimality of the incumbent primal solution using the available dual objective function value. If necessary, the algorithm then applies a penalty function method to the primal problem as in Sen and Sherali (1986) and Sherali and Ulular (1989), to further improve the primal solution toward a near feasible and optimal solution.

This chapter is organized as follows. In Section 2, we provide several subgradient deflection strategies, and present a combined direction finding rule for the first stage in conjunction with the variable target value method. A primal update scheme along with the dual procedure is stated in Section 3. Following this, a penalty function method for refining the primal solution is presented in Section 4. Finally, practical implementation guidelines and some results of computational experiments are reported in Section 5.

5.2 Stage I: Subgradient Deflection Strategies for Dual Optimization

Ever since Held et al.'s (1974) seminal paper, several modifications of the subgradient method have been proposed. It has been amply demonstrated that no matter what step-length rule is implemented, the pure subgradient method has a limit on its

computational performance, simply due to the direction of motion used. As with the steepest ascent direction for the differentiable case, the subgradient direction for the nondifferentiable case also results in a zig-zagging phenomenon that might manifest itself at any stage of the subgradient algorithm, causing 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 methods for the differentiable case. A conjugate gradient direction for the differentiable case is computed by combining the current gradient with the previous direction, while forcing some conjugacy requirement. However, for the nondifferentiable case, this concept is used only as a strategy to deflect the subgradient, since conjugacy may no longer be meaningful.

Accordingly, the direction of motion $\mathbf{d}_k$ at π_k is computed as

$$\mathbf{d}_k = \mathbf{g}_k + \psi_k \mathbf{d}_{k-1} \tag{5.1}$$

where ψ_k is a deflection parameter, $\mathbf{g}_k$ is a subgradient of θ at π_k, and where $\mathbf{d}_{k-1}$ is the previous direction with $\mathbf{d}_0 \equiv \mathbf{0}$. Note that a subgradient of θ at π_k can be obtained by solving the Lagrangian subproblem, that is, $\mathbf{g}_k = \mathbf{A}\mathbf{x}_{\pi_k} - \mathbf{b}$ where $\mathbf{x}_{\pi_k} \in X_{\pi_k}$. The new iterate is then computed for each component i as

$$(\pi_{k+1})_i = \text{ maximum } \{(\pi_k + \lambda_k \mathbf{d}_k)_i,\ 0\} \tag{5.2}$$

where λ_k is a suitable step-length.

There are two particular subgradient deflection algorithms of the type (5.2), that have been well tested in the literature, namely, 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 known to be superior to the pure subgradient method in practice, when used with a specially designed step-length selection rule. Moreover, in Chapter 2, we have proposed a promising new strategy called the Optimally Deflected Subgradient Algorithm (ODSA). Even though each method has its own advantages for some specially structured problems, we have seen that no single strategy clearly dominates (computationally) the others for general problems. Therefore, in the proposed algorithm, we adopt a combination of all these three strategies that have proven to perform well in different contexts, by appropriately shifting from one to another in turn, depending on the algorithmic progress. This approach is called Subroutine DFS, and is formally stated as follows. Note here that this subroutine is assumed to be used in conjunction with the variable target value method VTVM that will be addressed again later.

Direction Finding Strategy (DFS) Subroutine At an outer loop ℓ and inner loop k of VTVM, the direction of motion is found as $\mathbf{d}_k = \mathbf{g}_k + \psi_k \mathbf{d}_{k-1}$, where ψ_k is computed as follows.

(i) ODSA If $\ell \equiv 1 \pmod 3$, find

$$r_k = w_\ell - \theta_k \quad \text{and}$$

$$s_k = \begin{cases} 0, & \text{if } k = 1 \\ \text{maximum } \{r_{k-j} + \psi_{k-j} s_{k-j} - \mathbf{d}_{k-j}^t (\pi_k - \pi_{k-j}),\ 0\} & \text{if } k \geq 2 \end{cases}$$

where $j \geq 1$ is the smallest integer such that ψ_{k-j} is finite. Also, find

$$M_k = \text{maximum} \left\{ \frac{r_k}{\|\mathbf{g}_k\|},\ \frac{s_k}{\|\mathbf{d}_{k-1}\|},\ \overline{\phi} \right\}$$

$$\bar{\phi} = \frac{r_k + s_k\bar{\psi}}{\| -\mathbf{g}_k + \bar{\psi}\mathbf{d}_{k-1}\|}, \quad \text{and} \quad \bar{\psi} = \frac{(\mathbf{g}_k^t\mathbf{d}_{k-1})r_k + \|\mathbf{g}_k\|^2 s_k}{(\mathbf{g}_k^t\mathbf{d}_{k-1})s_k + \|\mathbf{d}_{k-1}\|^2 r_k}$$

If $M_k = s_k/\|\mathbf{d}_{k-1}\|$, let $\mathbf{d}_k = \mathbf{d}_{k-1}$ and put $\psi_k = \infty$. Otherwise, let $\mathbf{d}_k = \mathbf{g}_k + \psi_k\mathbf{d}_{k-1}$, where

$$\psi_k = \begin{cases} \bar{\psi}, & \text{if } M_k = \bar{\phi} \\ 0 & \text{if } M_k = r_k/\|\mathbf{g}_k\| > \bar{\phi} \end{cases}$$

(i) ADS If $l \equiv 2 \pmod 3$, compute

$$\psi_k = \frac{\|\mathbf{g}_k\|}{\|\mathbf{d}_{k-1}\|}$$

(ii) MGT If $l \equiv 0 \pmod 3$, compute

$$\psi_k = \begin{cases} -1.5\dfrac{\mathbf{g}_k^t\mathbf{d}_{k-1}}{\|\mathbf{d}_{k-1}\|^2}, & \text{if } \mathbf{g}_k^t\mathbf{d}_{k-1} < 0 \\ 0, & \text{if } \mathbf{g}_k^t\mathbf{d}_{k-1} \geq 0 \end{cases}$$

Step-Length Prescription

Now, let us consider step-length rules. There are several research studies that conduct computational experiments on using different step-length rules, and conjecture some superiority of one rule over another in practice. But no direct comparisons and *no* practical guidelines for selecting parameters have been made for general Lagrangian dual problems. Only recommendations for some specially structured

problems have been proposed without any guarantee of superiority (see Polyak, 1969, and Held et al., 1974). However, the step-length rule given by

$$\lambda_k = \beta_k \frac{w - \theta(\pi_k)}{\|\mathbf{d}_k\|^2} \tag{5.3}$$

where w is a target value and β_k is some positive value, is known to be quite promising in practice. Bazaraa and Sherali (1981) present some rules to choose the target value as a convex combination of a fixed upper bound and the current value. Kim, Ahn, and Cho (1991) use a similar idea and present a variable target value method. However, in these two works, some bounds on the objective function value have been used initially.

In the proposed algorithm, we adopt one of two strategies. If a good upper bound on the Lagrangian dual is available a priori, the user can specify that this bound be used as a fixed target value w, and that the Block Halving strategy of Sherali and Ulular (1989) be adopted to control the parameter β_k in (5.3). On the other hand, if this is not the case, then we adopt the variable target value method (VTVM) discussed in Chapter 2, which not only performs well in comparison with any other step-length rule when used in conjunction with most direction finding strategies, but also admits convergence without having any knowledge of optimality. The convergence properties of Subroutine VTVM via deflected subgradient methods have been discussed in Chapter 2. The following is a brief synopsis of Subroutine VTVM.

Subroutine VTVM

Iteration Input

At an outer iteration ℓ and inner iteration k, we have the current solution π_k, its objective function value $\theta(\pi_k)$, a subgradient $\mathbf{g}_k$ of θ at π_k, the current target value w_ℓ, and the best incumbent solution $\bar{\pi}$ with objective value $\bar{z}$.

Iteration Output

The algorithm first tests whether a new target value is needed or not. If the incumbent dual function value has reached the target (or lies in a neighborhood of it), the algorithm computes a new target value by increasing the target value. On the other hand, if the algorithm fails to improve the incumbent value for some consecutive iterations (this number is dynamically changed for each outer loop), the target is decreased. For both these cases, the direction of motion is taken as $\mathbf{d}_k = \mathbf{g}_k$. Otherwise, if the target value is not updated, then the algorithm calls Subroutine DFS to find a deflected direction of motion $\mathbf{d}_k$. Finally, a suitable step-length λ_k is computed using (5.3). Also, the subroutine stores the best incumbent solution $\bar{\pi}$ and the incumbent function value $\bar{z}$.

The algorithmic details of Subroutine VTVM have been given in Chapter 2, and the recommended parameter values will be addressed later. Below, we summarize the default strategy used for Stage I based on Algorithm VTVM. As mentioned above, if a fixed target value is being used, then VTVM is assumed to be modified to skip any target value update steps.

Algorithm for Stage I (Dual Optimization)

Initialization Select the necessary parameter values for Subroutine VTVM, the maximum number of iterations N_1, and a termination parameter $\varepsilon_1 > 0$. Determine a starting solution π_0, evaluate its dual objective function value $\theta(\pi_0)$, and a subgradient $\mathbf{g}_0$ of θ at π_0. Initialize the best known dual solution as $\bar{\pi} = \pi_0$, having a function value of $\bar{z} = \theta(\pi_0)$. Let the direction of motion $\mathbf{d}_0 = \mathbf{g}_0$, and compute the initial target value w_0. Set the outer loop counter $\ell = 0$ and the inner loop counter $k = 0$.

Step 1. Call Subroutine VTVM (this procedure uses Subroutine DFS, if necessary as described above) to find a direction of motion $\mathbf{d}_k$ and a step-length λ_k.

Step 2. Update the dual solution π_{k+1} using (5.2), evaluate the dual objective function value $\theta(\pi_{k+1})$ and a subgradient $\mathbf{g}_{k+1}$ of θ at π_{k+1}. If $\|\mathbf{g}_{k+1}\| < \varepsilon_1$, terminate the algorithm with a near optimal solution π_{k+1}, and proceed to Stage III with $\bar{\mathbf{x}} \equiv \mathbf{x}_{\pi_{k+1}}$. If $k + 1 = N_1$, then terminate the algorithm with the incumbent solution $\bar{\pi}$ having an objective value $\bar{z} = \theta(\bar{\pi})$, and proceed to Stage II. Otherwise, increment k by one and return to Step 1.

5.3 Stage II: Recovery of Primal Solutions

In this section, we present a procedure for recovering primal solutions via the dual subgradient-based method. Note that the dual subgradient-based methods do not necessarily provide a primal optimal or even feasible solution when solving the dual subproblems, even at a dual optimal solution. This poses a problem of finding a primal optimal solution when subgradient methods are used for solving the Lagrangian dual problem. In Chapter 4, we addressed this issue by proposing certain generic primal convergence theorems that enable us to recover a primal optimal solution via a suitable convex combination of the dual subproblem solutions. Hence, for the second stage of the proposed algorithm, we begin to recover primal solutions while continuing to update the dual solutions. Note that we can continue to use the same strategy that was designed for Stage I, or we might choose to adopt a strategy that is particularly geared toward generating better primal solutions. Accordingly, in this stage, we adopt the pure-subgradient method along with an average weighting primal recovery rule which seems to be the best in our experience. Furthermore, in order to enhance the quality of individual subproblem solutions (with respect to feasibility to the original problem) as well as to consequently improve the subgradient generated thereby (in the sense of having a smaller Euclidean norm), we might wish to alter the Lagrangian dual subproblems used in this stage. For example, we can add some bound constraints or other implied/valid constraints to the subproblem for this purpose, at the expense of increasing the effort required to solve each subproblem, but hopefully, to gain the advantage of generating improved primal solutions and subgradient based directions. Noting that Stage II is to be run for relatively fewer iterations than used for Stage I, this can be implemented without any undue increase

in the computational burden. Below, we present an algorithmic statement for Stage II.

Algorithm for Stage II (Dual Optimization and Primal Recovering)

Initialization Select the maximum allowed iteration limit N_2 along with termination parameters $\varepsilon_2 > 0$ and $\varepsilon_3 > 0$. (Note that the termination parameter $\varepsilon_1 > 0$ is selected in Stage I.) Initialize the dual solution at $\pi_1 = \bar{\pi}$, the primal solution at $\mathbf{x}_1 = \mathbf{x}_{\bar{\pi}}$, and let the direction of motion be given by $\mathbf{g}_1 = \bar{\mathbf{g}}$. (Note that $\bar{\pi}$, having a dual objective function value $\bar{z}$, the subgradient $\bar{\mathbf{g}}$ evaluated at $\bar{\pi}$, and the corresponding subproblem optimal solution $\mathbf{x}_{\bar{\pi}}$ have been obtained during Stage I.) Set the iteration counter $k = 1$.

Step 1 (Dual Optimization) Select an appropriate step-length λ_k and find a new iterate

$$(\pi_{k+1})_i = \text{maximum}\{(\pi_k + \lambda_k \mathbf{g}_k)_i,\ 0\} \quad \text{for each component } i$$

Step 2 Evaluate $\theta(\pi_{k+1})$, find a subproblem optimal solution $\mathbf{x}_{\pi_{k+1}}$, and find a subgradient $\mathbf{g}_{k+1}$ of θ at π_{k+1}. If $\theta(\pi_{k+1}) < \bar{z}$, then set $\bar{\pi} = \pi_{k+1}$ and $\bar{z} = \theta(\pi_{k+1})$.

Step 3 (Primal Update) Find a new primal solution using

$$\mathbf{x}_{k+1} = \frac{k}{(k+1)}\,\mathbf{x}_k + \frac{1}{(k+1)}\,\mathbf{x}_{\pi_{k+1}}$$

(Note that this is an average of all the previous optimal solutions to the subproblems LS(π_j), where $1 \leq j \leq k$, that is,

$$\mathbf{x}_{k+1} \equiv \frac{1}{(k+1)} \sum_{j=1}^{k+1} \mathbf{x}_{\pi_j}$$

where $\mathbf{x}_{\pi_j}$ is the optimal solution to the subproblem LS(π_j).)

Step 4 (Termination Check) Increment k by one. If $\|\mathbf{g}_k\| > \varepsilon_1$ and $k \leq N_2$, then return to Step 1. Otherwise, set $\bar{\mathbf{x}} = \mathbf{x}_{k+1}$ and compute the relative duality gap

$$RGAP = \frac{|\mathbf{c}^t\bar{\mathbf{x}} - \bar{z}|}{|\bar{z}|}$$

and the feasibility ratio RFEAS given by the average violation over the violated constraints, if any. (Note that we assume here that the dualized constraints have been pre-scaled as mentioned earlier.) If $RGAP \leq \varepsilon_2$ and $RFEAS \leq \varepsilon_3$, then terminate the overall procedure with the near optimal primal-dual solutions $(\bar{\mathbf{x}}, \bar{\pi})$. Otherwise, proceed to Stage III with the incumbent primal-dual solutions $\bar{\mathbf{x}}$ and $\bar{\pi}$.

5.4 Stage III: Primal Penalty Function Method

Note that our primal convergence theorems guarantee only that every accumulation point of the sequence of the primal iterates $\{\mathbf{x}_k\}$ generated at Stage II is an optimal solution to the primal problem. Hence, a primal solution obtained at some maximum allowed iteration limit N_2 could be neither optimal nor even feasible to the primal problem. Therefore, in addition to the previous primal recovery scheme, we might

need to further polish the primal solution to achieve near feasibility and optimality. Toward this end, we adopt the penalty function method of Sen and Sherali (1986) and Sherali and Ulular (1989) at the final (third) stage of the proposed algorithm.

For a given dual solution $\bar{\pi} \geq \mathbf{0}$ that is obtained throughout the previous dual update, let us define the penalty function

$$h(\mathbf{x}) = \mathbf{c}^t\mathbf{x} + \sum_{i=1}^{m} (\bar{\pi}_i + \omega) \max\{0,\ (\mathbf{a}_i^t\mathbf{x} - b_i)\} \tag{5.4}$$

where $\bar{\pi}_i$ denotes the i-th component of the vector $\bar{\pi}$ and $\omega > 0$ is a penalty parameter, and $\mathbf{a}_i$ and b_i are, respectively, the i-th row of the matrix $\mathbf{A}$ and i-th component of b. Note that if the original problem includes equality constraints, then in (5.4), $\bar{\pi}_i$ is replaced by $|\bar{\pi}_i|$ and the $\max\{0, \cdot\}$ operation is also replaced by the absolute constraint function value. Also, we remark here that Sherali and Ulular (1989) have proposed a wide range of Augmented Lagrangian types of penalty functions that can be used in lieu of (5.4). However, since we have an advanced dual solution, then by the theory of ℓ_1-penalty functions (see Bazaraa, et al., 1993), (5.4) appears to be attractive in our context. Moreover, it satisfies the generic penalty function properties required by the convergence results of Sherali and Ulular. In particular, $h(\mathbf{x})$ is a convex function of $\mathbf{x}$ having bounded subgradients for any $\mathbf{x} \in X$, and for any $\mathbf{x}_k \in X$ at some iteration k of an algorithmic process, we have

$$h(\mathbf{x}_k) \geq \mathbf{c}^t\mathbf{x}_k + \sum_{i=1}^{m} (\bar{\pi})_i(\mathbf{a}_i\mathbf{x}_k - b_i) \geq \theta(\bar{\pi})$$

where $(\bar{\pi})_i$ is the i-th component of the dual solution $\bar{\pi}$. Also, for any primal optimal solution $\mathbf{x}^*$, we have

$$h^* \equiv h(\mathbf{x}^*) = \mathbf{c}^t\mathbf{x}^* \equiv f^*$$

Moreover, we have the following conditions that are similar to the theorems in Sen and Sherali (1986) and Sherali and Ulular (1989) for terminating the algorithm with a near-feasible and near-optimal primal solution. (An identical statement holds analogously for equality constraints.)

Proposition 5.1 Suppose that the penalty parameter ω is selected as $\omega = 2M + \Delta$, where $M = \text{maximum}\{(\bar{\pi})_i; i = 1,..., m\}$ and Δ is some constant. If $\theta(\bar{\pi}) \geq h(\mathbf{x}_k) - \varepsilon$, for some $\mathbf{x}_k \in X$ and for some $\varepsilon > 0$, then we have

$$\mathbf{a}_j\mathbf{x}_k - b_j \leq \frac{\varepsilon}{M} \quad \text{for all } 1 \leq j \leq m \quad \text{and} \quad |\bar{\pi}^t(\mathbf{A}\mathbf{x}_k - \mathbf{b})| \leq m\varepsilon$$

Proof By assumption, we have,

$$\mathbf{c}^t\mathbf{x}_k + \bar{\pi}^t(\mathbf{A}\mathbf{x}_k - \mathbf{b}) \geq \theta(\bar{\pi}) \geq h(\mathbf{x}_k) - \varepsilon$$

Thus,

$$\bar{\pi}^t(\mathbf{A}\mathbf{x}_k - \mathbf{b}) \geq \sum_{i=1}^{m} [\bar{\pi}_i + \omega] \max\{0, \mathbf{a}_i\mathbf{x}_k - b_i\} - \varepsilon \tag{5.5}$$

$$= \sum_{l=1,\ l \neq j}^{m} [\bar{\pi}_l + \omega] \max\{0, \mathbf{a}_l\mathbf{x}_k - b_l\} + [\bar{\pi}_j + \omega] \max\{0, \mathbf{a}_j\mathbf{x}_k - b_j\} - \varepsilon$$

$$\geq \sum_{i=1,\ i\neq j}^{m} \bar{\pi}_i(\mathbf{a}_i\mathbf{x}_k - b_i) + \omega \max\{0, \mathbf{a}_j\mathbf{x}_k - b_j\} - \varepsilon$$

Hence, for any j, $1 \leq j \leq m$, we have

$$\bar{\pi}_j(\mathbf{a}_j\mathbf{x}_k - b_j) \geq \omega \max\{0,\ \mathbf{a}_j\mathbf{x}_k - b_j\} - \varepsilon \tag{5.6}$$

If $\mathbf{a}_j\mathbf{x}_k - b_j \geq 0$, then (5.6) becomes $\bar{\pi}_j(\mathbf{a}_j\mathbf{x}_k - b_j) \geq \omega(\mathbf{a}_j\mathbf{x}_k - b_j) - \varepsilon$ that leads to $(\mathbf{a}_j\mathbf{x}_k - b_j)(\omega - \bar{\pi}_j) \leq \varepsilon$. Hence, we have

$$\mathbf{a}_j\mathbf{x}_k - b_j \leq \frac{\varepsilon}{M} \tag{5.7}$$

Now, note that

$$\bar{\pi}^t(\mathbf{A}\mathbf{x}_k - \mathbf{b}) = \sum_{i=1}^{m} \bar{\pi}_i(\mathbf{a}_i\mathbf{x}_k - b_i) \leq \sum_{i=1}^{m} \bar{\pi}_i \frac{\varepsilon}{M} \leq m\varepsilon$$

From (5.5), we have, $\bar{\pi}^t(\mathbf{A}\mathbf{x}_k - \mathbf{b}) \geq -\varepsilon \geq -m\varepsilon$, (note that the equality holds for $m = 1$) and hence, we have

$$|\bar{\pi}^t(\mathbf{A}\mathbf{x}_k - \mathbf{b})| \leq m\varepsilon$$

and this completes the proof. ■

Now, based on the foregoing constricts, let us formally state the algorithm.

Algorithm for Stage III (Primal Penalty Function Method)

Initialization Choose a maximum allowed iteration number N_3, and the penalty parameter ω. Set the starting solution $\mathbf{x}_0 = \bar{\mathbf{x}}$ (as obtained via the previous stages). Compute the penalty function value $h(\mathbf{x}_0)$ and a subgradient $\mathbf{q}_0$ of h at $\mathbf{x}_0$. Set the iteration counter $k = 0$. (Note that the incumbent solution $\bar{\pi}$ and the dual function value $\bar{z}$ were determined through the previous stages, and also that the termination tolerances ε_1, ε_2, and ε_3 have been previously defined.)

Step 1 Choose a step-length parameter β_k, (we suggest the Block Halving strategy of Sherali and Ulular, 1989), and compute the step-length

$$\lambda_k = \beta_k \frac{h(\mathbf{x}_k) - \bar{z}}{\|\mathbf{q}_k\|^2} \tag{5.8}$$

Step 2 Find a direction of motion $\mathbf{d}_k$ using Subroutine DFS of Section 5.2 (revised for a minimization problem), and compute the new iterate $\mathbf{x}_{k+1} = P_X(\mathbf{x}_k + \lambda_k \mathbf{d}_k)$. Evaluate the function value $h(\mathbf{x}_{k+1})$, and find a subgradient $\mathbf{q}_{k+1}$ of h at $\mathbf{x}_{k+1}$.

Step 3 Compute the relative duality gap

$$\text{RGAP} = \frac{h(\mathbf{x}_{k+1}) - \bar{z}}{|\bar{z}|} \tag{5.9}$$

If either $\|\mathbf{q}_k\| \leq \varepsilon_1$ or $RGAP \leq \varepsilon_2$ then terminate the algorithm with a near-optimal/feasible solution $\mathbf{x}_{k+1}$. Otherwise, increment k by one and return to Step 1.

Consider the following convergence theorem relates to the above procedure.

Theorem 5.1 Suppose that the dual solution $\bar{\pi}$ satisfies $\bar{z} \geq h(\mathbf{x}^*) - \tilde{\varepsilon}$ for some $\tilde{\varepsilon} > 0$, where $\mathbf{x}^*$ is an optimal primal solution. Furthermore, if the step-length parameter β_k is chosen as

$$\beta_k = 1 - \frac{\tilde{\varepsilon}}{h(\mathbf{x}_k) - \bar{z}}$$

for all k, and assume that $\mathbf{d}_k = -\mathbf{q}_k$ for all k. Then, the pure-subgradient algorithm either finds a solution $\mathbf{x}_k \in X$ such that $h(\mathbf{x}_k) \leq \bar{z} + \tilde{\varepsilon}$ for some k, or generates a sequence $\{\mathbf{x}_k\}$ with $h(\mathbf{x}_k) > \bar{z} + \tilde{\varepsilon}$ for all k such that $\{h(\mathbf{x}_k)\} \to \bar{z} + \tilde{\varepsilon}$ as $k \to \infty$

Proof Suppose that the algorithm generates an infinite sequence $\{h(\mathbf{x}_k)\}$ satisfying $h(\mathbf{x}_k) > \bar{z} + \tilde{\varepsilon}$ and let $\mathbf{x}^*$ be a primal optimal solution. Then,

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 = \|P_X(\mathbf{x}_k - \lambda_k \mathbf{q}_k) - \mathbf{x}^*\|^2$$

$$\leq \|\mathbf{x}_k - \lambda_k \mathbf{q}_k - \mathbf{x}^*\|^2$$

$$= \|\mathbf{x}_k - \mathbf{x}^*\|^2 + \lambda_k^2 \|\mathbf{q}_k\|^2 - 2\lambda_k \mathbf{q}_k^t (\mathbf{x}_k - \mathbf{x}^*)$$

Since $h(\mathbf{x}^*) \geq h(\mathbf{x}_k) + \mathbf{q}_k^t(\mathbf{x}^* - \mathbf{x}_k)$, we have $\mathbf{q}_k^t(\mathbf{x}_k - \mathbf{x}^*) \geq h(\mathbf{x}_k) - h(\mathbf{x}^*)$ and hence,

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 \leq \|\mathbf{x}_k - \mathbf{x}^*\|^2 + \lambda_k^2 \|\mathbf{q}_k\|^2 - 2\lambda_k [h(\mathbf{x}_k) - h(\mathbf{x}^*)]$$

From the selection of β_k and (5.8), we have

$$\lambda_k = \frac{h(\mathbf{x}_k) - \bar{z} - \tilde{\varepsilon}}{\|\mathbf{q}_k\|^2}$$

and hence,

$$\|\mathbf{x}_k - \mathbf{x}^*\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 \geq 2\lambda_k[h(\mathbf{x}_k) - h(\mathbf{x}^*)] - \lambda_k^2\|\mathbf{q}_k\|^2$$

$$= 2\frac{[h(\mathbf{x}_k) - \bar{z} - \tilde{\varepsilon}][h(\mathbf{x}_k) - h(\mathbf{x}^*)]}{\|\mathbf{q}_k\|^2} - \frac{[h(\mathbf{x}_k) - \bar{z} - \tilde{\varepsilon}]^2}{\|\mathbf{q}_k\|^2}$$

Since $\bar{z} + \tilde{\varepsilon} \geq h(\mathbf{x}^*)$, we have

$$\|\mathbf{x}_k - \mathbf{x}^*\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 \geq \frac{1}{\|\mathbf{q}_k\|^2}[h(\mathbf{x}_k) - \bar{z} - \tilde{\varepsilon}]^2 > 0$$

Therefore, $\{\|\mathbf{x}_k - \mathbf{x}^*\|\}$ is a monotone decreasing, and hence a convergent sequence. Moreover, since $\|\mathbf{q}_k\| \leq M_1$ for all k for some $M_1 > 0$ by the property of h, we have $\{h(\mathbf{x}_k)\} \to \bar{z} + \tilde{\varepsilon}$, and this completes the proof. ■

5.5 *Implementation and Computational Experience*

For computational test runs, we have attempted to solve some transportation problems without exploiting any special structures in the Lagrangian dual scheme. Let us assume that there are m supply nodes having respective supplies S_i for each $1 \leq i \leq m$, and n sink nodes having respective demands D_j for each $1 \leq j \leq n$. Denote $I(j)$ and $J(i)$ to be, respectively, the set of supply nodes that are linked to demand node j and the set of sink nodes that are linked to supply node i. Also, let us denote x_{ij} to be the flow from supply node i to sink node j, at a corresponding cost of c_{ij} units. To achieve a better control over the primal solutions produced by the Lagrangian dual

subproblems, let us impose certain implied upper bounds U_{ij} on the variables x_{ij} defined by $U_{ij} = \text{minimum}\{S_i, D_j\}$, for all ij. Then, the transportation problem can be written as

$$\text{Minimize} \sum_{i \in I(j)} \sum_{j \in J(i)} c_{ij} x_{ij}$$

$$\text{subject to} \sum_{j \in J(i)} x_{ij} = S_i \qquad \text{for all} \quad 1 \leq i \leq m$$

$$\sum_{i \in I(j)} x_{ij} = D_j \qquad \text{for all} \quad 1 \leq j \leq n$$

$$0 \leq x_{ij} \leq U_{ij} \qquad \text{for all} \quad ij$$

In our test runs, we dualize the demand or the supply constraints, whichever are fewer in number in order to reduce the dimension of the dual space (see Sherali and Myers, 1988, for several dualization schemes and their performances). Motivated by some real problems such as empty car relocation problems encountered in railroad networks (see Glickman and Sherali, 1985), we assume that the number of sink nodes (plants) are relatively less than the number of supply nodes (unloading ramps), and so, we dualize the demand constraints. The Lagrangian dual problem, therefore, becomes to maximize $\{\theta(\pi)\}$, where

$$\theta(\pi) = \quad \text{Minimize} \sum_{i \in I(j)} \sum_{j \in J(i)} (c_{ij} - \pi_j) x_{ij} + \sum_{j \in J(i)} \pi_j D_j$$

$$\text{subject to} \sum_{j \in J(i)} x_{ij} = S_i \qquad \text{for all} \quad 1 \leq i \leq m$$

$$0 \leq x_{ij} \leq U_{ij} \qquad \text{for all} \quad ij$$

The test problems are randomly generated using a standard linear programming generation scheme. First, given the numbers of source nodes, demand nodes, and arcs, we generate the index sets $I(j)$ and $J(i)$ for each $1 \leq i \leq m$ and $1 \leq j \leq n$, a prior primal optimal solution on [0, 10], and a prior dual optimal solution on [-10, 10]. Using the above entities, we then compute the amounts of supplies and demands, and the costs so that the generated solutions are indeed primal and dual optimal solutions (see Rosen and Suzuki, 1965). Table 5.1 gives the problem size along with the known optimal objective function value and the average demand (AVG-DEM) values for each of the 15 test problems thus generated. (The average demand values will be useful for evaluating the quality (feasibility) of the incumbent primal derived by the algorithm.)

Table 5.1 Transportation Test Problem Characteristics

PROBLEM	SUPPLIES	SINKS	ARCS	f^*	AVG-DEM
TR1	100	50	2000	17577	45.79
TR2	100	100	500	26439	60.06
TR3	100	100	50000	25389	59.51
TR4	150	100	7000	30522	49.67
TR5	200	50	5000	29815	38.25
TR6	200	100	5000	35950	45.39
TR7	200	100	7000	39359	43.30
TR8	200	100	9000	35082	42.73
TR9	200	100	10000	34004	43.30
TR10	200	100	12000	36273	44.14
TR11	250	100	10000	40506	40.46
TR12	300	100	10000	48778	39.52
TR13	300	100	10000	48476	49.56
TR14	300	150	15000	53559	44.70
TR15	300	200	20000	63558	49.77

Computational results for Stages I-II and for Stage III are presented in Tables 5.2 and 5.3, respectively. As before, f^* denotes the optimal objective function value, $\bar{\pi}$ denotes the incumbent dual solution that gives the best dual objective function value $\bar{z} \equiv \theta(\bar{\pi})$, and $\bar{\mathbf{x}}$ is the primal solution obtained at termination of Stage II, with objective function value $\bar{f}$. The final primal solution obtained at the end of Stage III is denoted by $\hat{\mathbf{x}}$, having a corresponded primal objective function value $\hat{f}$. Also, D(%) and P(%), respectively, represent the dual and primal optimalities in percentage that were achieved during the procedure, and are computed as follows.

$$\text{D}(\%) = \left[1 - \frac{f^* - \theta(\bar{\pi})}{|f^*|} \right] \times 100 \qquad \text{P}(\%) = \left[1 - \frac{|\bar{f} - f^*|}{|f^*|} \right] \times 100$$

Moreover, we also present the magnitude of constraint violations at termination, by specifying the maximum violation magnitude, denoted as MAX-VIO, and the average violation magnitude, denoted as AVG-VIO. These are respectively computed as follows.

$$\text{MAX-VIO} = \text{maximum} \left\{ \left| \sum_{i \in I(j)} x_{ij} - D_j \right| : j = 1,2,\ldots.. n \right\}$$

$$\text{AVG-VIO} = \frac{\sum_{j \in V} \left| \sum_{i \in I(j)} x_{ij} - D_j \right|}{|V|}$$

where $|V|$ denotes the cardinality of the set, V, of the violated constraints. Also, CPU indicates the corresponding computational cpu time in seconds on an IBM3090-300E computer at termination.

Table 5.2 presents the results obtained from Stages I and II. The dual problems for Stage I were solved as in Section 5.2 using Subroutine DFS to find a deflected subgradient within the default procedure VTVM. For VTVM, the parameters that we have used are $N_1 = 250$, MNCTI = 30, $\varepsilon = 0.1$, $\varepsilon_1 = 10^{-6}$, $(\gamma^{(1)}, \gamma^{(2)}) = (5,10)$, $(\sigma^{(1)}, \sigma^{(2)}) = (0.1, 0.5)$, and $(\beta^{(1)}, \beta^{(2)}) = (0.001, 0.005)$. (See Chapter 2 for the definitions of these parameters.) To reduce the computational burden that occurs when we impose the upper bounding constraints on the **x** variables in the subproblems, Stage I of the algorithm (HPDSA) solves the subproblems using only the nonnegativity constraints. Similar to our observations in Chapter 2, we found that the dual procedure DFS along with VTVM performs well, quickly achieving dual solutions within 96.50% of optimality at an average (over the 15 test problems).

In Stage II, we have used the pure-subgradient method for the dual problems, and we have used the simple strategy of averaging the subproblem optimal solutions for the primal recovery scheme as prescribed in Section 5.3. The main objective of this stage is to find a primal solution that is either near-feasible and optimal, so that the overall procedure HPDSA can be terminated (see Section 5.3 for the termination rule), or to find a good starting primal solution for Stage III. Also, note that we already have achieved acceptable dual solutions from Stage I. As motivated in Section 5.3, we incorporated the prescribed upper bounds on the **x** variables in the Lagrangian subproblems for this stage. Hence, each subproblem requires us to solve a set of m separable, bounded knapsack problems. (We also attempted a heuristic scheme for possibly obtaining a primal optimal solution via the Lagrangian dual subproblem when the incumbent dual solution is optimal. However, we found that it caused a heavy computational burden, without affecting the quality of the final solution achieved, and so, this scheme was abandoned.) For the step-length rule, we tested

several well known methods including VTVM and the standard rules given in (2.1), but we found that Shor's fixed step-length rule, $\lambda_k = \lambda_0{}^{k-1}$, for some $0 < \lambda_0 < 1$, performed well both in regard to feasibility and computational efficiency. Hence, we adopted this rule in Stage II, along with the parameters $N_2 = 50$, $\lambda_0 = 0.95$, $\varepsilon_2 = 0.02$, and $\varepsilon_3 = 0.001$.

The computational results show that, as expected, the dual solutions obtained at Stage I were not improved significantly due to the pure-subgradient direction strategy. However, the primal recovery scheme produces primal solutions that lie on near optimal objective contours, although their measure of feasibility is not favorable. The average optimality index obtained is 99.52%, and the average constraint violation AVG-VIO as a percent of the average demands AVG-DEM is 31%. Also, we note that none of the 15 problems has satisfied the near-feasibility and near-optimality termination rules during the N_2 iterations of Stage II (see Step 3 of the algorithm for Stage II). To further test the primal recovery scheme of Stage II, we ran the algorithm, by increasing N_2 up to 500. In this experiment, we observed that the feasibility index improved continuously, but slowly, with a disproportionate growth in the computational effort. However, we also found that the use of Stage II was necessary for the performance of Stage III. When we skipped Stage II, and instead increased the maximum number of iterations for Stage III (N_3) to 300, up from 100, we found that the final results obtained were much worse than the results given in Table 5.3. Based on this experimentation, we have selected to run Stage II for $N_2 = 50$ iterations.

Table 5.3 presents the results for Stage III. Here, RGAP of (5.9) denotes the relative duality gap between the penalty function h value and the dual value $\bar{z}$ at termination, $\hat{f}$ represents the primal objective function value at termination, and ITR(CPU) indi-

cates the number of iterations and the corresponding computational cpu time (in seconds) at termination.

As mentioned earlier in Section 5.4, we have used the standard exact penalty function recommended for the Primal-Dual subgradient algorithm by Sen and Sherali (1986), and we have adopted Subroutine DFS that was used in Stage I along with the Block-Halving method of Sherali and Ulular (1989) for the direction finding and step-length strategies.

The parameter values used in this context are the limit on the maximum number of iterations $N_3 = 100$, the maximum allowed number of iterations in each block $\nu = 45$, the maximum allowed consecutive failures before resetting $\tau = 10$, and the initial step-length parameter $\beta_1 = 1$. Also, for the termination criterion for the relative gap, we used $\varepsilon_2 = 0.02$ so that the algorithm stops whenever RGAP $\leq \varepsilon_2$, that is, $h(\mathbf{x}_k) \leq (1 + \varepsilon_2)\bar{z}$. For, the penalty parameter ω, we have used

$$\omega = \text{maximum}\left\{\overline{\omega},\ 2\ \text{maximum}\ \{\,|\bar{\pi}_i| : 1 \leq i \leq n\}\right\}$$

where $\overline{\omega} = 400$.

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, as one might expect, since this influences the definition of the penalty function as well as the determination of the step-length. This fact was further confirmed by running Stage III along with an improved dual solution. In general, we have observed that if the dual results in a dual solution that lies within 98% of optimality, then the primal refinement scheme of Stage III readily generates a near-optimal sol-

ution. In an overall examination, the penalty function method seems to perform well over Stage II in the sense that the constraint violation magnitudes are reduced significantly. Note here that the mean values of AVG-VIO and MAX-VIO over the 15 test problems have improved dramatically, from 14.029188 to 0.013351, and from 60.40807 to 0.132288, respectively. Moreover, the average percent optimality has also improved slightly from 99.52% to 99.96%. That is quite promising.

To summarize, in this research effort, we have proposed a hybrid primal-dual algorithm that enjoys the global convergence property, and produces a near-feasible and near-optimal solutions in a reasonably efficient manner. This can serve as a valuable tool in the context of Lagrangian dual/relaxation optimization. However, we remark here 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 5.2 Computational results for Stages I and II

PROB	f^*	$\bar{z}$	D(%)	$\bar{f}$	P(%)	AVG-VIO	MAX-VIO	CPU
TR1	17577	17505.52	99.59	17533.63	99.75	5.66996	21.87994	1.15
TR2	26439	26083.11	98.65	26444.40	99.98	5.44450	37.28006	2.25
TR3	25389	24073.45	94.82	25232.93	99.39	7.94063	30.78008	3.63
TR4	30552	29692.33	97.28	30621.62	99.67	14.12212	70.60005	7.18
TR5	29815	29528.93	99.04	29948.41	99.55	17.00705	50.68019	3.19
TR6	35950	35572.03	98.95	35842.92	99.70	8.06935	35.04016	3.32
TR7	39359	38788.69	98.55	39527.00	99.57	11.29029	52.68005	5.75
TR8	35082	33415.10	95.25	35095.97	99.96	15.32783	69.68022	8.74
TR9	34004	31364.95	92.24	34565.16	98.35	15.52072	61.28014	10.59
TR10	36273	34419.53	94.89	35913.88	99.01	25.73723	80.03981	14.73
TR11	40506	39769.76	98.18	40630.21	99.69	16.24890	94.70009	9.09
TR12	48778	46835.29	96.02	48592.46	99.62	15.60474	69.22021	7.95
TR13	48746	47087.65	97.14	48469.47	99.99	15.80388	61.18007	7.96
TR14	53559	50540.38	94.36	53992.92	99.19	17.37624	77.76009	16.35
TR15	63588	58846.03	92.54	63964.33	99.41	19.27438	93.31990	15.84

Table 5.3 Computational results for Stage III

PROB	f^*	$\bar{z}$	$\bar{f}$	RGAP	$\hat{f}$	P(%)	AVG-VIO	MAX-VIO	ITR(CPU)
TR1	17577	17505.52	17533.63	0.01439	17574.37	99.99	0.00900	0.11092	54(0.99)
TR2	26439	26083.11	26444.40	0.02017	26433.94	99.98	0.00425	0.03370	96(2.59)
TR3	25389	24073.45	25232.93	0.08254	25383.33	99.96	0.01678	0.04701	59(2.10)
TR4	30522	29692.33	30621.62	0.03790	30511.17	99.97	0.00127	0.01051	100(6.29)
TR5	29815	29528.93	29948.41	0.01754	29806.13	99.98	0.01161	0.03054	98(4.39)
TR6	35950	35572.03	35842.92	0.01758	35941.52	99.97	0.00618	0.02989	68(3.15)
TR7	39359	38788.69	39527.00	0.01951	39346.82	99.96	0.00466	0.03620	100(6.34)
TR8	35082	33415.10	35095.97	0.06403	35067.20	99.96	0.01181	0.07099	97(7.70)
TR9	34004	31364.95	34565.16	0.10876	33987.93	99.95	0.01929	0.06148	98(8.65)
TR10	36273	34419.53	35913.88	0.07999	36251.41	99.94	0.02250	0.57939	75(8.02)
TR11	40506	39769.76	40630.21	0.03118	40489.14	99.96	0.01260	0.11880	100(8.88)
TR12	48778	46835.29	48592.46	0.05179	48760.70	99.96	0.01209	0.19462	99(8.69)
TR13	48476	47087.65	48469.47	0.04395	48459.26	99.97	0.01700	0.34099	80(7.14)
TR14	53559	50540.38	53992.92	0.08358	53532.50	99.95	0.02009	0.06454	89(11.39)
TR15	63588	58846.03	63964.33	0.11462	63553.36	99.95	0.02503	0.18044	89(15.84)

6. Summary and Conclusions

6.1 Summary of Completed Results

Since the 1970's, nondifferentiable optimization techniques have been recognized as an important tool in mathematical programming, and many practical algorithmic approaches continue to employ subgradient based methods for expediting the solution process. However, most approaches deal with implementation techniques in conjunction with exploiting special structures inherent in the underlying problems. In other words, few theoretical and algorithmic developments exist for general nondifferentiable optimization problems. Also, recent computational reports indicate the need for some good and promising algorithms for such problems. Our research effort was therefore aimed at making some contributions in developing theoretically sound and computationally effective procedures for general nondifferentiable optimization problems. Moreover, since Lagrangian dual problems are solved very frequently in practice, we wanted to apply the proposed approaches to develop an

effective and robust procedure for obtaining both primal and dual solutions in this context.

Our first commitment was carried out by designing a new step-length rule and then, developing suitable subgradient deflection strategies. Three main theoretical step-length rules (see Chapter 1) that exist in the literature have being variously modified and widely used by many researchers. Our approach also modifies one of these rules which is computationally popular, but is focused on making no a priori assumption on the problem, and is based on changing a suitable target value in a timely fashion so that the resulting step-lengths yield good convergence behavior. In our analysis, we have first proved a convergence theorem for deflected subgradient methods (with bounded deflection parameters), using a fixed target value, and we have then extended this to the case of using variable target values, proving ε-convergence for this case. The new target value method can now be used to solve nondifferentiable optimization problems without any prior information about the solution space or the optimal objective function value.

The phenomenon of nondifferentiability often makes a selected subgradient almost perpendicular to the direction toward optimality, resulting in a zig-zagging of the solution path, and consequently, a slow tail-end behavior for the pure-subgradient method. Moreover, this phenomenon cannot be overcome by simply attacking the adopted step-length rule, because the ideal step-length that gives the closest point to the set of optimal solutions for each adopted direction of motion is relatively small due to the angle between the anti-subgradient and the direction toward optimality, and in any case, this does not truly address the zig-zagging behavior. On the other hand, changing (reducing) the foregoing angle could resolve this difficulty. This could

be done by deflecting subgradients favorably. However, the term favorable rotation cannot be easily executed because of the blindness of a selected subgradient and the lack of information about the set of optimal solutions. In this research, we have presented a deflection strategy that can be used in conjunction with the variable target value method. This new scheme deflects a subgradient to a direction that tends to point toward solutions having an objective function value that is less than the target value. Based on computational experiments on well-known standard test problems as well as on strongly convex problems, we have shown that the new variable target value method not only enjoys global convergence whether any information about the optimal objective function value is available or not, but it also appears to perform well empirically under reasonable stopping criteria. Also, we note that the deflection strategy is quite promising and competitive with other existing methods.

As an alternative deflection strategy, we have presented a variant of Shor's r-algorithm. This algorithm is motivated by the memoryless updates of quasi-Newton methods. Despite the theory and algorithmic performance of the r-algorithm, this method needs to store the space dilation matrix and update it at every iteration, resulting in a substantial computational burden for large-sized problems. Under a suitable choice of parameter values, the proposed memoryless scheme yields a direction of motion that turns out to be a convex combination of two successive anti-subgradients. We have proved convergence of this new method in conjunction with the variable target value method. Computational results on some standard dual transportation problems and assignment problems exhibit a substantial advantage of the proposed scheme over Shor's r-algorithm, even without any storage considerations.

Recovering a primal optimal solution in dual based subgradient approaches is another important issue. For example, when subgradient-based methods are used to solve Lagrangian dual problems, there is a great practical need for also obtaining a primal solution. In this context, we have presented primal convergence theorems that not only generalize the existing theorems for the pure-subgradient algorithm, but also can be applied to the case of using deflected subgradients. Our procedure simply involves taking a suitable convex combination of the Lagrangian dual subproblem solutions as the dual optimization proceeds, and so, adds a negligible burden to the dual algorithm in recovering a primal solution. We first addressed a generalized sufficiency condition that admits primal feasibility, and then we proved a primal-dual convergence property under the assumption of dual convergence. Along with the theorem, we presented some implementable choices of step-length rules, and corresponding rules for convex combinations weights, for general deflected subgradient procedures. The existing approaches turn out to be particular special cases of our procedure. With these results, we now have a wider class of more practically efficient choices for convex combination weights along with step-length rules for recovering primal solutions in dual based deflected subgradient approaches.

As a final contribution of this dissertation, we have presented a hybrid primal-dual subgradient algorithm for solving the Lagrangian dual problem. This procedure uses computationally effective deflection strategies to solve the dual problem itself, adopts an average weighting rule to recover primal optimal solutions during this process, and uses a primal penalty function method for further polishing the obtained primal solution. This algorithm includes several promising strategies and theories developed throughout this dissertation. Based on computational experiences using some transportation test problems, we have observed that the proposed algorithm performs

well in the sense of both feasibility and optimality at termination. An algorithmic shell is being developed for general distribution in order to assist researchers in solving various practical problems that are amenable to Lagrangian dual/relaxation approaches.

6.2 Recommendations for Further Research

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

When we were dealing with nondifferentiable optimization problems, we have assumed that the corresponding constraint set is sufficiently simple so that a projection operation can be achieved without any computationally expensive burden. On the other hand, there are only a few algorithms that can solve constrained problems, especially, having nondifferentiable constraint functions. Since our approaches presented in this dissertation perform well for general nondifferentiable optimization (NDO) problems (with unknown bounds), our next research should include a development of new NDO algorithms for generally constrained or nondifferentiably constrained NDO problems. Also, for problems that possess objective functions that are very expensive to evaluate so that the usual line search that requires several function evaluations causes a serious computational burden, we might adopt the variable target value method that requires only one function evaluation at each iteration, in con-

junction with conjugate subgradient or subgradient deflection methods. Moreover, a new conjugate gradient method for differentiable problems, motivated by the new deflection strategy, ODSA, that can be combined with the variable target value method, could also be developed.

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. However, our test results show that by far, the simple average weighting rule seems to perform the best, with even this "best" performance yielding not entirely satisfactory results. Therefore, we are eager to find some implementable, practical, and promising rules to enhance the second stage of Algorithm HPDSA, and hence, to reduce the burden of the third stage of the algorithm. Also, as we mentioned earlier in Chapter 5, developing a new subroutine that can generate a better (shorter norm) subgradient via an improved (with respect to primal feasibility) optimal solution to the underlying Lagrangian dual subproblem, would be necessary to accelerate the dual procedure and to obtain a satisfactory primal optimal solution. Therefore, for each specially structured problem, we need to exploit the inherent structure to develop subroutines with such a capability, that can then be interfaced with the optimization shell of Algorithm HPDSA developed herein. This needs experimentation with different classes of important, special problems.

Bibliograph

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

Ali, A. and J. Kennington. "The Symmetric M-Traveling Salesman Problem: A Duality Based Branch-and-Bound Algorithm," Discrete Applied Mathematics, 12, 259-276, 1986.

Allen, E., R. Helgason, J. Kennington, and B. Shetty. "A Generalization of Polyak's Convergence Result for Subgradient Optimization," Mathematical Programming, 37, 309-317, 1987.

Bazaraa, M.S. and J.J. Goode. "A Survey of Various Tactics for Generating Lagrangian Multipliers in the Context of Lagrangian Duality," European Journal of Operational Research, 3, 322-338, 1979.

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.

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

Bazaraa, M.S., H.D. Sherali, and C.M. Shetty. Nonlinear Programming: Theory and Algorithms, Second edition, John Wiley & Sons, New York, N.Y., 1993.

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

Clarke, F.H. "Generalized Gradients and Applications," Transaction of American Mathematical Society, 205, 247-262, 1975.

Clarke, F.H. "A New Approach to Lagrange Multiplier," Mathematics of Operations Research, 1(2), 165-174, 1976.

Dantzig, G.B. Linear Programming and Extensions, Princeton University Press, Princeton, N.J., 1963.

Dantzig, G.B. and P. Wolfe. "The Decomposition Algorithm for Linear Programming," Econometrica, 29, 767-778, 1961.

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

Erlenkotter, D. "A Dual-Based Procedure for Uncapacitated Facility Location," Operations Research, 26, 1, 992-1009, 1978.

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

Ermolev, Y.M. "Methods of Solution of Nonlinear Extremal Problems," Kibernatika, 2(4), 1-17, 1966.

Everett, H. "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation Resources," Operations Research, 11, 399-417, 1963.

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.

Fisher, M.L. and J.F. Shapiro. "Constructive Duality in Integer Programming," SIAM J on Applied Mathematics, 27(1), 31-52, 1974.

Fisher, M.L., W.D. Northup, and J.F. Shapiro. "Using duality to solve discrete optimization problems: theory and computational experience," Mathematical Programming Study, 3, 56-94, 1975.

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

Frendewey, J. and F. Glover. "Improved Algorithmic and Computation Procedure for Large Scale Embedded Network Linear Programming Problems," ORSA/TIMS Meeting, Houston, October, 1981.

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

Geoffrion, A.M. "Duality in Nonlinear Programming: A Simplified Applications-oriented Development," SIAM Review, 13, 1-37, 1971.

Geoffrion, A.M. "Lagrangian Relaxation for Integer Programming," in 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.

Goffin, J.L. "Nondifferentiable Optimization and the Relaxation Method," Nonsmooth Optimization: Proceedings of IIASA Workshop, C. Lemarechal and R. Mifflin (Eds.), 31-50, 1978.

Goffin, J.L. "Convergence Results in a Class of Variable Metric Subgradient Methods," Nonlinear Programming 4 , Mangasarian, Meyer, and Robinson (Eds.), 283-326, 1980.

Golub, G.H. and C. Van Loan. Matrix Computations, 2nd Ed., The Johns Hopkins University Press, 1989.

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.

Hooke, R. and T.A. Jeeves. "Direct Search Solution of Numerical and Statistical Problems," J. Association Computer Machinery, 8, 212-229, 1961.

Kelly, J.E. "The Cutting Plane Method for Solving Convex Programs," SIAM Journal on Applied Mathematics, 8, 703-712, 1960.

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.

Kiwiel, K.C. "A variable metric method of centers for nonsmooth minimization," CP-81-27, IIASA, Laxenburg, Austria, 1981.

Kiwiel, K.C. Methods of Descent of Nondifferentiable Optimization, Lecture Notes in Mathematics, 1133, Springer-Verlag, 1985.

Kiwiel, K.C. "A Survey of Bundle Methods for Nondifferentiable Optimization," Mathematical Programming, M. Iri and K. Tanabe (Eds.), KTK Scientific Publishers, Tokyo, Japan, 263-282, 1989.

Kiwiel, K.C. "Proximality Control in bundle Methods for convex Nondifferentiable Minimization," Mathematical Programming, 46, 105-122, 1990.

Kiwiel, K.C. "A Tilted Cutting Plane Proximal Bundle Method for Convex Nondifferentiable Optimization," OR Letters, 10, 75-81, 1991.

Kuhn, H.W. and A.W. Tucker. "Nonlinear Programming," in Proceeding of the Second Berkely Symposium in Mathematical Statistics and Probability, J. Neyman (Ed.), University of California Press, Berkely, CA, 1951.

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

Lemarachal, C. "Bundle Method in Nonsmooth Optimization," Nonsmooth Optimization: Proceedings of IIASA Workshop, C.Lemarechal and R. Mifflin (Eds.), 79-102, 1978.

Lemarechal, C. "A View of Line-search," Lecture Notes in Control and Information Science, A. Auslaunder, W. Oettli, J, Stoer (Eds.), Optimization and Optimal Control; Proceeding of a Conference Held at Oberwolfach, March 16-22, 59-78, 1980.

Lemarechal, C. "Numerical Experiments in Nonsmooth Optimization," Progress in Nondifferentiable Optimization, E. A. Nurminski (Ed.), IIASA, 61-84, 1982.

Lemarechal, C. "Bundle Methods, Cutting Plane Algorithm and σ-Newton Directions," Nondifferentiable Optimization: Motivation and Application,, V. F.Demyanov and D. Pallaschke (Eds.), Lecture Notes in Economics and Mathematical System, 255, 25-33, 1985.

Lemarechal, C. "Construction Bundle Methods for convex Optimization," Fermat Days 85: Mathematics for Optimization, Hiriart-Urrurty (Ed.), Elsevier Science Publisher B. V., North-Holland, 201-241, 1986.

Lemarechal, C. and R. Mifflin (Eds.). Nonsmooth Optimization, Proceedings of a IIASA Workshop, March 28-April 8, 1977, Pergamon Press, 1978.

Marsten, R.E. "The Use of the Boxstep in Discrete Optimization," Mathematical Programming Study, 3, M.L. Balinski and P. Wolfe (Eds.), North-Holland, Amsterdam, 127-144, 1975.

Marsten, R.E., W.W. Hogan, and J.W. Blankenship. "The Boxstep Method for Large-Scale Optimization," Operations Research, 23, 3, 389-405, 1975.

Mifflin, R. "An Algorithm for Constrained Optimization with Semismooth Functions," Mathematics of Operations Research, 2, 191-207, 1977b.

Mifflin, R. "A Modification and an Extension of Lemarechal's Algorithm for Nonsmooth Minimization," Mathematical Programming Study 17, 77-90, 1982a.

Mifflin, R. "Convergence of a Modification of Lemarechal's Algorithm for Nonsmooth Optimization," Progress in Nondifferentiable Optimization, E. A. Nurminski (Ed.), IIASA, 85-95, 1982b.

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

Nazareth, J.L. "A Relationship Between the BFGS and Conjugate-Gradient Algorithms and Its Implications for New Algorithms," SIAM Journal on Numerical Analysis, 26, 794-800, 1979.

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

Norkin, V.I. "A Method of Minimizing an Undifferentiable Function with Generalized-Gradient Averaging," Kibernetika, 6, 88-89, 1980.

Nurminski, E.A. and A.A. Zhelikhovskii. "ε-Quasigradient Method for Solving Non-smooth Extremal Problems," Kibernetika, 1, 109-113, 1977.

Oettli, W. "An Iterative Method, Having 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 Dokaly, 8, 593-597, 1967.

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

Polyak, B.T. "Subgradient Methods: A Survey of Soviet Research," Nonsmooth Optimization: Proceedings of IIASA Workshop, C. Lemarechal and R. Mifflin (Eds.), 5-30, 1978.

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

Polyak, E., D.Q. Mayne, and Y. Wardi. "On the Extension of the Constrained Optimization Algorithm from Differentiable to Nondifferentiable Problems," SIAM Journal on Control and Optimization, 21, 179-203, 1983.

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

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. 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 Subgradient Algorithm for Specially Structured Linear and Convex Programming Problems," Applied Mathematics and Optimization, 20, 193-221, 1989.

Shor, N.Z. "The Rate of Convergence of the Generalized Gradient Descent Method," Kibernetica, 4(3), 79-80, 1968.

Shor, N.Z. "Utilization of the Operation of Space Dilation in the Minimization of Convex Functions," Kibernetika, 6(1), 6-12, 1970a.

Shor, N.Z. "Convergence Rate of the Gradient Descent Method with Dilation of the Space," Kibernetika, 2, 80-85, 1970b.

Shor, N.Z. "Convergence of a Gradient Method with Space Dilation in the Direction of the Difference between Two Successive Gradients," Kibernetika, 4, 48-53, 1975.

Shor, N.Z. "Generalized Gradient Methods of Nondifferentiable Optimization Employing Space Dilatation Operations," Mathematical Programming: The State of Art, A. Bachem, M.Grotschel, and B. Korte (Eds.), 500-529, 1983.

Shor, N.Z. Minimization Methods for Nondifferentiable Function, Translated from the Russian by K. C. Kiwiel and A. Ruszczynski, Springer-Verlag, 1985.

Shor, N.Z. and N.G. Zhurbenko. "A Minimization Method Using the Operation of Space Dilation in the Direction of the Difference of Two Successive Gradients," Kibernetika, 3, 51-59, 1971.

Skokov, V.A. "Note on Minimization Methods Employing Space Stretching," Kibernetika, 4, 115-117, 1974.

Wolfe, P. "A Method of Conjugate Subgradients for Minimizing Nondifferentiable Functions," Mathematical Programming Study 3 , 145-173, 1975.

Zowe, J. "Nondifferential Optimization," Computational Mathematical Programming, K. Schittkowski (Ed.), NATO ASI Series, 323-355, 1985.

APPENDIX: User Interfaces for Algorithm HPDSA

Algorithm HPDSA can be used to find primal-dual solutions with user provided Lagrangian dual subproblem subroutine.

ALGORITHM PROCEDURE

Stage I: Dual Ascent

Stage II: Dual Ascent along with a Primal Recovering Scheme

Stage III: Primal Penalty Function Method

VARIABLES

NVAR: number of primal decision variables = dimension of primal problem

NCON: number of dualized constraints = dimension of dual problem

DUAL(NCON): dual solution

PRIM(NVAR): primal solution

G(NCON): subgradient of Lagrangian dual function

X(NVAR): optimal solution of Lagrangian dual subproblem

DVAL: dual function value

H(NVAR): subgradient of primal penalty function

PVAL: primal penalty function value

Note: These variables could be declared within a COMMON statement for necessary subroutines.

SUBROUTINE CALLS User supplied subroutines should be named and interfaced with the program shell as follows:

Stage I: SUBROUTINE DFUNCT (DUAL,G,DVAL,X)

Stage II: SUBROUTINE DDFUNCT (DUAL,G,DVAL,X)

Note: Subroutines DFUNCT and DDFUNCT may be exactly the same codes. However, as in Chapter 5, we recommend users to separate the two.

Stage III: SUBROUTINE PFUNCT (PRIM,DUAL,H,PVAL,PENAL)

SUBROUTINE PROJECT(PRIM)

Note: For problems that have linear objective functions with equality constraints, this program automatically uses the exact penalty function as default. However, for nonlinear programming problems, user should provide their own subroutine for defining an appropriate penalty function.

Note: Subroutine PROJECT performs the operation of projecting the primal solution onto the set of un-penalized primal constraints.

PARAMETERS

The following list includes all the necessary parameter values. The number in parenthesis indicates the default value. (For the definitions of unexplained parameters, see VTVM in Chapter 2.)

N1 (250): maximum number of iterations allowed for Stage I

MNCTI (30): maximum allowed number of consecutive target increases

GAMMA1 (5)

GAMMA2 (10)

SIGMA1 (0.1)

SIGMA2 (0.5)

BETA1 (0.001)

BETA2 (0.005)

EPS (0.1)

N2 (50): maximum number of iterations allowed for Stage II

N3 (100): maximum number of iterations allowed for Stage III

MITBL (45): maximum number of iterations for each block for block-halving method

MNCF (10): maximum allowed consecutive failures for block-halving method

BETAINI (1): starting step-length parameter for Stage III

PENAL (400): penalty parameter (for linearly constrained problems)

EPS1 (10^{-6}): termination parameter for the Euclidean norm of a selected sub-gradient

EPS2 (0.02): termination parameter for near-optimality

EPS3 (0.001): termination parameter for near-feasibility

Vita

Gyunghyun Choi was born on July 23, 1957 in Deagu, Korea. He graduted Kyung-Book high school in 1976. He received B.S. and M.S. degrees in Mathematics in 1980 and 1882, respectively, at Sogang University, Seoul, Korea. He had been an instructor at Sogang University and Seung-Kyun-Kwan University from 1983 to 1985. On December 1989, he received M.S. degree in Industrial Engineering and Operations Research at Virginia Polytechnic Institute and State University, and he joined the Ph.D. program of Industrial and Systems Engineering Department.

Gyunghyun Choi