Berhanu Guta

Subgradient Optimization Methods in Integer Programming with an Application to a Radiation Therapy Problem

Vom Fachbereich Mathematik

der Universität Kaiserslautern

genehmigte

Dissertation

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften

(Doctor rerum naturalium, Dr. rer. nat.)

Gutachter: Prof. Dr. Horst W. Hamacher

Prof. Dr. Francesco Maffioli

Datum der Disputation: 17. September 2003

D 386

Acknowledgments

I would like to express my sincere gratitude to my supervisor Prof. Dr. Horst W. Hamacher for making this research work possible. His guidance, valuable suggestions and great support has been a good influence for this work to be successfully done. I want to express my thanks also to Prof. Dr. Francesco Maffioli for his good will and effort to read and evaluate my thesis. I am indebted also to Prof. Dr. Helmut Neunzert who not only gave me the chance to come to Germany but also genuinely and fatherly concerned about my personal life, in general, and academic work, in particular. I would also like to thank Prof. Dr. Dietmar Schweigert for his support during the beginning of my Ph.D studies.

Many thanks go to all my colleagues at Optimization Group, AG Hamacher, in the University of Kaiserslautern for the good working atmosphere as well as for their friendly and unreserved support which have always made me to feel like being in my family at home.

I am also very grateful to DAAD (German Academic Exchange Service) for the financial support.

Finally, special thanks to my wife Asefash Geleta for her various supports and encouragements. I would like to thank also my son Naol and my daughter Hana for their patience and understanding while I was preparing the thesis.

CONTENTS

1. INTRODUCTION

The focus of this thesis is on the subgradient optimization methods which are used to solve nonsmooth optimization problems. We are particularly concerned with solving integer programming problems using the methodology of Lagrangian relaxation and dualization. This involves the problem of maximization of a Lagrangian dual function which is concave but nondifferentiable. The subgradient methods are suitable to solve such a problem since these procedures can make use of the concavity of the objective function. The goal of the thesis is to employ the subgradient optimization techniques to solve a large-scale practical radiation therapy planning problem which involves certain difficult constraints but imbedded in some tractable nice mathematical structures.

Integer optimization problems are generally difficult to solve because of their inherent combinatorial complexity. Unfortunately, almost all important generic classes of integer programming problems are NP-hard. Furthermore, many integer programming problems of practical relevance are of large-size. Hence, the scale at which these problems arise in applications and the explosive exponential complexity of their search spaces preclude the use of simplistic enumeration and search techniques.

Therefore, in order to solve practical integer programming problems we may need to resort to approximation schemes and problem specific algorithms which can exploit some special structures of the problem at hand. Indeed, many practically relevant integer programming problems are usually composed of some nice mathematical structures adjoined with certain complicating conditions (or constraints)

which make any solution procedure difficult. The method of Lagrangian relaxation lifts these complicating constraints and makes use of the special structure to solve the relaxed problem. This produces a lower bound for a minimization problem. At least a good approximate solution or the best (tight) bound can be obtained by subgradient optimization methods where a subgradient vector is obtained by minimizing the relaxed primal problem and the dual variables are updated iteratively along the direction of this subgradient vector. This approach has received recognition and is proven to be a very useful tool to solve various difficult discrete optimization problems since the first successful works of Held and Karp in 1970's ([44], [45]). They used Lagrangian relaxation approach based on minimum spanning trees to devise a successful algorithm for the travelling salesman problem.

In Chapter 2, we will review some important techniques in the approach of the Lagrangian relaxation of integer programming problems and formulate the related Lagrangian dual problem. We also discuss properties of the dual problem, optimality conditions as well as the general structure of the dual objective function.

A major challenge in the method of Lagrangian relaxation of a minimization problem of an integer programming problem is to maximize effectively the Lagrangian dual function which is defined only implicitly and is nondifferentiable, concave, and piecewise affine. The subgradient method is frequently used to solve such problems since this method does not demand differentiability. In fact, it is an iterative procedure which searches for an optimal solution using the direction of the gradient vector at each point where the gradient of the function exists; but replaces the gradient vector by a subgradient vector at a point where the gradient does not exist. The subgradient of a function at a point is generally not unique and the functional value of the objective function may not be improved along a given subgradient direction. Consequently, line search techniques as in the gra-

dient methods of the case of minimizing smooth functions can not be suitable for subgradient methods. Hence, subgradient methods are not directly analogous to the gradient methods of smooth nonlinear optimization problems. Depending on the particular strategy of determining a step direction, there are different kinds of subgradient procedures which differ in their performance. These procedures will be discussed in detail.

In Chapter 3, different variants of subgradient methods will be investigated and a unified presentation of the methods will be given. Their optimality conditions and convergency properties will be also analyzed. The central drawback of this methods is that their convergence is usually slow which is mainly caused by the so called zigzagging phenomena. The slow convergence is a crucial problem, particularly, in view of the need to solve large-scale problems. Various versions of subgradient methods such as the modified subgradient method [21], the average direction strategy [77], and conditional subgradient methods [53] have been developed with the intent to improve the speed of convergence of the traditional subgradient method by controlling its zigzagging phenomena. Although these procedures address the issues of zigzagging phenomena, zigzagging has not been precisely defined in the literature beyond the geometrical and intuitive understanding of the phenomena. Furthermore, none of these procedures can completely eliminate zigzagging. In order to make the analysis precise, we formalize the definition of zigzagging in terms of mathematical expressions and explore in detail the kind of zigzagging which can be manifested by each of the variants of the subgradient methods. Furthermore, based on the formalized definitions of zigzagging, we determine suitable values of parameters, such as deflection parameter, in some of the existing subgradient procedures in order to strengthen their capability to control zigzagging. Yet, even with such a choice of suitable parameters, none of the available procedures can completely eliminate zigzagging. Consequently, the basic difficulty with regard to the slow convergence has not been altered.

In Chapter 4, we will introduce a new and general strategy for step direction selection that can help to completely eliminate all kinds of zigzagging phenomena of subgradient procedures and provide convergence conditions. We also consider application of the new procedure to solve the Lagrangian dual of integer programming problem.

Another drawback of the traditional subgradient method and its modifications is that the methods do not directly provide primal solutions. Indeed, in the context of Lagrangian dual formulations, subgradient methods solve the Lagrangian dual problem and provide a tight bound to the objective value of the primal problem. But solving the dual problem might not provide primal optimal, or even a feasible, solution.

In Chapter 5, we discuss the techniques of directly constructing a primal solution within the subgradient schemes. In particular, we will describe extensions of the subgradient methods which, without a significant additional computational effort, can produce primal as well as dual solutions. One of these methods is the ergodic subgradient method due to Larsson et al. 1999 ([54]) which is an extension of the conditional subgradient procedure. This method constructs ergodic (averaged) sequences of the solutions of a subproblem within the conditional subgradient method such that the sequence converges to a primal solution which satisfies the primal-dual optimality conditions together with certain limit point of the dual variables. The authors originally proposed the algorithm for a convex programming problem and have proved the convergence of the ergodic sequence to a point which satisfies the dual-primal optimality conditions. Indeed, such a point exists for convex programming problems satisfying an appropriate constraint qualification. However, we cannot expect such a result for the problem discussed in this thesis due to our demand for integrality and the existence of a duality gap; as a consequence of which the primal-dual optimality conditions can

never be generally satisfied. Hence we can aim only at a near-optimal solution which can be reached as will be shown.

In Chapter 5, we will also describe and analyze the volume algorithm which has been recently developed by Barahona and Anbil [6]. The volume algorithm was developed as an extension of the deflected subgradient methods to produce primal solutions of a linear programming problem within the deflected subgradient procedure. Its name originated from a result on linear programming duality stating that one can derive a primal solution from the volumes below the faces which are active at the maximum point of the dual function. Basically, a primal solution is constructed as a convex combination of the solutions of the subproblem. Similar to the case of the ergodic subgradient method, we can expect only a near-optimal primal solution when the volume algorithm is applied to solve an integer programming problem, since there is no guarantee to satisfy integrality from convex combinations of points.

In Chapter 6, we apply the subgradient optimization methods to solve the problem of modulation of radiation beam in the cancer radiation therapy using a multileaf collimator. Particularly, we will consider the problem of minimizing total delivery time of a given radiation dose (intensity matrix) to a cancer patient. The problem is NP-hard [20] and thus far there exists no method for solving the problem to optimality. We introduce a new, fast and efficient algorithm which combines exact and heuristic procedures to solve the problem. The exact subproblem in our heuristic algorithm deals with the problem of minimizing beam-on time of binary intensity matrices generated by the algorithm. We use the Hamacher-Boland network flow model [16] to solve this subproblem. The difficulty here is that the network involves side constraints and consists of large number of nodes and arcs. Hence problems with a large number of bixels in the multileaf collimator cannot be solved with a reasonable time. We prescribe the rules for preprocessing in order to reduce the number of arcs and nodes. More-

over, by using the Lagrangian relaxation and dualization technique, we lift the side constraints and obtain a pure minimum cost flow (circulation) problem as a subproblem. We solve the resulting circulation problem using the negative cycle cancelling method. Numerical implementations show that our solution method dramatically reduces computational time as well as memory space requirement of large-sized problems. Furthermore, comparisons of numerical results of our new algorithm with that of other alternative algorithms show its various good qualities.

Some Notations:

The following notations are frequently appearing in this thesis.

Let $u = (u_1, u_2, \ldots, u_n)$ and $v = (v_1, v_2, \ldots, v_n)$ be vectors in $\mathbb{R}^n$. Then,

- $uv := \sum_{i=1}^{n} u_i v_i$, independent of whether any of the vectors is a row or column vector.

- $u \geq 0$ means $u_i \geq 0$ for each $i = 1, \ 2, \ldots n$.

- $\|u\| := \sqrt{uu}$ - the Euclidean norm .

- $\mathrm{P}_\Omega(u) = \mathrm{argmin}_{z \in \Omega}\{\|z - u\|\}$ - the Euclidean projection of u onto a closed, convex set Ω; i.e., the point in Ω closest to u.

- $\mathrm{dist}(x, S) := \min\{\|y - x\| : y \in S\}$- distance of a point $x \in \mathbb{R}^n$ from a set $S \subseteq \mathbb{R}^n$.

- $\mathbb{Z}_+^n = \{x \in \mathbb{Z}^n \ : \ x_i \geq 0, \ i = 1, 2, ..., n\}$ - the set of vectors of non-negative integers.

- $A^i = (a_{i1}, a_{i2}, \ldots, a_{in})$, *i-th* row of a matrix $A \in \mathbb{R}^{m \times n}$.

- $\mathrm{int}(\Omega) = \{x \in \Omega : \ \|y - x\| < \epsilon \Rightarrow y \in \Omega, \text{ for some } \epsilon > 0\}$ - the set of interior points of Ω, where $\Omega \subseteq \mathbb{R}^n$.

- $\mathrm{bd}(\Omega) = \Omega - \mathrm{int}(\Omega) = \{u \in \Omega : u \notin \mathrm{int}(\Omega)\}$ - the set of boundary points of a set Ω.

- $\mathrm{conv}(X) = \{x | \ x = \sum_{i=1}^{m} \alpha_i x^i, \quad \sum_{i=1}^{m} \alpha_i = 1, \quad \alpha_i \geq 0, \quad x^i \in X \ \ \forall i\}$ - the convex hull of a set X.

- scalars are indexed by subscripts as $\alpha_1, \ldots, \alpha_k$; but vectors are indexed by superscripts as $u^1, \ldots, u^k$.

- Ω^* - set of optimal solutions of an optimization problem whose feasible set is Ω.

- ϕ^* - optimal objective value of an optimization problem whose objective function is $\phi(.)$.

2. LAGRANGIAN RELAXATION AND DUALITY

In this chapter we review some important techniques of the Lagrangian relaxation approach for integer programming problems and formulate the related Lagrangian dual problem. We also discuss properties of the Lagrangian dual problem, optimality conditions and the structure of the dual objective function.

2.1   Introduction

Lagrangian dual arises from a Lagrangian relaxation. Lagrangian relaxation is a useful technique in nonlinear programming, large-scale or structured linear, convex, and integer programming. In this thesis, we restrict ourselves to the case of integer programming. Hence, we consider an integer programming problem

$$\text{(IP)} \qquad z^* = \quad \min \quad cx \qquad\qquad (2.1)$$
$$\text{s.t.} \quad Ax \geq b$$
$$x \in \mathbb{X} = \{x \in \mathbb{Z}_+^n : \ Dx \geq d\}$$

where $c \in \mathbb{R}^n$, (A, b) and (D, d) are $m \times (n+1)$ and $r \times (n+1)$ matrices, respectively, and $x \in \mathbb{Z}_+^n$ means that x is an n-vector of non-negative integers. $\mathbb{X}$ is a set of discrete (integral) points in a polyhedral and assumed to be non empty and bounded for convenience. We call the problem (IP) the *primal problem* and its solution a *primal solution*. Suppose that the constraints $Dx \geq d$ are "nice" in a sense that an integer program with just these constraints, i.e.,

$$\min\{\bar{c}x : Dx \geq d, \ x \in \mathbb{Z}_+^n\},$$

can be "easily " solved for any choice of $\bar{c}$ while the whole problem which includes also the other constraints $Ax \geq b$ may be significantly harder to solve. We

call those constraints $Ax \geq b$, which make a solution procedure difficult, the *complicating constraints*. A common approach to solve this problem, perhaps approximately, is to solve its Lagrangian dual problem obtained via Lagrangian relaxation([44], [45], [74],[39], [75], [36]). In the Lagrangian relaxation approach, the complicating constraints $Ax \geq b$ are relaxed by introducing a multiplier vector $u \in \mathbb{R}_+^m$, called *Lagrangian multiplier*, and the Lagrangian function

$$\mathcal{L}(x, u) = cx + u(b - Ax).$$

Given $u \in \mathbb{R}_+^m$, the *Lagrangian relaxation* problem is then to solve the subproblem:

SP(u) $$\phi(u) = \min \mathcal{L}(x, u) \tag{2.2}$$
$$\text{s.t.} \quad x \in \mathbb{X}$$

yields the function ϕ determined pointwise by the optimal objective value of the subproblem. Note that, for any $u \in \mathbb{R}_+^m$, $x \in \mathbb{X} \cup \{x : Ax \geq b\}$, and any optimal solution x^* of the (IP) it holds that

$$\phi(u) \leq \mathcal{L}(x, u) \leq cx \quad \text{and} \quad \phi(u) \leq \mathcal{L}(x^*, u) \leq cx^* = z^*.$$

The relative simplicity of solving the subproblem and the fact that $\phi(u) \leq z^*$ allows SP(u) to be used to provide lower bounds for (IP). In general, corresponding to different values of u, one obtains different lower bound $\phi(u)$ to the primal optimal value z^*. Thus, to obtain the best (greatest) lower bound of z^*, the best choice of u would be any one which is an optimal solution to the *Lagrangian dual problem*:

(LD) $$\phi^* = \max \{\phi(u) : u \geq 0\} \tag{2.3}$$

where $\phi(u)$ is given pointwise by the subproblem SP(u):

$$\phi(u) = \min \quad cx + u(b - Ax)$$
$$\text{s.t.} \quad x \in \mathbb{X}. \tag{2.4}$$

The function ϕ is called the *dual function*. Observe that when m constraints that have been dualized are equality constraints of the form $Ax = b$, the corresponding

Lagrangian multipliers are unrestricted in sign and the Lagrangian dual becomes

$$\phi^* = \max_{u \in \mathbb{R}^m}\{\phi(u)\}.$$

Other possible relaxation problem of the IP is a linear programming relaxation. For the IP problem the linear programming relaxation is given by

$$\text{(LP)} \qquad \begin{aligned} z_{LP}^* = \min \quad & cx \\ \text{s.t.} \quad & Ax \geq b \\ & x \in \bar{\mathbb{X}} = \{\, x \in \mathbb{R}_+^n : \ Dx \geq d\}. \end{aligned}$$

That is, the integrality constraints are simply replaced by its continuous relaxation. In the case of a small sized and simple problem, the IP can be solved by solving its LP relaxation using a simplex based procedure and then apply a branch and bound or a cutting plane method to generate an integral solution. In such a procedure the optimal value of the LP relaxation problem z_{LP}^* provides a lower bound to z^* just as in the case of the LD. However, in IP problems with some complicating constraints or with a large number of constraints and variables the Lagrangian dual approach would be preferable since

- LD can make use of the available special structures of the problem by removing the complicating constraints.

- even if no special structure is available, the number of constraints are reduced in the subproblem (2.4).

- using concavity of the dual function (to be justified later), there is easier method to solve the Lagrangian dual than the simplex-based methods of solving LP.

- ϕ^* can be tighter than or else at least as good as z_{LP}^* (see, Corollary 2.5).

- one can construct an approximate solution of the IP from the solution of the subproblem (2.4) easily as compared to the simplex-based solution of the LP relaxation (see Chapter 5).

2.2   Properties of the Dual Problem and Dual Function

In this section we review some properties of the dual function related to concavity and subdifferentiability. We also summarize the main properties of the Lagrangian dual problem (LD) which will be used in the remainder of our discussion. The proof of the next theorem follows directly from the fact that $\mathrm{SP}(u)$, for any $u \in \mathbb{R}_+^m$, is a relaxation of (IP) and the detail of the proof can be found on standard text books such as [64] and [84].

Theorem 2.1: **(Weak Lagrangian Duality)**
Let (IP), $\mathrm{SP}(u)$ and (LD) be as defined above and x be a feasible solution of (IP). Then for any $u \geq 0$,
$\phi(u) \leq cx$. Consequently $\phi(u) \leq \phi^* \leq z^*$.

$\square$

The above theorem shows that any feasible solution of the dual problem as well as its maximum value is a lower bound to the optimal value of the primal problem (IP). The next theorem provides the conditions for which the optimal dual problem yields a solution to (IP).

Theorem 2.2: **(Strong Lagrangian Duality)**
Let (IP), $\mathrm{SP}(u)$ and (LD) be as defined above. If $\hat{x}$ solves the subproblem $\mathrm{SP}(\hat{u})$ for some $\hat{u} \geq 0$, and in addition

$$A\hat{x} \geq b \tag{2.5}$$

$$\hat{u}(b - A\hat{x}) = 0 \tag{2.6}$$

then $\hat{x}$ is an optimal solution of the primal problem (IP) and $\hat{u}$ is an optimal solution of the Lagrangian dual problem (LD).

Proof: An $\hat{x}$ satisfying the hypothesis of the theorem is feasible in (IP) because of (2.5) and

the fact that $\hat{x}$ solves SP($\hat{u}$) means $\hat{x} \in \mathbb{X}$. In particular, feasibility of $\hat{x}$ implies that $c\hat{x} \geq z^*$. But from the weak duality theorem, Theorem 2.1, we also have $z^* \geq \phi(\hat{u}) = c\hat{x} + \hat{u}(b - A\hat{x}) = c\hat{x}$. The first equality in this relations follows from the definition of $\phi(\hat{u})$ and the second equality follows from (2.6). Putting these together we have

$$c\hat{x} \geq z^* \geq \phi(\hat{u}) = c\hat{x}$$

from which one can conclude that $z^* = \phi(\hat{u}) = c\hat{x}$. That is, $\hat{x}$ is an optimal solution of (IP). Moreover, using this result and the fact that $c\hat{x}$ is an upper bound of ϕ^*, we obtain

$$\phi(\hat{u}) \leq \phi^* \leq c\hat{x} = \phi(\hat{u}).$$

which means $\phi^* = \phi(\hat{u})$. That is, $\hat{u}$ is an optimal solution of the Lagrangian dual (LD). $\qquad\qquad\square$

The implication of (2.5) and (2.6) is that $\hat{x}$ computed by the subproblem SP($\hat{u}$) is optimal for the IP problem if it satisfies the dualized constraints, i.e., $A\hat{x} \geq b$, and $A^i\hat{x} = b_i$ whenever $\hat{u}_i > 0$, where A^i is the *i-th* row of the matrix A. In the case of a problem where the dualized constraints are all equations, i.e, if $Ax = b$ in the primal problem (IP), then the LD is

$$\max\{\phi(u) : \ u \gtrless 0\}$$

and Theorem 2.2 implies that if $\hat{x}$ solves the subproblem SP($\hat{u}$) and satisfies the dualized constraints, then $\hat{x}$ and $\hat{u}$ are optimal solutions of the primal problem and the Lagrangian dual problem, respectively, since (2.6) is satisfied for any $\hat{u} \in \mathbb{R}^m$ and any primal feasible solution $\hat{x}$.

In general, however, it is not possible to guarantee finding feasible solutions $\hat{x}$ and $\hat{u}$ for which $\phi(\hat{u}) = c\hat{x}$. For most problem instances of integer programming the strong Lagrangian duality does not hold and thus, there is in general a gap between the optimal primal and dual objective values. The difference $z^* - \phi^*$ is known as the *Lagrangian duality gap*. Even for a problem without such

duality gap, an optimal solution $\hat{x}$ of the subproblem SP($\hat{u}$) corresponding to a Lagrangian optimal dual solution $\hat{u}$ may not be feasible in the primal problem, IP.

If $\hat{x}$ is an optimal solution of the subproblem SP($\hat{u}$) and satisfies (2.5), but not necessarily (2.6), then $\hat{x}$ is called an ϵ-*optimal solution* of (IP) with respect to $\hat{u}$ with $\epsilon = \hat{u}(A\hat{x} - b) > 0$. In this case $\hat{x}$ is a feasible solution of (IP) and hence $c\hat{x}$ is an upper bound to the optimal primal objective value z^*. In fact, it holds that

$$c\hat{x} - z^* \leq \epsilon$$

since $\phi(\hat{u}) = c\hat{x} + \hat{u}(b - A\hat{x}) \leq z^*$ implies that $c\hat{x} - z^* \leq \hat{u}(A\hat{x} - b) = \epsilon$. Hence we have shown the following result.

Corollary 2.3: Let (IP), SP(u), and (LD) be as given above. If $\hat{x}$ is an ϵ-optimal solution of (IP) with respect to $\hat{u}$, then

$$\phi(\hat{u}) \leq z^* \leq c\hat{x} \quad \text{and} \quad c\hat{x} - \phi(\hat{u}) = \epsilon. \tag{2.7}$$

$\square$

The diagram below, Figure 2.1, illustrates the relations given in Corollary 2.3 where $\hat{x}$ is an ϵ-optimal solution of (IP) with respect to $\hat{u}$.

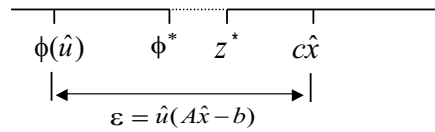*Fig. 2.1:* Order relation of dual and primal values with the gap ϵ where $\hat{x}$ is an ϵ-optimal solution of (IP) with respect to $\hat{u}$.

The next theorem characterizes the Lagrangian dual problem. This characterization was provided by Geoffrion [39] and is based on the convex hull, denoted by conv($\mathbb{X}$), of points in $\mathbb{X}$ where conv($\mathbb{X}$) is a set of all convex combinations of

the points in $\mathbb{X}$, i.e.

$$\text{conv}(\mathbb{X}) = \{\bar{x} : \ \bar{x} = \sum_i \alpha_i x^i, \ \sum_i \alpha_i = 1, \alpha_i \geq 0, x^i \in \mathbb{X}, \forall i \ \}.$$

(See, for instance, [47] or [70] for further detailed discussion on convex set and convex hull.)

Theorem 2.4: (Geoffrion, 1974)
Let (IP), SP(u) and (LD) be as defined above and ϕ^* be the optimal value of the Lagrangian dual problem (LD). Then

$$\phi^* = \min \ \ cx \tag{2.8}$$
$$\text{s.t.} \ \ Ax \geq b$$
$$x \in \text{conv}(\mathbb{X}).$$

Proof: Since $\mathbb{X} \subseteq \mathbb{Z}_+^n$ and bounded, $\mathbb{X}$ consists of a finite, but possibly very large, number of points $x^1, x^{2,}, \ldots, x^T$. Then,

$$
\begin{aligned}
\phi^* \ &= \ \max_{u \geq 0} \ \{\phi(u)\} \\
&= \ \max_{u \geq 0} \ \{\min \ cx + u(b - Ax) : \ x \in \mathbb{X}\} \\
&= \ \max_{u \geq 0} \ \{\min \ cx^i + u(b - Ax^i) \ : \ i = 1, 2, \ldots, T\} \\
&= \
\begin{cases}
\max \ \eta \\
\text{s.t.} \ \ \eta \leq cx^i + u(b - Ax^i), \quad i = 1, 2, \ldots, T. \\
\quad \eta \in \mathbb{R}^1, \quad u \in \mathbb{R}_+^m,
\end{cases}
\tag{2.9}
\end{aligned}
$$

where the new variable η is a lower bound to $\{cx^i + u(b - Ax^i) : \ i = 1, 2, \ldots, T\}$. The latter problem, (2.9), is a linear programming problem (usually large scale LP) with variables $(\eta, u) \in \mathbb{R}^1 \times \mathbb{R}_+^m$. Taking its dual yields:

$$
\begin{aligned}
\phi^* = \min \ \ & \sum_{t=1}^T \alpha_t (cx^t) \\
\text{s.t.} \ \ & \sum_{t=1}^T \alpha_t (Ax^t - b) \geq 0 \\
& \sum_{t=1}^T \alpha_t = 1 \\
& \alpha_t \geq 0; \ \ t = 1, 2, \ldots, T.
\end{aligned}
$$

Now setting $x = \sum_{t=1}^{T} \alpha_t x^t$, with $\sum_{t=1}^{T} \alpha_t = 1$, $\alpha_t \geq 0$, for each $t = 1, 2, \ldots, T$, we get

$$\phi^* = \min \quad cx$$
$$\text{s.t} \quad Ax \geq b$$
$$x \in \text{conv}(\mathbb{X}),$$

which is as required. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

The above theorem tells us how strong a bound obtained from the Lagrangian dual is. Indeed, the bound provided by the Lagrangian dual is at least as large as (in some cases larger than) the lower bound obtained from the linear programming relaxation of (IP) as shown in the next corollary.

Corollary 2.5: (**Lagrangian Dual versus Linear Programming Relaxation**)
Let (IP), and (LD) be as given above and let (LP) be the linear programming relaxation of (IP). Then

$$\phi^* \geq z_{LP}^*$$

where ϕ^* and z_{LP}^* are the optimal objective values of the (LD) and (LP), respectively.

Proof: The problem (LP) is defined as

$$z_{LP}^* = \min \quad cx$$
$$\text{s.t.} \quad Ax \geq b$$
$$x \in \bar{\mathbb{X}} = \{x \in \mathbb{R}_+^n : \; Dx \geq d\}.$$

Since $\mathbb{X} \subseteq \bar{\mathbb{X}}$ implies that $\text{conv}(\mathbb{X}) \subseteq \text{conv}(\bar{\mathbb{X}}) = \bar{\mathbb{X}}$, it follows that the (LP) is a relaxation of problem (2.8), in Theorem 2.4. Hence, $\phi^* \geq z_{LP}^*$. $\qquad$ $\square$

We now consider two examples to demonstrate the existence of a problem instance for which the optimal Lagrangian dual value is strictly greater than that of the linear programming relaxation and also where the two are equal.

Example 2.1: In this example we consider Lagrangian dual of uncapacitated warehouse location problem as given by Parker and Rardin ([66], page 208). In warehouse location problems one chooses which of a given set of potential warehouses $i \in W = \{1, 2, \ldots, n\}$ to build in order to supply demand points $j \in D = \{1, 2, \ldots, m\}$ at minimum total cost. Costs include both a fixed cost $f_i > 0$ for building the warehouse i and transportation costs $\sum_{i \in W} \sum_{j \in D} c_{ij} x_{ij}$, where x_{ij} is the amount shipped from i to j and c_{ij} is its unit transportation cost. By introducing variables y_i where

$$y_i = \begin{cases} 1, & \text{if warehouse } i \text{ is built} \\ 0, & \text{otherwise} \end{cases}$$

we obtain the formulation:

$$\text{(P1)} \qquad \min \ \sum_{i \in W} \sum_{j \in D} c_{ij} x_{ij} + \sum_{i \in W} f_i y_i$$

$$\text{s.t.} \ \sum_{i \in W} x_{ij} \geq d_j \quad \forall j \in D \tag{2.10}$$

$$\sum_{j \in D} x_{ij} \leq (\sum_{j \in D} d_j) y_i \quad \forall i \in W \tag{2.11}$$

$$0 \leq x_{ij} \leq d_j \tag{2.12}$$

$$y_i \in \{0, 1\} \tag{2.13}$$

Here, d_j is the demand at point j. Suppose we dualize (2.10). The Lagrangian dual of (P1) is then

$$\text{(LD-1)} \qquad \phi^* = \max \ \{\phi(u) : \ u \in \mathbb{R}_+^{|D|} \}$$

where $\phi : \mathbb{R}_+^{|D|} \longrightarrow \mathbb{R}$ is given by the subproblem

$$\text{SP1}(u) \qquad \phi(u) = \sum_{j \in D} u_j d_j + \min \sum_{i \in W} \sum_{j \in D} (c_{ij} - u_j) x_{ij} + \sum_{i \in W} f_i y_i$$
$$\text{s.t.} \ (2.11), (2.12), (2.13),$$

which is equivalent to:

$$\phi(u) = \sum_{j \in D} u_j d_j + \sum_{i \in W} \begin{pmatrix} \min \ f_i y_i + \sum_{j \in D} (c_{ij} - u_j) x_{ij} \\ \text{s.t} \quad (2.11), (2.12), (2.13) \end{pmatrix}. \tag{2.14}$$

Thus, the subproblem can be solved by separately solving one trivial problem for each i. Specifically, we consider $y_i = 0$ which implies, by (2.11), $x_{ij} = 0$ for all j, versus $y_i = 1$ in which case

$$x_{ij} = \begin{cases} d_j, & \text{if } c_{ij} - u_j < 0 \\ 0, & \text{otherwise} \end{cases}$$

This implies, for any $u \geq 0$,

$$\phi(u) = \sum_{j \in D} u_j d_j + \sum_{i \in W} [\min\{0, f_i + \min\{(c_{ij} - u_j)d_j, 0\}\}].$$

One specific instance with $i = 1, 2, 3$ and $j = 1, 2$ is:

$$\begin{aligned}
\min \quad & 8x_{11} + 7x_{12} + 5x_{21} + 4x_{22} + 1x_{31} + 3x_{32} + 36y_1 + 12y_2 + 36y_3 \\
\text{s.t.} \quad & x_{11} + x_{21} + x_{31} \geq 6 \\
& x_{12} + x_{22} + x_{32} \geq 6
\end{aligned}$$

$$x \in \mathbb{X} = \left\{ x : \begin{array}{l} x_{11} + x_{12} \leq 12y_1 \\ x_{21} + x_{22} \leq 12y_2 \\ x_{31} + x_{32} \leq 12y_3 \\ 0 \leq x_{ij} \leq 6 \quad \text{for} \quad i = 1, 2, 3 \quad \text{and} \quad j = 1, 2 \\ y_1, y_2, y_3 \in \{0, 1\} \end{array} \right\}$$

The optimal solution of this problem is to open only warehouse 3, i.e., $y_3 = 1$; $x_{31} = x_{32} = 6$ and it yields $z^* = 6 + 18 + 36 = 60$. All other variables are 0. Now consider (2.14) for $(u_1, u_2) = (4, 6)$. The subproblem for $i = 2$ is:

$$\begin{aligned}
\min \quad & (5 - 4)x_{21} + (4 - 6)x_{22} + 12y_2 \\
\text{s.t} \quad & x_{21} + x_{22} \leq 12y_2 \\
& 0 \leq x_{2j} \leq 6, \quad j \in \{1, 2\} \\
& y_2 \in \{0, 1\}.
\end{aligned}$$

If $y_2 = 0$, then $x_{21} = x_{22} = 0$ at cost 0. If $y_2 = 1$, $x_{21} = 0$ but $x_{22} = 6$, then cost is $12 + (-2)6 = 0$; so y_2 may be either 0 or 1. A check of similar problems for

$i = 1$ and 3 shows that $y_1 = y_3 = 0$ is optimal. Thus

$\phi(u) = 6(4) + 6(6) + 0 + 0 + 0 = 60$, which is equal to z^* and thus ϕ^*. i.e., $\phi^* = z^* = 60$.

However, the LP relaxation of the problem yields a solution $\bar{x}_{22} = \bar{x}_{31} = 6$, $\bar{y}_2 = \bar{y}_3 = 0.5$, with cost $z^*_{LP} = 54$. Therefore, in this case, ϕ^* is strictly greater than z^*_{LP}.

$\square$

Example 2.2: Consider the class of a Boolean problem

$$(\text{P2}) \qquad\qquad \min \quad cx$$
$$\text{s.t} \quad Ax \geq b$$
$$x \in B^n = \{x \in \mathbb{Z}^n : x_j \in \{0, 1\}\}$$

where all linear constraints are to be dualized. The Lagrangian dual of (P2) is then

$$(\text{LD-2}) \qquad\qquad \phi^* = \max\{\phi(u), u \geq 0\}$$

where the corresponding subproblem for $u \geq 0$ is given by

$$\text{SP2}(u) \qquad\qquad \phi(u) = \quad \min \quad cx + u(b - Ax)$$
$$\text{s.t} \quad x \in B^n.$$

Note that, the objective function of the subproblem SP2(u) can be rewritten as

$$\sum_{j=1}^{n} (c_j - uA_j)x_j,$$

where A_j is the *j-th* column of the matrix A. Thus, SP2(u) can be solved by inspection:

$$x_j = \begin{cases} 1, & \text{if } c_j - uA_j < 0 \\ 0, & \text{otherwise} \end{cases}$$

This implies, for any $u \geq 0$

$$\phi(u) = ub + \sum_{j=1}^{n} \min\{c_j - uA_j, 0\}.$$

One specific instance is:

$$
\begin{aligned}
\min \quad & 3x_1 + 2x_2 \\
\text{s.t.} \quad & 2x_1 + 5x_2 \geq 3, \\
& 5x_1 + 2x_2 \geq 3, \\
& x_1, x_2 \in \{0, 1\}.
\end{aligned}
$$

This problem has only one feasible solution $x = (1, 1)$; and hence $x^* = (1, 1)$ is the unique optimal solution with $z^* = 5$.

For $u^1 = (1, 1)$,

$$\phi(u^1) = (1, 1) \begin{pmatrix} 3 \\ 3 \end{pmatrix} + \min\{3 - 7, 0\} + \min\{2 - 7, 0\} = -3.$$

For $u^2 = (0, 1)$,

$$\phi(u^2) = (0, 1) \begin{pmatrix} 3 \\ 3 \end{pmatrix} + \min\{3 - 5, 0\} + \min\{2 - 2, 0\} = 1,$$

and so on. These are lower bounds on z^*. The best of such bounds occur for $u^* = (4/21, 11/21)$ which solves the Lagrangian dual (we will discuss in a later section methods for determining such multipliers). At u^*,

$$\phi(u^*) = (4/21, 11/21) \begin{pmatrix} 3 \\ 3 \end{pmatrix} + \min\{3 - 63/21, 0\} + \min\{2 - 42/21, 0\} = 15/7.$$

On the other hand, solving the linear programming relaxation of this particular problem yields an LP relaxation solution $\bar{x} = (3/7, 3/7)$ with the optimal objective value $z_{LP}^* = 15/7$ which is equal to ϕ^*.

$\square$

The above examples, Example 2.1 and Example 2.2, show one case where the Lagrangian dual value is strictly greater than that of the LP relaxation and one where the two are equal. Indeed, Theorem 2.4 reveals a condition under which the bound obtained from the optimal value of LD is strictly better than that of the LP. In particular, if

$$\text{conv}(\mathbb{X}) \subsetneqq \{x \in \mathbb{R}^n_+ :\ Dx \geq d\},$$

then we can obtain $z^*_{LP} < \phi^*$. This is a useful result, say for instance, in the branch and bound procedure since such strictly better bounds can reduce substantially the size of the branch and bound tree to be searched. On the other hand, if $\text{conv}(\mathbb{X}) = \bar{\mathbb{X}} = \{x \in \mathbb{R}^n_+ :\ Dx \geq d\}$, then the subproblem $\text{SP}(u)$ can be solved by solving the LP relaxation. In this case, we say that $\mathbb{X}$ has the *Integrality Property*. The consequence is $\phi^* = z^*_{LP}$.

Theorem 2.6: **(Integrality Property)**
Let (IP), $\text{SP}(u)$, (LD), be as defined above. If $\mathbb{X}$ has the integrality property, then $\phi^* = z^*_{LP}$.

Proof: The proof follows immediately from Theorem 2.4 and the definition of the integrality property.   □

It should be emphasized that the integrality property is not defined relative to a given problem class but relative to a given integer programming formulation, in particular $\mathbb{X}$. This is important distinction because a problem often has more than one formulation or there can be different possible choices of constraints that will be kept in $\mathbb{X}$.

One of the question that we would like to answer is how one can solve the Lagrangian dual. The linear programming formulations appearing in Theorem 2.4, i.e., (2.8) or (2.9), may provide one way to calculate ϕ^*. Note that problem (2.9) is usually a large scale linear program because the number T of constraints

can easily be on the order of thousands or millions; one constraint corresponds to each feasible point in $\mathbb{X}$. The large number of constraints means that a constraint generation (or cutting plane) approach is required (see, [84] Chapter 8, for instance). Other possible technique of solving a large scale linear programming is the Dantzig-Wolfe decomposition method ([26], [55], [61], [11]). However, when the problem size is large (as typically the case) such simplex-based algorithms require excessive amounts of storage and are also quite slow. Hence the problems cannot be efficiently solved as an ordinary linear programming problem. Alternatively, we solve the Lagrangian dual using a technique known as *Subgradient Method* which will be described in Chapter 3. The method is designed to solve the problem of maximizing(minimizing) a non-differentiable concave (convex) function iteratively. It will be shown that the Lagrangian function which we would like to maximize is a non-differentiable concave function.

Definition 2.1: A function $f : \mathbb{R}^m \longrightarrow \mathbb{R}$ is said to be concave iff

$$f(\alpha x^1 + (1 - \alpha)x^2) \geq \alpha f(x^1) + (1 - \alpha)f(x^2)$$

for all $x^1, x^2 \in \mathbb{R}^m$ and all $\alpha \in [0, 1]$.

This suggests the following theorem.

Theorem 2.7: The dual function $\phi : \mathbb{R}^m \longrightarrow \mathbb{R}$, defined by

$$\phi(u) = \min \quad cx + u(b - Ax)$$
$$\text{s.t.} \quad x \in \mathbb{X}$$

is concave.

Proof:

Let $u_1, u_2 \in \mathbb{R}^m$ and $\alpha \in [0, 1]$. Then,

$$\phi(u_1) = \min \quad cx + u_1(b - Ax)$$
$$\text{s.t.} \quad x \in \mathbb{X}$$

and,

$$\phi(u_2) = \min \quad cx + u_2(b - Ax)$$
$$\text{s.t.} \quad x \in \mathbb{X}.$$

Now for $\bar{u} = \alpha u_1 + (1 - \alpha)u_2$,

$$\begin{aligned}
\phi(\bar{u}) = \quad & \min \quad cx + \bar{u}(b - Ax) \\
& \text{s.t.} \quad x \in \mathbb{X} \\
= \quad & c\bar{x} + \bar{u}(b - A\bar{x}) \qquad \text{for some } \bar{x} \in \mathbb{X}. \\
= \quad & \alpha \underbrace{[c\bar{x} + u_1(b - A\bar{x})]}_{\geq \phi(u_1)} + (1 - \alpha) \underbrace{[c\bar{x} + u_2(b - A\bar{x})]}_{\geq \phi(u_2)} \\
\geq \quad & \alpha\phi(u_1) + (1 - \alpha)\phi(u_2),
\end{aligned}$$

and this completes the proof. $\square$

Piecewise linearity is another property of the Lagrangian dual function. To see this, let $x^1, x^2, \ldots, x^T$ be the points of $\mathbb{X}$ as in the proof of Theorem 2.4 . Then,

$$\phi(u) = \min \quad cx + u(b - Ax) \quad = \quad \min \quad cx^i + u(b - Ax^i)$$
$$\text{s.t.} \quad x \in \mathbb{X} \qquad\qquad\qquad i \in \{1, 2, \ldots, T\}.$$

Therefore,

$$\phi(u) = \quad \min \quad cx^i + u(b - Ax^i) \qquad\qquad (2.15)$$
$$i \in \{1, 2, \ldots, T\} \ .$$

This relation, (2.15), tells us the exact structure of the dual function by describing it as the minimum of the set of linear functions. Hence the dual function is a piecewise linear concave function, see Figure 2.2 for $T = 6$ where a line i corresponds to the graph of $\theta = cx^i + u(b - Ax^i)$, for each $i = 1, 2, \ldots, 6$. Thus the Lagrangian dual problem $\phi^* = \max_{u \geq 0}\{\phi(u)\}$ can be viewed as the problem of maximizing a piecewise linear concave but non-differentiable function ϕ.

That is, the Lagrangian dual program consists of maximizing a concave function over a convex set; it is thus a "convex" programming problem which can be solved
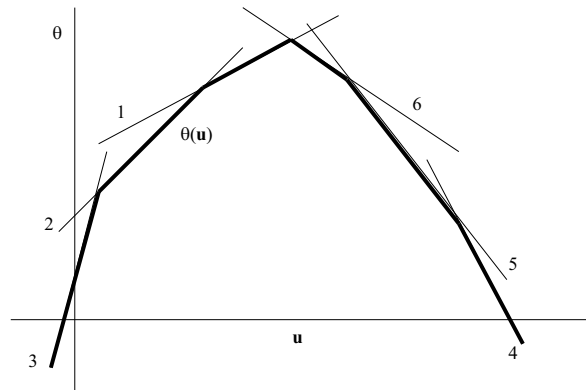
Fig. 2.2: Form of the dual function

iteratively.

Note that given $u \in \mathbb{R}_+^m$, the relaxation of the subproblem $\text{SP}(u)$ which may be given as

$$\phi(u) = \min \quad cx + u(b - Ax) \qquad (2.16)$$
$$\text{s.t.} \quad x \in \text{conv}(\mathbb{X})$$

is also a convex programming since $cx + u(b - Ax)$, for a given u, is affine and hence a convex function. Clearly $\text{conv}(\mathbb{X})$ is compact as $\mathbb{X}$ is a closed and bounded subset of $\mathbb{R}^n$. Hence, the solution set of (2.16) is compact, too. The non-empty, convex and compact solution set to the program (2.16) for a given $u \in \mathbb{R}_+^m$ is

$$\mathbb{X}(u) = \{\hat{x} \in \text{conv}(\mathbb{X}) : \ c\hat{x} + u(b - A\hat{x}) = \phi(u)\}. \qquad (2.17)$$

We show next that $\mathbb{X}(.)$ is a closed map, when a closed map is defined below. In defining this property we allow the point-to-set mapping to map points in one space $\mathbb{R}^m$ into subsets of another space $\mathbb{R}^n$.

A point-to-set map

$$M : \mathbb{R}^m \longrightarrow 2^{\mathbb{R}^n}$$

is called a *closed map* if $\{u^t\} \longrightarrow u, \quad x^t \in M(u^t)$ for all t, and $\{x^t\} \longrightarrow x$ imply that $x \in M(u)$.

The next theorem due to Larsson, Patriksson and Strömberg [54] shows that the mapping $u \longrightarrow \mathbb{X}(u)$, where the set $\mathbb{X}(u)$ is given by (2.17), is a closed map. We will use this result in Chapter 5 to analyze some properties of optimal solutions of the subproblem $\text{SP}(u)$.

Theorem 2.8: (Larsson-Patriksson-Strömberg, 1999)
Let the sequence $\{u^t\} \subseteq \mathbb{R}^m_+$, the map $\mathbb{X} : \mathbb{R}^m_+ \longrightarrow 2^{(\mathbb{R}^n)}$ be given by the definition (2.17), and the sequence $\{x^t\}$ by the inclusion $x^t \in \mathbb{X}(u^t)$. If $\{u^t\} \longrightarrow \bar{u}$ and $\{x^t\} \longrightarrow \bar{x}$ then $\bar{x} \in \mathbb{X}(\bar{u})$.

Proof: Given a sequence $x^t \in \mathbb{X}(u^t)$ such that $\{x^t\} \longrightarrow \bar{x}$ and $\{u^t\} \longrightarrow \bar{u}$, where $u^t \in \mathbb{R}^m_+$, we want to show that: (i) $\bar{x} \in \text{conv}(\mathbb{X})$ and (ii) $c\bar{x} + \bar{u}(b - A\bar{x}) = \phi(\bar{u})$. Note that (i) follows immediately since $x^t \in \text{conv}(\mathbb{X})$ and $\text{conv}(\mathbb{X})$ is closed (compact). Hence,

$$\bar{x} \in \text{conv}(\mathbb{X}). \tag{2.18}$$

Since the function $\mathcal{L}(x, u) = cx + u(b - Ax)$ is continuous on $\text{conv}(\mathbb{X}) \times \mathbb{R}^m_+$, it holds that $\mathcal{L}(x^t, u^t) \longrightarrow \mathcal{L}(\bar{x}, \bar{u})$ as $(x^t, u^t) \longrightarrow (\bar{x}, \bar{u})$. That is,

$$cx^t + u^t(b - Ax^t) \longrightarrow c\bar{x} + \bar{u}(b - A\bar{x}) \tag{2.19}$$

as $(x^t, u^t) \longrightarrow (\bar{x}, \bar{u})$. On the other hand, since $x^t \in \mathbb{X}(u^t)$ and the dual function ϕ is continuous it holds that

$$cx^t + u^t(b - Ax^t) = \phi(u^t) \longrightarrow \phi(\bar{u}). \tag{2.20}$$

Thus, from (2.18), (2.19) and (2.20) we have $\bar{x} \in \text{conv}(\mathbb{X})$ and $c\bar{x} + \bar{u}(b - A\bar{x}) = \phi(\bar{u})$. Consequently, $\bar{x} \in \mathbb{X}(\bar{u})$. $\qquad\square$

From Theorem 2.8 it follows that in a particular case when $\mathbb{X}(u) = \{\bar{x}\}$, a singleton, then $\{x^t\} \longrightarrow \bar{x}$. Consider now the Lagrangian dual of a convex programming

problem (2.8) in Theorem 2.4 where the constraints $Ax \geq b$ is to be dualized, as before. This can be written as :

$$(LD^c) \qquad \max_{u \geq 0} \min\{cx + u(b - Ax) :\ x \in \text{conv}(\mathbb{X})\} \qquad (2.21)$$

with a convex solution set Ω^*. Note that if the primal-dual optimality relation (strong duality) holds for this problem, then the optimal objective value of problem (2.21) is equal to ϕ^* (Theorem 2.4). To obtain primal-dual optimality relation, the primal feasible set must fulfil a constraint qualification.

Assumption (Slater constraint qualification):
The set $\{x \in \text{conv}(\mathbb{X}) :\ Ax > b\}$ is non-empty.

Under this assumption, the convex set Ω^* of solution of (LD^c) is non empty and compact and the strong duality holds for some pair $(\bar{x}, \bar{u}) \in \mathbb{R}^n_+ \times \mathbb{R}^m_+$ such that $b - A\bar{x} \leq 0$ holds ([12], Theorem 6.2.4). The next theorem states conditions under which a point x is optimal in (2.8) for the case that an optimal dual solution is at hand.

Theorem 2.9: **(Primal-dual optimality conditions)**
Let the assumption of the Slater constraint qualification holds and let $\bar{u} \in \Omega^*$. Then, $\bar{x}$ is a primal optimal solution of (2.8) if and only if $\bar{x} \in \mathbb{X}(\bar{u})$, $b - A\bar{x} \leq 0$ and $\bar{u}(b - A\bar{x}) = 0$.

The proof of this theorem follows from Theorem 2.2 and ([12], Theorem 6.2.5). $\square$

3. SUBGRADIENT OPTIMIZATION METHODS

This chapter deals with subgradient optimization methods and designs procedures that can be used to solve the Lagrangian dual of Integer Programming. In the proof of Theorem 2.4 we have seen that the Lagrangian dual problem can be formulated as a linear programming problem whose number of constraints are equal to the number of elements of the set $\mathbb{X}$. This makes the direct use of linear programming system impractical since in many application problems the number of elements in the set $\mathbb{X}$ can be very large and also can be very difficult to list all of them explicitly.

An alternative approach which is usually used to solve the Lagrangian dual problem without using a linear programming system is a subgradient optimization method. The subgradient optimization method that we would like to consider is an iterative procedure that can be used to solve the problem of maximizing a non-differentiable concave function $\phi(u)$ on a closed convex set Ω, i.e.,

$$\max\{\phi(u): \ u \in \Omega\},$$

using the following generic procedure:

- Choose an initial point $u^0 \in \Omega$.
- Construct a sequence of points $\{u^n\} \subseteq \Omega$ which eventually converges to an optimal solution using the rule

$$u^{n+1} = \mathrm{P}_\Omega(u^n + \lambda_n v^n)$$

  where $\mathrm{P}_\Omega(.)$ is a projection on the set Ω, $\lambda_n > 0$ is a positive scalar called *step length* and v^n is a vector, called *step direction*, which has to be determined at each iterate point.
- Until: (some stopping condition).

The direction of motion (step direction) that has to be determined at each iterate point in the procedure plays a crucial role in order to be able to obtain a desired outcome. Depending on a particular strategy for finding the direction of motion, the subgradient optimization methods can be categorized mainly into the pure subgradient, the deflected subgradient and the conditional subgradient methods.

The *pure subgradient method* uses a subgradient of the objective function at each iterate point as the stepping direction to generate a sequence of iterates. The procedure which is based on such stepping direction can, however, generate a sequence of iterates whereby the difference between a consecutive iterate points is insignificant since if the subgradient vector at a given iterate point form an obtuse angle with the previous direction of motion, then a zigzagging path is resulted. We call such phenomenon *zigzagging of kind I* (formal definition will be given in Section 3.2). Such zigzagging phenomenon that might manifest itself at any stage of the subgradient algorithm can cause a slow convergence of the procedure. As a tool to overcome this difficulty a deflection of subgradient procedure is used in which the direction of motion is computed by combining the current subgradient with the previous stepping direction. We may call such a strategy as the *deflected subgradient method* and it is the subject of discussion in Section 3.2. While the iterate points are generated by either the pure or deflected subgradient procedure, we need to project the resulting iterate point onto the feasible set in order to maintain feasibility. The operation of the projection can also hamper the motion from a given point to the next iterate point and forms also another type of zigzagging path if the selected direction of motion is almost parallel to the normal vector of a face of the feasible region that contains the given point since in such case the projection operator projects the iterate point $u^n + \lambda_n v^n$ back to a point near to u^n. We call such a phenomenon *zigzagging of kind II* (formal definition will be given in Section 3.3). The *conditional subgradient method* which defines the direction of motion as a combination of a subgradient and a vector from a normal cone at the given point helps us to handle such difficulties. The

conditional subgradient method will be discussed in Section 3.3 . We will also see that the phenomenon of zigzagging of kind II can manifest itself only in case the iterates are moving across the relative boundary of the feasible set.

3.1 The Pure Subgradient Method

3.1.1 Introduction

The maximum value of a smooth concave function can be usually determined by the gradient methods. A gradient method, say the steepest ascent method, finds an optimal solution of the problem $\max_x f(x)$ by iterative method in which, starting with some x^0, a sequence of x^n which eventually converges to an optimal solution is constructed according to the relation

$$x^{n+1} = x^n + \lambda_n \nabla f(x^n)$$

where $\lambda_n \geq 0$ is a suitable step length and $\nabla f(x^n)$ is the gradient vector of f at x^n. One may refer to [12] and [65] for complete coverage of this and related subjects.

In the case of our problem, however, the dual function is not differentiable. Hence, we cannot use the gradient method since there are points at which $\nabla \phi$ does not exist. Instead we use the subgradient method which is the adaption of the gradient method in which gradients are replaced by subgradients in order to make use of the concave structure of the dual function.

Definition 3.1: Let $f : \mathbb{R}^m \longrightarrow \mathbb{R}$ be concave. The vector $s \in \mathbb{R}^m$ is called a *subgradient* of f at $\bar{x} \in \mathbb{R}^m$ if

$$f(\bar{x}) + s(x - \bar{x}) \geq f(x) \quad \forall x \in \mathbb{R}^m.$$

Definition 3.2: The *subdifferential* of f at $\bar{x}$ is the set of all subgradients of f

at $\bar{x}$ which is given by

$$\partial f(\bar{x}) = \{s : \ f(\bar{x}) + s(x - \bar{x}) \geq f(x) \quad \forall x \in \mathbb{R}^m\}.$$

If $\partial f(\bar{x})$ is non-empty, then f is said to be subdifferentiable at $\bar{x}$. It is known that a concave function is subdifferentiable at every point in its domain. Furthermore, the subdifferential is a non-empty convex, closed and bounded set (see, for instance, Rockafellar [70] p. 217 or Dem'yanov and Vasil'ev [29] p. 49, Theorem 5.1). A concave function is not necessarily differentiable at all points in its domain. As will be shown in the following theorem, if a concave function is differentiable at a point $\bar{x}$ then $\nabla f(\bar{x})$ is a subgradient of f at the point. In this sense, subgradient is considered as a generalized gradient of a concave function.

Theorem 3.1: Let $f : \mathbb{R}^m \longrightarrow \mathbb{R}$ be concave and differentiable, and $\nabla f(\bar{x})$ be the gradient of f at $\bar{x}$. Then $\nabla f(\bar{x}) \in \partial f(\bar{x}) \quad \forall x \in \mathbb{R}^m$.

Proof: It suffices to show that for any $\bar{x} \in \mathbb{R}^m$,

$$\nabla f(\bar{x})(x - \bar{x}) \geq f(x) - f(\bar{x}) \quad \forall x \in \mathbb{R}^m.$$

For $x = \bar{x}$, this inequality holds obviously. So we need to consider only the case $x \neq \bar{x}$. Since f is differentiable, the directional derivative of f at $\bar{x}$ in the direction of $x - \bar{x}$, given by

$$\lim_{t \to 0^+} \frac{f(\bar{x} + t(x - \bar{x})) - f(\bar{x})}{t}$$

exists and is equal to $\nabla f(\bar{x})(x - \bar{x})$. Since f is concave, the following holds for $t \in (0, 1)$:

$$
\begin{aligned}
f(x) - f(\bar{x}) &= \frac{tf(x) + (1 - t)f(\bar{x}) - f(\bar{x})}{t} \\
&\leq \frac{f(tx + (1 - t)\bar{x}) - f(\bar{x})}{t} \\
&= \frac{f(\bar{x} + t(x - \bar{x})) - f(\bar{x})}{t}
\end{aligned}
$$

which implies

$$f(x) - f(\bar{x}) \ \leq \ \lim_{t \to 0^+} \frac{f(\bar{x} + t(x - \bar{x})) - f(\bar{x})}{t} = \nabla f(\bar{x})(x - \bar{x}).$$

This completes the proof. □

Note that, Definition 3.1 means that a subgradient vector is a gradient of a hyperplane supporting the *epigraph* of f at $(\bar{x}, f(\bar{x})) \in \mathbb{R}^{m+1}$, where epigraph of f is

$$epif := \{(x, z) \in \mathbb{R}^{m+1} : z \leq f(x)\},$$

which is a closed convex set. If the concave function f is also smooth at $\bar{x}$, then it is known that such a supporting hyperplane is uniquely determined by the gradient $\nabla f(\bar{x})$. This means that the subgradient of f at $\bar{x}$ is uniquely determined and given by $\nabla f(\bar{x})$. Thus we have the following theorem.

Theorem 3.2: If $\nabla f(\bar{x})$ exists, then $\partial f(\bar{x})$ is a singleton and $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$. □

However, at a point $\bar{x}$ where the function is non-differentiable we can have infinitely many elements in the subdifferential set $\partial f(\bar{x})$.

Example 3.1: Let

$$f(x) = \min_{x \in \mathbb{R}}\{3x, x + 2, \frac{-5}{3}x + 10\}.$$

Then f is a piecewise linear concave function given by

$$f(x) = \begin{cases} 3x, & x \leq 1 \\ x + 2, & 1 \leq x \leq 3 \\ \frac{-5}{3}x + 10, & x \geq 3. \end{cases}$$

(See Figure 3.1.)

f is differentiable at every point $\bar{x} \in \mathbb{R} \setminus \{1, 3\}$. Hence, for any $\bar{x} \notin \{1, 3\}$ the subgradient $s(\bar{x})$ of f at $\bar{x}$ is given by

$$s(\bar{x}) = f'(\bar{x}).$$
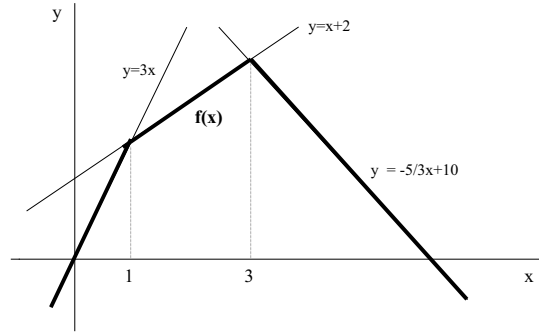
Fig. 3.1: Graph of the piecewise linear concave function f.

That is,

$$
s(\bar{x}) = \begin{cases} 3, & \bar{x} < 1 \\ 1, & 1 < \bar{x} < 3 \\ \frac{-5}{3}, & \bar{x} > 3 \end{cases}
$$

However, at $\bar{x} = 1$ both $s_1 = 3$ and $s_2 = 1$ are subgradients of f. Moreover, any convex combination of s_1 and s_2 is also a subgradient of f at $\bar{x} = 1$ as can be shown also in the following theorem. Similarly, both $s_2 = 1$ and $s_3 = \frac{-5}{3}$ as well as any of their convex combinations are the subgradients of f at $\bar{x} = 3$.

$\square$

Theorem 3.3: The subdifferential $\partial f(\bar{x})$ of f at $\bar{x} \in \mathbb{R}^m$ is a convex set.

Proof: Suppose $s_1, s_2 \in \partial f(\bar{x})$. Then it holds that

$$
s_1(x - \bar{x}) \geq f(x) - f(\bar{x}), \qquad \forall x \in \mathbb{R}^m
$$

and

$$
s_2(x - \bar{x}) \geq f(x) - f(\bar{x}), \qquad \forall x \in \mathbb{R}^m.
$$

So, for any $\alpha \in [0, 1]$ and $x \in \mathbb{R}^m$ we have,

$$
\begin{aligned}
[\alpha s_1 + (1 - \alpha)s_2](x - \bar{x}) = {} & \alpha s_1(x - \bar{x}) + (1 - \alpha)s_2(x - \bar{x}) \\
\geq {} & \alpha(f(x) - f(\bar{x})) + (1 - \alpha)(f(x) - f(\bar{x})) \\
= {} & f(x) - f(\bar{x})
\end{aligned}
$$

which, by the definition of a subgradient, implies

$$\alpha s_1 + (1 - \alpha)s_2 \in \partial f(\bar{x})$$

and this completes the proof. □

Theorem 3.4: A necessary and sufficient condition for $x^* \in \mathbb{R}^m$ to be a maximizer of a concave function f over $\mathbb{R}^m$ is $0 \in \partial f(x^*)$.

Proof: By the definition of the subgradient, $0 \in \partial f(x^*)$ for $x^* \in \mathbb{R}^m$ if and only if

$$f(x) - f(x^*) \leq 0(x - x^*) \qquad \forall x \in \mathbb{R}^m,$$

Which is equivalent to

$$f(x) \leq f(x^*) \qquad \forall x \in \mathbb{R}^n,$$

as claimed. □

Note that the condition "$0 \in \partial f(x^*)$" is a generalization of the usual stationary condition "$\nabla f(x) = 0$" of the smooth case. For the problem given in the Example 3.1, $x^* = 3$ is an optimal solution since $0 \in \partial f(3)$. Indeed, $0 = \frac{5}{8}s_2 + (1 - \frac{5}{8})s_3$ where $s_2 = 1$ and $s_3 = \frac{-5}{3}$ are the subgradients of f at $x^* = 3$. But, In general, it is difficult to construct a zero subgradient using convex combinations of the subgradients even if the point is an optimal solution since there is no general method that can be used to compute all the subgradients and the zero subgradient at the point.

3.1.2 The Pure Subgradient Algorithm

The subgradient procedure which is described in this section is an adaption of the gradient (steepest ascent) method of the smooth case and solves the problem of maximizing a nondifferentiable concave function. It is an iterative procedure which attempts to climb up the hill using the direction of the gradient vector at

each point where the gradient of the function exists; but replaces the gradient vector by a subgradient vector at a point where the gradient does not exist. That is, it starts at some point u^0 and construct a sequence of points $\{u^k\}$ according to the rule:

$$u^{k+1} = \mathrm{P}_\Omega(u^k + \lambda_k s^k), \quad k = 0, 1, 2, \ldots \quad (3.1)$$

where s^k is a subgradient of a concave function ϕ at a point u^k, $\lambda_k > 0$ is an appropriately chosen step length and $\mathrm{P}_\Omega(.)$ is the Euclidean projection on the feasible set Ω.

Beside the need for an appropriate termination criterion and a relevant rule to determine a suitable step length (step size) λ_k, two important requirements are desirable, from an implementation point of view, for the subgradient scheme. First, an easy method of computing subgradient vector $s^k \in \partial\phi(u^k)$ at every point $u^k \in \Omega$ must be available; and second, Ω must be simple enough to admit an easy projection.

The two requirements are fulfilled in the case of the Lagrangian dual problem (LD) of the linear integer programming problem since

- $\Omega = \mathbb{R}_+^m$ implies that the projection $\mathrm{P}_\Omega(u) = u_+$, where its components are defined by $(u_+)_i = \max\{0, u_i\}, \quad i = 1, 2, \ldots, m$.

- a subgradient vector can be determined easily using an optimal solution of the subproblem as will be shown in the next theorem, Theorem 3.5. In particular, we will show that given a point $u^k \in \mathbb{R}_+^m$, then $s^k = b - Ax^k \in \partial\phi(u^k)$ where x^k is an optimal solution of the corresponding subproblem and the dualized constraints are $Ax \geq b$.

Since a subgradient plays a central role in the course of maximizing the dual function, we give next a method to determine a subgradient of the dual function and will discuss the underlying reason to use the subgradient vector in the procedure

of maximizing a non-differentiable concave function.

In what follows, the function $\phi : \mathbb{R}^m \longrightarrow \mathbb{R}$ is the dual function given by

$$\phi(u) = \min\{cx + u(b - Ax) : x \in \mathbb{X}\}$$

unless stated otherwise, and given $\bar{u} \in \mathbb{R}^m_+$, the set $\mathbb{X}^*(\bar{u})$ denotes the set of optimal solution(s) of the subproblem

$$\phi(\bar{u}) = \min\{cx + \bar{u}(b - Ax) : x \in \mathbb{X}\}.$$

That is,

$$\mathbb{X}^*(\bar{u}) = \{\bar{x} \in \mathbb{X} : \ \phi(\bar{u}) = c\bar{x} + \bar{u}(b - A\bar{x})\}.$$

Theorem 3.5: Consider the dual function $\phi : \mathbb{R}^m \longrightarrow \mathbb{R}$. Then $s(\bar{x}) = b - A\bar{x}$ is a subgradient of ϕ at $\bar{u}$, where $\bar{x} \in \mathbb{X}^*(\bar{u})$.

Proof: Given $\bar{u} \in \mathbb{R}^m_+, \ \ \bar{x} \in \mathbb{X}^*(\bar{u})$ means that

$$\phi(\bar{u}) = c\bar{x} + \bar{u}(b - A\bar{x}),$$

and for any $u \in \mathbb{R}^m$,

$$\phi(u) = \min\{cx + u(b - Ax) : \ x \in \mathbb{X}\} \le c\bar{x} + u(b - A\bar{x}).$$

Thus it holds,

$$\begin{aligned} \phi(u) - \phi(\bar{u}) &\le \ c\bar{x} + u(b - A\bar{x}) - [c\bar{x} + \bar{u}(b - A\bar{x})] \\ &= \ (b - A\bar{x})(u - \bar{u}) \quad \forall u \in \mathbb{R}^m. \end{aligned}$$

This completes the proof. $\qquad\qquad\square$

In the following theorem, we use the relation

$$\phi(u) = \min\{cx^i + u(b - Ax^i)| \ i = 1, 2, \ldots, T\} \tag{3.2}$$

where $\{x^1, x^2, \ldots, x^T\} = \mathbb{X}$.

Theorem 3.6: Let $\phi(u)$ be the dual function given as (3.2) and
$\mathcal{I}(\bar{u}) = \{i : \phi(\bar{u}) = cx^i + \bar{u}(b - Ax^i)\}$. Then,

(a) $s^i = b - Ax^i$ is a subgradient of ϕ at $\bar{u}$ for all $i \in \mathcal{I}(\bar{u})$.

(b) If $i_1, i_2, \ldots, i_k \in \mathcal{I}(\bar{u})$, then

$$\bar{s} = \sum_{j=1}^{k} \alpha_j s^j$$

is also a subgradient of ϕ at $\bar{u}$, where

$$s^j = b - Ax^{i_j}, \quad \sum_{j=1}^{k} \alpha_j = 1, \quad \alpha_j \geq 0,$$

for $j = 1, 2, \ldots, k$.

Proof: (a) Follows directly from Theorem 3.5 .

(b) From (a), each of s^j for $j = 1, 2, \ldots, k$ is subgradient of ϕ at $\bar{u}$ and by Theorem 3.3 a convex combination of the subgradients is again a subgradient of the function at the given point. Hence also (b) holds. $\square$

At a given point, therefore, we have no unique subgradient of the function. This causes some difficulties with regard to a construction of a good iterative procedure that uses a subgradient vector as its stepping direction. In the smooth case, it is well known that the gradient vector is the local direction of maximum increase of the function. Unfortunately, this is not the case for a subgradient vector. In general, unlike the gradient direction of the smooth case, the subgradient is not an ascent direction and hence the iteration procedure of the subgradient method does not necessarily improve the objective function value at some steps. As a consequence, the procedures such as the line search techniques of the smooth case is not applicable to determine a suitable step length λ_k in the subgradient scheme. The following alternative point of view, however, provides an intuitive justification for moving in the direction of a subgradient:

Suppose $\hat{u}$ and $\bar{u}$ are any points such that $\phi(\hat{u}) \geq \phi(\bar{u})$ and let $\bar{s} \in \partial\phi(\bar{u})$. Then

$$\bar{s}(\hat{u} - \bar{u}) \geq \phi(\hat{u}) - \phi(\bar{u}) \geq 0,$$

where the first inequality is due to concavity of ϕ and the second follows from $\phi(\hat{u}) \geq \phi(\bar{u})$. Thus the hyperplane H= $\{(u, z) \in \mathbb{R}^{m+1} : z = \phi(\bar{u}) + \bar{s}(u - \bar{u})\}$ through $(\bar{u}, \phi(\bar{u}))$ having $\bar{s}$ as its normal determines two half-spaces and the closed half space into which $\bar{s}$ is pointing contains all $\hat{u}$ such that $\phi(\hat{u}) \geq \phi(\bar{u})$. That means if we are at the point $\bar{u}$ and want to increase ϕ, we should have to move to a point $\hat{u}$ with $\bar{s}(\hat{u} - \bar{u}) > 0$. This direction is given by the subgradient vector $\bar{s}$. In particular, this half space includes any point where $\phi(.)$ assumes its maximum value. In other word, the subgradient vector $\bar{s}$ at $\bar{u}$ forms an acute angle with the best direction leading from $\bar{u}$ to an optimal solution u^*, since

$$\bar{s}(u^* - \bar{u}) > 0.$$

Therefore, a sufficiently small step from $\bar{u}$ along the direction of $\bar{s}$ produces a point closer than $\bar{u}$ to any such maximum point (see Figure 3.2).
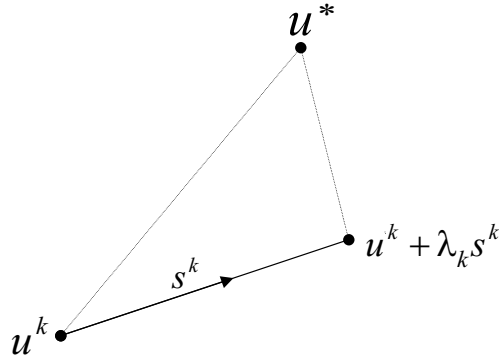

Fig. 3.2: For $s^k \in \partial\phi(u^k)$ and sufficiently small λ_k, $u^k + \lambda_k s^k$ is closer to an optimal u^* than u^k is.

That is, there exists $\hat{\lambda}_k$ such that for any $0 < \lambda_k < \hat{\lambda}_k$,

$$\|u^{k+1} - u^*\| < \|u^k - u^*\|,$$

where u^{k+1} is obtained by the subgradient scheme (3.1) and u^* is an optimal solution to $\max_{u \in \Omega} \phi(u)$ and $\|.\|$ is the Euclidean norm. The next theorem which is due to Polyak [68], justifies this and indicates the limits on the appropriate step sizes.

Theorem 3.7: Let u^* be an optimal solution to $\max_{x \in \Omega} \phi(x)$, where ϕ is concave over $\mathbb{R}^n$ and Ω is a closed convex subset of $\mathbb{R}^n$.
If

$$0 < \lambda_k < \frac{2(\phi(u^*) - \phi(u^k))}{\|s^k\|^2}, \tag{3.3}$$

then

$$\|u^{k+1} - u^*\| \;<\; \|u^k - u^*\|,$$

where u^{k+1}, u^k, s^k and λ_k are related as given in the equation (3.1) and $\|.\|$ denotes the Euclidean norm.

Proof:

$$
\begin{aligned}
\|u^{k+1} - u^*\|^2 =\;& \|\mathrm{P}_\Omega(u^k + \lambda_k s^k) - u^*\|^2 \\
\leq\;& \|u^k + \lambda_k s^k - u^*\|^2 \\
=\;& \|u^k - u^*\|^2 + 2\lambda_k s^k(u^k - u^*) + \lambda_k^2 \|s^k\|^2 \\
=\;& \|u^k - u^*\|^2 + \lambda_k[\lambda_k\|s^k\|^2 - 2[\underbrace{s^k(u^* - u^k)}_{\geq\; \phi(u^*) - \phi(u^k)}]] \\
\leq\;& \|u^k - u^*\|^2 + \lambda_k[\underbrace{\lambda_k\|s^k\|^2 - 2[\phi(u^*) - \phi(u^k)]}_{<\; 0, \;\; by \;\; (3.3)}] \\
<\;& \|u^k - u^*\|^2.
\end{aligned}
$$

Therefore, $\|u^{k+1} - u^*\| \;<\; \|u^k - u^*\|$. $\qquad\square$

The theoretical basis of the subgradient scheme lies on the above theorem, Theorem 3.7, since convergence analysis of the subgradient schemes based on the fact that the sequence $\{\|u^k - u^*\|\}$ is strictly decreasing in contrast to the gradient methods of the smooth case that rely on a monotonic decrease of $\{|\phi(u^k) - \phi(u^*)|\}$. Using this fact, Polyak [68] has also shown for a general convex programming

problem that the step rule given by (3.3) guarantees convergence. However, note that in order to apply the Polyak's step length rule specified in (3.3) one has to know the optimal objective value $\phi(u^*)$ a priori, which is impossible for most problems. Polyak suggested that $\phi(u^*)$ be replaced by a lower bound $\phi^L < \phi(u^*)$. He proved, in this case, that the sequence generated is such that $\phi(u^k) > \phi^L$ for some k or else $\phi(u^k) \leq \phi^L$ for all k and $\phi(u^k) \longrightarrow \phi^L$. In either case, however, we have no assurance of convergence to $\phi(u^*)$. In general, one needs to use rules based on a combination of theory, common sense and practical experimentation. The step size used most commonly in practice to solve the Lagrangian dual problem (LD) is

$$\lambda_k = \frac{\mu_k(UB - \phi(u^k))}{\|s^k\|^2} \tag{3.4}$$

where μ_k is a *step size parameter* satisfying $0 < \mu_k \leq 2$ and UB is an upper bound on the dual function ϕ, which may be obtained by applying a heuristic to the primal problem (IP). The empirical justification of this formula is given by Held, Wolfe and Crowder [46].

The step- size given by (3.4) is usually known as *relaxation step length* or also *Polyak's step length*. The step-size parameter μ_k controls the step size along the subgradient direction s^k. A first approach used by Held and Karp [45] and also recommended by Fisher [36] is to determine μ_k by setting $\mu_0 = 2$, and halving μ_k whenever $\phi(u^k)$ has failed to increase in some fixed number of iterations. However, Caprara et al. [22] observed that in some particular instance of problems the classical approach halves the step-size parameter after so many iterations, although in these iterations the growth of the value of the dual function ϕ is far from regular and can cause a slow convergence. In order to obtain a faster convergence, they proposed the following strategy: Start with $\mu_0 = 0.1$. For every $p = 20$ subgradient iterations compare the biggest and lowest values of ϕ computed on the last p iterations. If these two values differ by more than 1%, the current value of μ is halved. If, on the other hand, the two values are within 0.1% from each other, multiply the current value of μ by 1.5. Following this

idea one can determine the step-size parameters by setting $\mu_0 = 0.1$ and using the following rule to update μ_k, for $k = 1, 2, 3, \ldots$:

$$\mu_k = \begin{cases} (0.5)\mu_{k-1}, & \text{if } \bar{\phi} - \underline{\phi} > (0.01)\underline{\phi} \\ (1.5)\mu_{k-1}, & \text{if } \bar{\phi} - \underline{\phi} < (0.001)\underline{\phi} \\ \mu_{k-1}, & \text{otherwise} \end{cases}$$

where

$$\bar{\phi} = \max \left\{ \phi(u^t) \; : \; t = k - p + 1, \; k - p + 2, \; \ldots, \; k \right\}$$

and

$$\underline{\phi} = \min \left\{ \phi(u^t) \; : \; t = k - p + 1, \; k - p + 2, \; \ldots, \; k \right\}.$$

Despite its simplicity, the subgradient method gives rise to a number of problems regarding the choice of step lengths since the choice of the best step length for an implementation of the subgradient method is not yet well understood. The alternative and most general theoretical result is that $\phi(u^k) \longrightarrow \phi^*$ if the step lengths $\lambda_k > 0$ satisfy the following two conditions:

$$\lim_{k \to \infty} \lambda_k = 0, \quad \text{and} \quad \sum_{k=0}^{\infty} \lambda_k = \infty. \tag{3.5}$$

The next theorem justifies the convergence of the subgradient scheme with the step length given by (3.5).

Theorem 3.8: Consider the LD problem

$$\max\{\phi(u) : \; u \in \Omega = \mathbb{R}_+^m\}$$

where ϕ is the dual function and bounded from above on Ω so that the set $\Omega^* = \{\bar{u} \in \Omega : \; \phi(\bar{u}) \geq \phi(u), \; \forall u \in \Omega\} \neq \emptyset$. If $\{u^k\} \subseteq \Omega$ is a sequence of points generated by the recursive formula (3.1) where $s^k \in \partial\phi(u^k)$ is given by $s^k = b - Ax^k$ for some $x^k \in \mathbb{X}^*(u^k)$ and λ_k are positive quantities satisfying condition (3.5), then $\phi(u^k) \longrightarrow \phi^* = \phi(u^*)$, where $u^* \in \Omega^*$.

Proof: This result may be considered as a special case of more general statement given by Polyak [67] for the solution of extremum problems. To prove the

theorem, we need to show that for any arbitrary $\epsilon > 0$, $\exists K_0 > 0$ such that $k \geq K_0 \Rightarrow \phi(u^*) - \phi(u^k) < \epsilon$. Let us suppose by contradiction that $\exists \epsilon > 0$ such that

$$\phi(u^*) - \phi(u^k) \geq \epsilon \quad \forall k. \tag{3.6}$$

Suppose $x^k \in \mathbb{X}^*(u^k)$ so that $s^k = b - Ax^k \in \partial\phi(u^k)$ for each k. By definition:

$$\phi(u^k) + s^k(u - u^k) \geq \phi(u) \quad \forall u$$

and hence setting $u = u^*$ we obtain from (3.6) that

$$s^k(u^* - u^k) \geq \epsilon.$$

Now multiply the last result by the negative quantity $(-2\lambda_k)$, we have

$$2\lambda_k s^k(u^k - u^*) \leq -2\lambda_k\epsilon.$$

It follows that

$$
\begin{aligned}
\|u^{k+1} - u^*\|^2 &= \|P_\Omega(u^k + \lambda_k s^k) - u^*\|^2 \\
&\leq \|u^k + \lambda_k s^k - u^*\|^2 \\
&= \|u^k - u^*\|^2 + \lambda_k^2\|s^k\|^2 + 2\lambda_k s^k(u^k - u^*) \\
&\leq \|u^k - u^*\|^2 + \lambda_k^2\|s^k\|^2 - 2\lambda_k\epsilon.
\end{aligned}
$$

Now recalling that $\mathbb{X}$ is a finite set that can be given as $\mathbb{X} = \{x^t : t = 1, 2, \ldots, T\}$, let

$$\|s^*\|^2 = \max\{\|s^t\|^2 : \ s^t = b - Ax^t, t = 1, 2, \ldots, T\}.$$

Under the condition of the first part of (3.5) a $K_1 > 0$ can always be found such that

$$\lambda_k \leq \frac{\epsilon}{\|s^*\|^2} \quad \forall k \geq K_1$$

or equivalently:

$$\lambda_k\|s^*\|^2 \leq \epsilon.$$

Then we can write :

$$
\begin{aligned}
\|u^{k+1} - u^*\|^2 &\leq \|u^k - u^*\|^2 + \lambda_k(\lambda_k\|s^*\|^2 - 2\epsilon) \\
&\leq \|u^k - u^*\|^2 - \lambda_k\epsilon, \quad \forall k \geq K_1
\end{aligned}
$$

This last inequality, recursively written, gives for any arbitrary integer $N > K_1$:

$$0 \leq \|u^{N+1} - u^*\|^2 \leq \|u^{K_1} - u^*\|^2 - \epsilon \sum_{k=K_1}^{N} \lambda_k \tag{3.7}$$

Since, by (3.5), $\sum_{k=K_1}^{N} \lambda_k \to \infty$ as $N \to \infty$, the right hand side of (3.7) tends to $-\infty$; which is a contradiction. $\qquad\square$

The choice of the step size λ_k according to the rule (3.5) is also subjected to some criticisms with regard to the rate of convergence of the subgradient procedure. Some comments and numerical experiments in the literature (for instance,[45], [46], [73], [36]) show that such a choice of step length is, in general, inefficient for a practical application due to the resulting slow convergence. Hence the choice of step size according to the step relaxation rule, (3.4) is still popular in practical implementation of the subgradient procedure.

The following algorithm can use any subgradient at each step, but for computational purpose one of the subgradient directions $b - Ax^i$ will be chosen, where x^i is a solution of the corresponding subproblem at the *i-th* iteration.

Algorithm 3.1: **The Subgradient Algorithm for the Lagrangian Dual**

Step 0: (Initialization) Choose a starting point $u^0 \geq 0$ and let $k = 1$.

Step 1: Determine a subgradient vector s^k at u^k by solving the subproblem $\mathrm{SP}(u^k)$:

$$\phi(u^k) = \min \quad cx + u^k(b - Ax)$$
$$\text{s.t.} \quad x \in \mathbb{X}$$

Let x^k be a solution of this subproblem. Then,
$s^k = b - Ax^k$.

Step 2: (Feasibility and Optimality Test)

If $s^k \leq 0$, then x^k is an ϵ-optimal solution to the primal problem with $\epsilon = |u^k s^k|$.

If $s^k \leq 0$ and $u^k s^k = 0$, then x^k is an optimal solution of the primal problem and u^k is an optimal solution of the Lagrangian dual problem. STOP. Otherwise go to Step 3.

Step 3: Let $u^{k+1} = P_{\mathbb{R}_+^m}(u^k + \lambda_k s^k)$, where

$$P_{\mathbb{R}_+^m}(u) = \bar{u} \text{ for which its } \textit{i-th} \text{ component } \bar{u}_i = \begin{cases} u_i, & \text{if } u_i \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

and $\lambda_k \geq 0$ is a step length given by (3.4).

Let $k = k + 1$, and return to step 1.

Ideally the subgradient algorithm can be stopped when, on some iterate k, we find a subgradient which is a zero vector. However, in practise this can rarely happen since the algorithm just chooses one subgradient s^k and has no way of showing $0 \in \partial\phi(u^k)$ as a convex combination of subgradients. The stopping criteria stated in the Step 2 , i.e., $s^k \leq 0$ and $u^k s^k = 0$, can happen only if the strong duality holds(Theorem 2.2). But this is not generally possible for integer programming problems. Hence the typical stopping rule is either to stop after a sufficiently large but fixed number of iterations or to stop if the value of the function has not increased (by at least a certain amount) with in a given number of iterations.

3.2   The Deflected Subgradient Method

One of the most important behavior of the subgradient procedure is that at each of the iterate u^k, the subgradient direction s^k forms an acute angle with the direction leading from u^k to the optimal solution u^*. However, according to various reports (see for instance, Camerini et. al [21], Bazaraa et. al [12], Sherali and Ulular [77]), as the iterates progress the angle between the subgradient direction s^k can form an obtuse angle with the previous direction of motion s^{k-1} and this can force the next iterate point to become near to the previous one. This phenomenon can obviously slow the convergence of the procedure. The following figure, Figure 3.3, attempts to illustrate such a behavior in a two-dimensional case.
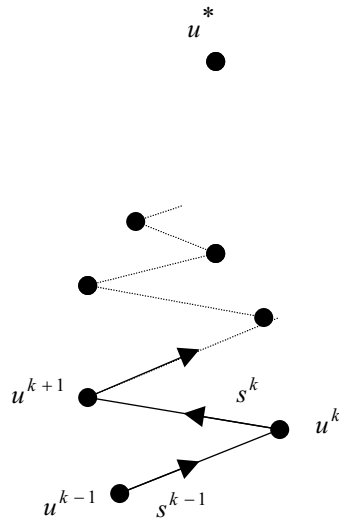

Fig. 3.3: Zigzagging of kind I in the pure subgradient procedure.

Definition 3.3: Let λ_n be a positive scalar and $d^n \in \mathbb{R}^m$. We say that an iterative procedure

$$u^{n+1} = \mathrm{P}_\Omega(u^n + \lambda_n d^n), \quad n = 0, 1, 2, \dots$$

forms a ***zigzagging of kind I*** if at any two (or more) consecutive iterate

points $u^k, u^{k+1} \in \Omega$, the angle between corresponding step directions d^k and d^{k+1} is obtuse; i.e., $d^k d^{k+1} < 0$.

Zigzagging of kind I is peculiar to the pure subgradient procedure. For instance, consider the special case of a piecewise linear ϕ on $\Omega = \mathbb{R}_+^m$:

$$\phi(u) = \min \ \{c_i + A^i u \ : 1 \leq i \leq T \ \}$$

where $A^i \in \mathbb{R}^m$ and $c_i \in \mathbb{R}^1$. Then, the problem is

$$\max \quad \phi(u)$$
$$s.t. \quad u \in \mathbb{R}_+^m.$$

Now consider dividing $\mathbb{R}_+^m$ into T subregions $\Omega_1, \Omega_2, \ldots, \Omega_T$, where $\Omega_i = \{u \in \mathbb{R}_+^m : \phi(u) = c_i + A^i u\}$. Note that $A^i \in \partial \phi(u)$ at any point $u \in \Omega_i$; and A^i is the only subgradient of ϕ on the interior of Ω_i since ϕ is differentiable on the interior of Ω_i with gradient A^i. Thus, if the procedure step from a region Ω_i into another region Ω_j by moving along the step direction A^i, the (sub)gradient A^j of ϕ in the new region may form an obtuse angle with A^i and points back into the region we just left. Figure 3.4 indicates a case in which the procedure will zigzag back and forth across the line of intersection of different regions.

Such zigzagging phenomena that might manifest itself at any stage of the subgradient procedure slow down the search process. In order to avoid such an unpleasant behavior one may need to deflect the subgradient direction whenever it forms an obtuse angle with the previous stepping direction. To this end, in order to form a smaller angle between the current stepping direction and the preceding direction than the traditional (pure) subgradient direction does, and hence to enhance the speed of convergence, Camerini et. al, [21] proposed a modification of the pure subgradient method in which the subgradient direction s^k at an iterate u^k is replaced by a *deflected subgradient* direction d^k, given by
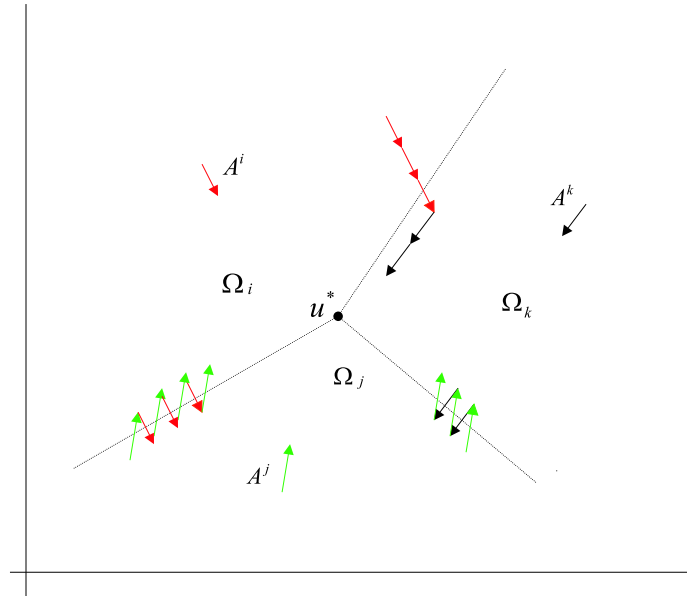
$$d^k = s^k + \delta_k d^{k-1}$$

Fig. 3.4: Zigzagging of kind I across the line of intersection of Ω_i and Ω_j as well as Ω_j and Ω_k by moving along subgradients.

where $s^k \in \partial\phi(u^k)$ and $\delta_k \geq 0$ is a suitable scalar called a *deflection parameter*, and $d^{k-1} = 0$ for $k = 0$. That is, the deflected subgradient method moves in the current search direction d^k which is a linear combination of the current subgradient direction and the direction used at the previous step.

In this section we consider the deflected subgradient method and show that any favorable property of the subgradient vector can be extended to the deflected subgradient direction while this method generates a point which forms a more acute angle with the direction to the optimal solution set than the point generated by the pure subgradient method and can also reduce the unfavorable zigzagging behavior of the pure subgradient method.

Algorithm 3.2: **The deflected subgradient algorithm**

Step 0: (Initialization):

Choose a starting point $u^0 \in \Omega = \mathbb{R}^m_+$, and let $k = 0, \;\; d^{k-1} = 0$.

Step 1: Determine a subgradient $s^k \in \partial\phi(u^k)$

$$d^k = s^k + \delta_k d^{k-1} \tag{3.8}$$

$$u^{k+1} = P_\Omega(u^k + \lambda_k d^k) \tag{3.9}$$

(Rules to determine δ_k and λ_k will be given.)

$k = k + 1$

Step 2: If a stopping condition is not yet hold, return to step 1.

We next consider some properties of the deflected subgradient directions and a rule to determine the deflection parameter δ_k and step length λ_k given in the equations (3.8) and (3.9) respectively. To that end, we will consider the following lemma whose result will be used in the proof of the next theorem.

Lemma 3.9: Suppose Ω is a closed convex subset of $\mathbb{R}^m$, $u^0 \in \Omega$, and $v = u^0 + \lambda d$ where d is a vector in $\mathbb{R}^m$ and λ is a positive scalar. If $u^1 = P_\Omega(v)$ and $p = u^1 - v$, then

(i) $pd \leq 0$.

(ii) $\|u^1 - u^0\| \leq \|v - u^0\|$.

Proof: The results of this lemma follows directly from the properties of convex set, in particular, the fact that the vector p is perpendicular to the supporting hyperplane of Ω at u^1 and hence the angle at u^1 of the resulting triangle $\triangle(u^0 u^1 v)$ is obtuse (see Figure 3.5). $\qquad\square$

Theorem 3.10: Suppose $s^k \in \partial\phi(u^k)$ and d^k is given by (3.8). If

$$0 < \lambda_k < \frac{\phi^* - \phi(u^k)}{\|d^k\|^2}, \qquad \forall k = 0, 1, 2, ... \tag{3.10}$$

Then,

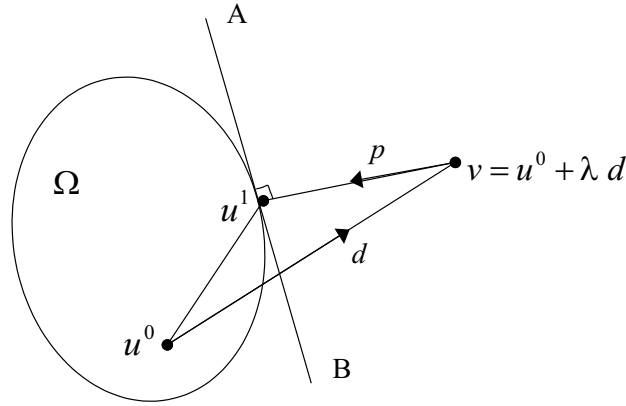$$d^k(u^* - u^k) \geq s^k(u^* - u^k) \tag{3.11}$$

for all k.

Fig. 3.5: Illustration of Lemma 3.9 in 2D.

Proof: We shall prove the theorem by induction on k. Clearly (3.11) is valid for $k = 0$ with an equal sign. Suppose that the assertion of the theorem is true for some $m = k$. To prove it for $m = k + 1$, observe that by (3.8)

$$d^{k+1}(u^* - u^{k+1}) = s^{k+1}(u^* - u^{k+1}) + \delta_{k+1}d^k(u^* - u^{k+1}) \tag{3.12}$$

Hence we need only to show $\delta_{k+1}d^k(u^* - u^{k+1}) \geq 0$.

Now setting $p^k = P_\Omega(u^k + \lambda_k d^k) - (u^k + \lambda_k d^k)$ we have

$$
\begin{aligned}
d^k(u^* - u^{k+1}) &= d^k(u^* - P_\Omega(u^k + \lambda_k d^k)) \\
&= d^k(u^* - u^k - \lambda_k d^k - p^k) \\
&= d^k(u^* - u^k) - \lambda_k\|d^k\|^2 - d^k p^k \\
&\geq d^k(u^* - u^k) - \lambda_k\|d^k\|^2 \tag{3.13}
\end{aligned}
$$

The inequality in the last expression follows from Lemma 3.9(i), i.e., $-p^k d^k \geq 0$. On the other hand, using the given condition in the theorem and the fact that $s^k \in \partial\phi(u^k)$, we get

$$0 < \lambda_k\|d^k\|^2 \leq \phi^* - \phi(u^k) \leq s^k(u^* - u^k) \leq d^k(u^* - u^k).$$

The last inequality follows from the induction hypothesis. Hence it holds that

$$d^k(u^* - u^k) - \lambda_k\|d^k\|^2 \geq 0.$$

This together with (3.13) yield

$$\delta_{k+1}d^k(u^* - u^{k+1}) \geq 0$$

since $\delta_{k+1} \geq 0$. From this last relation and (3.12), it follows that

$$d^{k+1}(u^* - u^{k+1}) \geq s^{k+1}(u^* - u^{k+1})$$

and this completes the proof. $\square$

The following theorem extends two important properties of a subgradient vector s^k to properties of the deflected subgradient direction d^k.

Theorem 3.11: Let $\{u^k\}$ be the sequence of iterates generated by the deflected subgradient scheme. Under the condition of the preceding theorem,

(i) $d^k(u^* - u^k) > 0$

(ii) $\|u^{k+1} - u^*\| < \|u^k - u^*\|$

for all k where u^k are non optimal points and u^* is an optimal solution.

Proof: (i) Since $s^k(u^* - u^k) \geq \phi^* - \phi^k > 0$, the claim follows directly from Theorem 3.10 .

(ii) Now to prove the second part of the theorem:

$$
\begin{aligned}
\|u^* - u^{k+1}\|^2 &= \|u^* - \mathrm{P}_\Omega(u^k + \lambda_k d^k)\|^2 \\
&\leq \|u^* - u^k - \lambda_k d^k\|^2 \\
&= \|u^* - u^k\|^2 + \lambda_k[\lambda_k\|d^k\|^2 - 2d^k(u^* - u^k)] \qquad (3.14)
\end{aligned}
$$

Then, by applying the condition in Theorem 3.10 we get

$$\lambda_k\|d^k\|^2 \leq \phi^* - \phi(u^k) < 2(\phi^* - \phi(u^k)).$$

Now using this inequality, concavity of the function ϕ, and Theorem 3.10 we obtain the following relations, respectively:

$$\lambda_k\|d^k\|^2 < 2(\phi^* - \phi(u^k)) \leq 2s^k(u^* - u^k) \leq 2d^k(u^* - u^k).$$

From this it follows,

$$\lambda_k \|d^k\|^2 - 2d^k(u^* - u^k) < 0.$$

This together with (3.14) provides

$$\|u^* - u^{k+1}\| < \|u^* - u^k\|,$$

as claimed. □

Theorem 3.11(i) tells us that, like the case of the subgradient vector, also the deflected subgradient direction d^k forms an acute angle at each iterate points with the direction leading to an optimal point u^*. Moreover, the second part of Theorem 3.11 guarantees that the sequence $\{\|u^k - u^*\|\}$ is strictly decreasing and as a result, a point closer and closer to an optimal solution is obtained at each iteration of the deflected subgradient procedure. The following theorem shows that with a particular choice of the deflection parameter δ_k one can also obtain a stronger result.

Theorem 3.12: Suppose $\{u^k\}$ is the sequence of iterates generated by the deflected subgradient procedure. Under the condition of Theorem 3.10, let

$$\delta_k = \begin{cases} -\tau_k \frac{s^k d^{k-1}}{\|d^{k-1}\|^2}, & \text{if } s^k d^{k-1} < 0 \\ 0, & \text{otherwise} \end{cases} \qquad (3.15)$$

with $0 \le \tau_k < 2$. Then,

(i)
$$\frac{d^k(u^* - u^k)}{\|d^k\|} \ge \frac{s^k(u^* - u^k)}{\|s^k\|}. \qquad (3.16)$$

(ii) If the vectors d^k and s^k form an angle θ_d^k and θ_s^k, respectively, with the vector $u^* - u^k$, then

$$0 \le \theta_d^k \le \theta_s^k \le 90^o.$$

Proof: (i) If $s^k d^{k-1} \ge 0$ then $\delta_k = 0$ and hence (3.16) holds obviously with equality as $d^k = s^k$. Consider the case $s^k d^{k-1} < 0$, in which case,

$$\delta_k = -\tau_k \frac{s^k d^{k-1}}{\|d^{k-1}\|^2} \qquad (3.17)$$

Then,

$$
\begin{aligned}
\|d^k\|^2 - \|s^k\|^2 &= \|s^k + \delta_k d^{k-1}\|^2 - \|s^k\|^2 \\
&= \delta_k(2s^k d^{k-1} + \delta_k\|d^{k-1}\|^2) \\
&= \delta_k(2s^k d^{k-1} - \tau_k s^k d^{k-1}), \qquad \text{(from (3.17))} \\
&= \delta_k(2 - \tau_k)s^k d^{k-1} \\
&< 0.
\end{aligned}
$$

The last inequality follows from the given condition on τ_k, i.e., $2 - \tau_k > 0$ and $s^k d^{k-1} < 0$. Hence,

$$
\|d^k\|^2 \quad < \quad \|s^k\|^2. \tag{3.18}
$$

Thus, (3.18) together with Theorem 3.10 yield

$$
\frac{d^k(u^* - u^k)}{\|d^k\|} \geq \frac{s^k(u^* - u^k)}{\|s^k\|},
$$

as claimed.

(ii) Now to prove the second part of the theorem: Clearly both θ_d^k and θ_s^k are acute angles and since,

$$
\cos(\theta_d^k) = \frac{d^k(u^* - u^k)}{\|d^k\|\|u^* - u^k\|} \;, \qquad \cos(\theta_s^k) = \frac{s^k(u^* - u^k)}{\|s^k\|\|u^* - u^k\|} \;,
$$

from the first part of the theorem it follows that

$$
\cos(\theta_d^k) \geq \cos(\theta_s^k).
$$

Thus, $\theta_d^k \leq \theta_s^k$ as the angles are acute and cosine function is monotonic decreasing on the interval $[0, 90^o]$. $\qquad\qquad\square$

Note that the choice of the deflection parameter δ_k according to the rule (3.15) of Theorem 3.12 tends to avoid zigzagging of kind I of the sequence of iterates since whenever the actual subgradient direction s^k forms an obtuse angle with the preceding moving direction d^{k-1}, the deflection parameter δ_k is set greater than zero to deflect the "unfavorable" direction. Diagram (b) in Figure 3.6 illustrates such a behavior in a two-dimensional case while diagram (a) illustrates the case
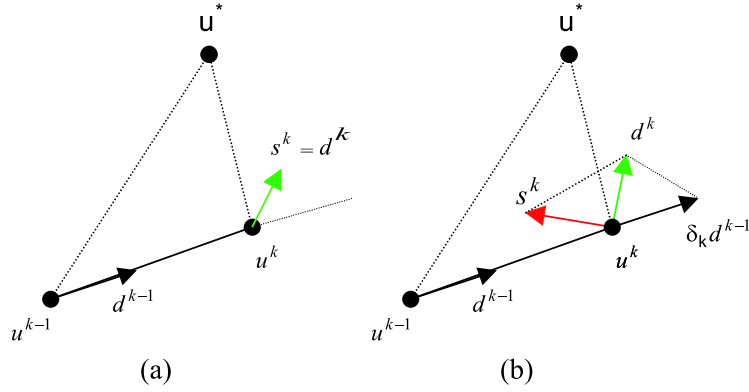
(a)   (b)

Fig. 3.6: (a) Case where s^k is a favorable stepping direction. (b) Case where s^k is deflected since it has formed an obtuse angle with d^{k-1}.

where the current subgradient direction is favorable.

From Theorem 3.12(ii) one can also observe that by a proper choice of the deflection parameter δ_k, the deflected subgradient vector direction d^k is always at least as good as the direction of the subgradient vector s^k in a sense that the stepping direction d^k can form a more acute angle with the best direction towards an optimal solution than the pure subgradient vector does, thus enhancing the speed of convergence. By imposing a lower limit on τ_k, as given in the next theorem, we can obtain the following useful result.

Theorem 3.13: Let d^k be the deflected subgradient direction at k-th iteration. Under the condition of Theorem 3.12, if $\tau_k \geq 1$, then

$$d^k d^{k-1} \geq 0.$$

Proof: Case $s^k d^{k-1} \geq 0 \ \Rightarrow \ d^k = s^k$ and hence the claim follows. Thus, consider the case $s^k d^{k-1} < 0$. In this case we have

$$
\begin{aligned}
d^k d^{k-1} = \ & (s^k + \delta_k d^{k-1}) d^{k-1} \\
= \ & s^k d^{k-1} - \tau_k s^k d^{k-1}, \quad \text{(followed from: } \delta_k = -\tau_k s^k d^{k-1}/\|d^{k-1}\|^2) \\
= \ & (1 - \tau_k) s^k d^{k-1} \\
\geq \ & 0,
\end{aligned}
$$

because $1 - \tau_k \leq 0$. This completes the prove. $\qquad\square$

The importance of Theorem 3.13 lies on the fact that choosing the deflection parameter δ_k using the strategy (3.15) with $1 < \tau_k < 2$ forces the current deflected subgradient direction to form always an acute angle with the previous step direction and hence this method eliminates the zigzagging of kind I of the pure subgradient procedure.

Note that the choice of $\tau_k = 1$ would amount to using a direction orthogonal to d^{k-1}. In [21], the use of $\tau_k = 1.5$ is recommended and its intuitive justification together with computational results are also given, which is indicating that also in practise the performance of deflected subgradient algorithm is superior to that of the pure subgradient algorithm. There are, in fact, various forms of the choices of the deflection parameter in literature (for instance, [77], [18]) other than the one proposed by Camerini, Fratta and Maffioli which is discussed above. Note that the direction of motion generated by the deflected subgradient procedure with the Camerini-Fratta-Maffioli deflection strategy (3.15) may turns out to be simply the subgradient direction itself, say, in case the pure subgradient procedure is free of zigzagging of kind I, since the deflection is initiated only when the current subgradient forms an obtuse angle with the previous direction of motion. Sherali and Ulular [77] recommend to make the deflection at each iterate point by choosing the direction of motion which bisects the angle between the current subgradient s^k and the previous direction of motion d^{k-1} irrespective of the type of angle between s^k and d^{k-1}. To get this direction, the deflection parameter is computed according to

$$\delta_k = \frac{\|s^k\|}{\|d^{k-1}\|}.$$

With this choice of the deflection parameter the direction $d^k = s^k + \delta_k d^{k-1}$ simply bisects the angle between the subgradient s^k and the previous direction d^{k-1} and, in this sense, is called an *average direction strategy*.

Furthermore, instead of choosing just the bisecting direction, a convex combination of the subgradient s^k and the previous direction of motion d^{k-1} is some times used to determine a current step direction d^k (see, e.g., [18]). That is,

$$d^k = \alpha_k s^k + (1 - \alpha_k) d^{k-1}, \quad \alpha_k \in (0, 1).$$

In this case, if $s^k d^{k-1} \geq 0$ then $d^k d^{k-1} \geq 0$ for any $\alpha_k > 0$. However, if $s^k d^{k-1} < 0$ then we need to restrict α_k in order to eliminate zigzagging of kind I. In particular, if $s^k d^{k-1} < 0$, then $d^k d^{k-1} \geq 0$ if and only if

$$0 \leq \alpha_k \leq \frac{\|d^{k-1}\|}{\|d^{k-1}\|^2 - s^k d^{k-1}}.$$

Hence, in if $s^k d^{k-1} < 0$, then α_k should be chosen so as $0 < \alpha_k < \hat{\alpha}_k$, where

$$\hat{\alpha}_k = \min \left\{ 1, \frac{\|d^{k-1}\|}{\|d^{k-1}\|^2 - s^k d^{k-1}} \right\}$$

in order to eliminate zigzagging of kind I.

3.3 Conditional Subgradient Method

Conditional subgradient method is another iterative procedure that can handle
a zigzagging of kind II of the pure subgradient method which cannot be elimi-
nated by the deflected subgradient method. Recall that the deflected subgradient
method can control a zigzagging phenomenon only in the case that the zigzagging
path would be created due to the formation of obtuse angle between the current
subgradient direction and the previous direction of motion. In some case, how-
ever, although the angle between the subgradient direction s^k and the previous
direction of motion is acute, it is possible that s^k is almost perpendicular to a
face of feasible region containing u^k in which case the projection of $u^k + \lambda_k s^k$ to
the feasible region yields a point near to u^k, or essentially u^k itself (see Example
3.2). In such situation, the iterate points may be almost unaltered or have no
significant change as a consequence of which the procedure may become very
slow. The following problem illustrates the possibly slow convergence of pure
(and deflected) subgradient methods.

Example 3.2: *(Slow convergence of subgradient method)*

$$\begin{aligned} \max \quad & -u_1 + 2u_2 \\ \text{s.t.} \quad & (u_1, u_2) \in \Omega \end{aligned}$$

where $\Omega = \{(u_1, u_2) \in \mathbb{R}^2 : u_1 - u_2 \geq 0, \ 0 \leq u_1 \leq 1, \ 0 \leq u_2 \leq 1 \}$.

The optimal solution is $u^* = (1, 1)$ with optimal objective value $\phi^* = 1$ where
$\phi(u) = -u_1 + 2u_2$. At any iterate point $u^k \in \Omega$, the subgradient $s^k = (-1, 2)$
forms an acute angle with the direction leading to the optimal solution u^* and
never form an obtuse angle with a preceding stepping direction. Indeed, the
stepping direction is always the direction of the subgradient vector (-1,2). Hence
the step direction of the deflected subgradient methods with either the deflection
rule (3.15), the average direction strategy, or a convex combination of current

subgradient and previous direction always coincide with the step direction of the pure subgradient method. Therefore, solving the problem with the pure subgradient method starting at the point $u^0 = (0,0)$ with the relaxation step length and step length parameter $\mu_k = 1$, yields $\lambda_k = \frac{\phi^* - \phi(u^k)}{\|s^k\|^2} = \frac{1 - \phi(u^k)}{5}$, for all k. Then, the iterates are

$$u^k = P_\Omega(u^{k-1} + \frac{1 - \phi(u^{k-1})}{5}(-1, 2)) = (1 - \frac{9^k}{10^k}, 1 - \frac{9^k}{10^k})$$

with objective value $\phi(u^k) = 1 - \frac{9^k}{10^k}$, for all k, (see Figure 3.7). Clearly, for this instance, the procedure converges very slowly toward the optimal solution. □
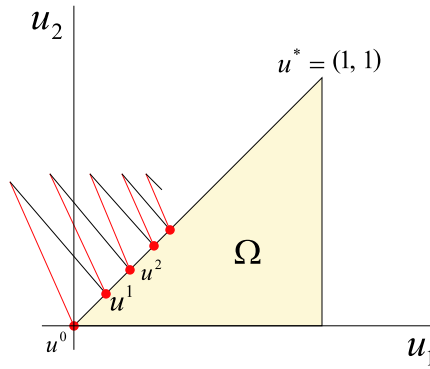
Fig. 3.7: Iteration points of Example 3.2: Zigzagging of kind II

The zigzagging phenomenon in this example, Example 3.2, is due to the fact that the subgradients are "almost perpendicular" to a face of the feasible set $\{(u_1, u_2) : u_1 = u_2\}$. We call such a zigzagging of subgradient method *zigzagging of kind II.* We will give a formal definition of such a zigzag after the following basic definitions.

A *normal cone* of Ω at some point $u \in \Omega$ is the set

$$N_\Omega(u) = \{y \in \mathbb{R}^m : y(z - u) \leq 0, \quad \forall z \in \Omega \}.$$

A *tangent cone* of the set Ω at some $u \in \Omega$ is the set

$$T_\Omega(u) = \{z \in \mathbb{R}^m : zy \leq 0, \quad \forall y \in N_\Omega(u) \}.$$

$N_\Omega(u)$ and $T_\Omega(u)$ are both non empty closed convex subsets containing 0. Figure 3.8 demonstrates the normal and the tangent cone at a $u \in \Omega$.

Fig. 3.8: A normal and tangent cone of a convex set at u.

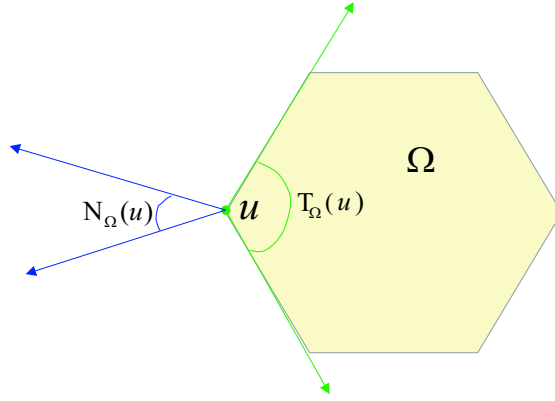If $u \in \text{int}(\Omega)$, where $\text{int}(\Omega)$ denotes the interior of Ω, then $N_\Omega(u) = \{0\}$ and $T_\Omega(u) = \mathbb{R}^m$. The elements of $N_\Omega(u)$ and $T_\Omega(u)$ are called *normal vectors* and *tangent vectors* of the set Ω at u, respectively. Note that for any $z \in \Omega$, we have $z - u \in T_\Omega(u)$ since $y(z - u) \leq 0$ for any $y \in N_\Omega(u)$. i.e, $\Omega - \{u\} \subseteq T_\Omega(u)$. Indeed, the definition of tangent cone can be also expressed (see, for instance, [47], Definition III. 5.1.1) as the closure of the cone generated by $\Omega - \{u\}$. i.e.,

$$T_\Omega(u) = \text{cl}\{z \in \mathbb{R}^m : \ z = \alpha(y - u), \ y \in \Omega, \ \alpha > 0 \ \}.$$

Or equivalently (see, for instance, [60], Theorem 2.2.7)

$$T_\Omega(u) = \text{cl}\{z \in \mathbb{R}^m : \ \text{there exists} \ \ \lambda > 0 \ \text{so that} \ u + \lambda z \in \Omega \ \}.$$

Note that if Ω is a polyhedral set, then

$$T_\Omega(u) = \{z \in \mathbb{R}^m : \ u + \lambda z \in \Omega \ \text{for some} \ \lambda > 0\}.$$

We refer to a vector $d \in \mathbb{R}^m$ as an *infeasible direction* at $u \in \text{bd}(\Omega)$ if $d \notin T_\Omega(u)$, where $\text{bd}(\Omega)$ is the set of points on a boundary of Ω. That is, a vector d is

infeasible direction at $u \in \mathrm{bd}(\Omega)$ if

$$dv > 0 \quad \text{fo some} \quad v \in \mathrm{N}_{\Omega}(u).$$

Now we give a formal definition of zigzagging of kind II.

Definition 3.4: Let λ_n be a positive scalar and $d^n \in \mathbb{R}^m$. We say that an iterative procedure

$$u^{n+1} = \mathrm{P}_{\Omega}(u^n + \lambda_n d^n), \quad n = 0, 1, 2 \ldots$$

forms a ***zigzagging of kind II*** if at any two (or more) consecutive iterate points $u^k, u^{k+1} \in \Omega$, there exist vectors $v \in \mathrm{N}_{\Omega}(u^k)$ and $w \in \mathrm{N}_{\Omega}(u^{k+1})$ such that

$$d^k v > 0 \quad \text{and} \quad d^{k+1} w > 0. \tag{3.19}$$

Note that zigzagging of kind II can arise, according to the definition, only when the iterate points are on the boundary of Ω, otherwise there exist no vector v and w satisfying the given condition (3.19). In this section we generalize the pure subgradient method in the sense that feasible set is taken into consideration while a step direction is determined and establish the convergence of the resulting conditional subgradient method. The conditional subgradient method, which is presented here, is shown by Larsson et, al. [53] to have significantly better practical performances than that of the pure subgradient method since it can avoid zigzagging of kind II as we will see in this Section.

We would like to first consider the general case. Let the function $f : \mathbb{R}^m \longrightarrow \mathbb{R}$ be concave. Thus f is continuous but not necessarily everywhere differentiable. Further, let $\Omega \subseteq \mathbb{R}^n$ be a non-empty, closed and convex set, and assume that $f^* = \sup_{u \in \Omega} f(u) = f(u^*) < \infty$ for some $u^* \in \Omega$. The problem considered as

$$\text{(DP)} \qquad\qquad f^* = \max_{u \in \Omega} f(u) \tag{3.20}$$

with the non-empty and convex set of optimal solutions

$$\Omega^* = \{\bar{u} \in \Omega : \ f(\bar{u}) = f^*\}.$$

The stated property of the problem (DP) are assumed to hold throughout our discussion unless stated otherwise. Note that the definition of the subdifferential and subgradient do not take the feasible set Ω into consideration. Dem'yanov and Shomesova ([27], [28]) generalize the definitions of the subdifferential and subgradient so that the feasible set of the problem (DP) is taken into account.

Definition: Let $f : \mathbb{R}^m \longrightarrow \mathbb{R}$ be a concave function and $\Omega \subseteq \mathbb{R}^m$. The *conditional subdifferential* of f at $u \in \Omega$ is the set

$$\partial^\Omega f(u) = \{\tilde{s} \in \mathbb{R}^m : \ f(u) + \tilde{s}(z - u) \geq f(z), \ \ \forall z \in \Omega\}.$$

The element $\tilde{s} \in \partial^\Omega f(u)$ is called a *conditional subgradient* of f at u.

We will see that subdifferential and conditional subdifferential are identical on the relative interior of a set Ω while this is not the case on the relative boundary of Ω. In the following figure, Figure 3.9, the normal vector $\tilde{s}_a$ of line AB is a conditional subgradient of a smooth concave function f on $\mathbb{R}$ whose graph is given in the figure where the set of interest is the closed interval $\Omega = [a, b]$. Note that the graph has only one subgradient vector, say, s_a at the point a which is normal to the tangent line CD while it possesses many other conditional subgradients including s_a. Similarly the normal vector $\tilde{s}_b$ of the line EF is one of the conditional subgradients of f at b. Observe that conditional subdifferential is depending not only on the function but also on the set of interest Ω.

It is known that, given a concave function f and u in its domain, $\partial f(u)$ is a non-empty, convex and compact set (see, e.g., Rockafellar [70]). Clearly $\partial f(u) \subseteq \partial^\Omega f(u)$ for all $u \in \Omega$ and hence $\partial^\Omega f(u)$ is non-empty. The following theorem is also immediate from the definition of conditional subdifferential.

Theorem 3.14: The conditional subdifferential $\partial^\Omega f(u)$ is non empty, closed and convex for all $u \in \Omega$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

The optimality conditions of the problem (DP) is given in the next theorem.
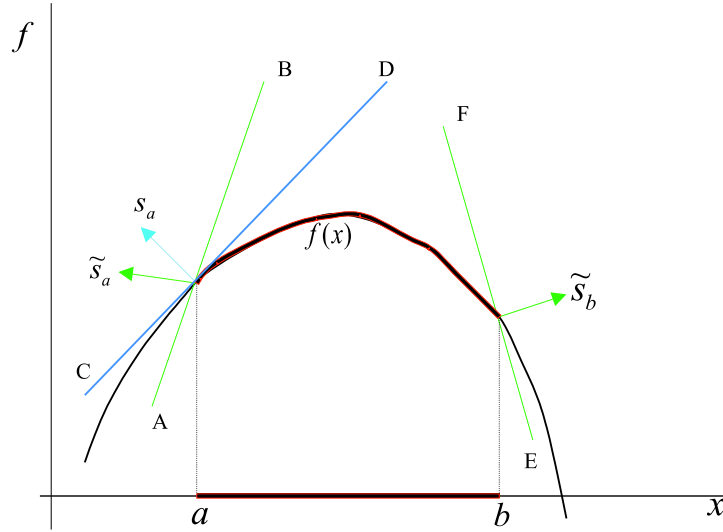
Fig. 3.9: $\tilde{s}_a$ and $\tilde{s}_b$ are conditional subgradients of f at a and b, respectively, where $\Omega = [a, b]$.

Theorem 3.15: **(Optimality Conditions)**

a) $\bar{u} \in \Omega^*$ if and only if $0 \in \partial^\Omega f(\bar{u})$.

b) $\bar{u} \in \Omega^*$ if and only if $\partial f(\bar{u}) \cap N_\Omega(\bar{u}) \neq \emptyset$.

Proof: (a) follows easily from the definition of conditional subdifferential and the fact that f is concave. We prove, therefore, only (b). Let $\bar{u} \in \Omega^*$; and suppose $\partial f(\bar{u}) \cap N_\Omega(\bar{u}) = \emptyset$. This implies $s(u - \bar{u}) > 0$ for all $s \in \partial f(\bar{u})$ and some $u \in \Omega$. Since $0 \in \partial f(\bar{u})$, it follows $0(u - \bar{u}) > 0$ which is a contradiction. Hence, $\partial f(\bar{u}) \cap N_\Omega(\bar{u}) \neq \emptyset$.

On the other hand, suppose $\partial f(\bar{u}) \cap N_\Omega(\bar{u}) \neq \emptyset$.   This means, there exists $s \in \partial f(\bar{u})$ and $s(u - \bar{u}) \leq 0$ for all $u \in \Omega$. Then, by the definition of conditional subgradient, since $s \in \partial f(\bar{u}) \subseteq \partial^\Omega f(\bar{u})$, it follows $f(u) \leq f(\bar{u}) + s(u - \bar{u}) \leq f(\bar{u})$ for all $u \in \Omega$. This completes the proof.   □

Note that if Ω is a polyhedral set, i.e., $\Omega = \{u \in \mathbb{R}^m : A^i u \leq b_i, \; i = 1, 2, \ldots, n\}$ where $A^i \in \mathbb{R}^m$ is a row vector , then

$$\mathrm{N}_\Omega(u) = \{v \in \mathbb{R}^m : \; v = \sum_{i=1}^n w_i A^i, \; \sum_{i=1}^n w_i(A^i u - b_i) = 0, \; w_i \geq 0, \; \forall i\}$$

and defining an index set for the active (binding) constraints at u by

$$\mathcal{I}(u) = \{i \in \{1, 2, \ldots, n\} : \; A^i u = b_i\},$$

the necessary and sufficient conditions for the optimality of $\bar{u}$ in the problem (DP) can be expressed as:

$$\exists s \in \partial f(\bar{u}) \text{ and } w_i \geq 0, \; \; i \in \mathcal{I}(\bar{u}) \text{ such that } s = \sum_{i \in \mathcal{I}(\bar{u})} w_i A^i.$$

That is, s lies in the cone generated by the gradients of binding constraints at $\bar{u}$, which is the generalization of the Karush-Kuhn-Tucker condition of a differentiable programming.

Theorem 3.16: (Characterization of a conditional subdifferential)

$$\partial^\Omega f(u) = \partial f(u) - \mathrm{N}_\Omega(u)$$

for each $u \in \Omega$.

Proof: Suppose $v \in \partial^\Omega f(u)$, where $u \in \Omega$ is fixed but arbitrary. Hence,

$$f(z) - f(u) \leq v(z - u), \quad \forall z \in \Omega.$$

Define an auxiliary function $h : \; \Omega \longrightarrow \mathbb{R}$ by

$$h(z) = f(z) - f(u) - v(z - u).$$

Clearly h is concave on Ω, $\quad h(z) \leq 0$ for all $z \in \Omega$ and $h(u) = 0$. Hence, u is a maximum point for h on the set Ω. Thus, by the optimality condition, $\partial h(u) \cap \mathrm{N}_\Omega(u) \neq \emptyset$. Moreover, from the definition of h, $\partial h(u) = \partial f(u) - \{v\}$.

Hence, there exists a vector $s_u \in \partial f(u)$ such that $s_u - v \in N_\Omega(u)$. That is,

$n_u = s_u - v$, for some $n_u \in N_\Omega(u)$

or

$v = s_u - n_u \in \partial f(u) - N_\Omega(u)$.

Thus,

$\partial^\Omega f(u) \subseteq \partial f(u) - N_\Omega(u)$.

On the other hand, suppose

$v = v_1 - v_2 \in (\partial f(u) - N_\Omega(u)), \quad \text{where} \quad v_1 \in \partial f(u) \quad \text{and} \quad v_2 \in N_\Omega(u)$.

Thus we have,

$f(z) - f(u) \leq v_1(z - u) \quad$, for all $z \in \mathbb{R}^m \quad$ since $v_1 \in \partial f(u) \quad$ and

$v_2(z - u) \leq 0 \quad$ for all $z \in \Omega \quad$ since $v_2 \in N_\Omega(u)$.

Summing the two inequalities, we obtain

$f(z) - f(u) \leq (v_1 - v_2)(z - u) = v(z - u) \quad$ for all $z \in \Omega$

which means $v = v_1 - v_2 \in \partial^\Omega f(u)$.

That is, $(\partial f(u) - N_\Omega(u)) \subseteq \partial^\Omega f(u) \quad$ and this completes the proof. $\qquad \square$

3.3.1 Conditional Subgradient Algorithm

Let $\tilde{s} \in \partial^\Omega f(u^k)$ be a conditional subgradient of f at $u^k \in \Omega$. A conditional subgradient optimization is a procedure which, starting at a given $u^0 \in \Omega$, generates a sequence of iterates $\{u^k\}$ for the problem (DP), (3.20), with the rule

$$u^{k+1} = P_\Omega(u^k + \lambda_k \tilde{s}^k), \quad k = 0, 1, 2, ... \tag{3.21}$$

where $\tilde{s}^k = s^k - v^k$, with $s^k \in \partial f(u^k)$, $v^k \in N_\Omega(u^k)$ and $\lambda_k > 0$ is a step length to be chosen according to a rule which guarantees convergence.

We will see that while the conditional subgradient procedure can alleviate some of the drawbacks of the pure and deflected subgradient procedures, it preserves their two important properties; namely if $\{u^k\} \subseteq \Omega$ is a sequence of iterates

generated by the conditional subgradient scheme (3.21), $\tilde{s}^k \in \partial^\Omega f(u^k)$, and u^* is an optimal solution then

(i) $\tilde{s}^k(u^* - u^k) > 0$ for all non-optimal u^k, since from the definition of conditional subgradient we have

$$\tilde{s}^k(u^* - u^k) \geq f(u^*) - f(u^k) > 0. \tag{3.22}$$

Therefore, like a subgradient vector also a conditional subgradient form an acute angle with the direction leading to an optimal solution.

(ii)
$$0 < \lambda_k < \frac{2(f(u^*) - f(u^k))}{\|\tilde{s}^k\|^2} \quad \Rightarrow \quad \|u^{k+1} - u^*\| < \|u^k - u^*\|. \tag{3.23}$$

i.e., the sequence $\{\|u^k - u^*\|\}$ is strictly decreasing. The justification of this result is similar to the proof of Theorem 3.7 except replacing s^k by $\tilde{s}^k$.

The next theorem establishes convergence of the sequence of iterates u^k that generated by the conditional subgradient procedure (3.21) by imposing a condition on the choice of the step length λ_k.

Theorem 3.17: Let $u^k \subseteq \Omega$ be a sequence of iterates generated by the conditional subgradient procedure (3.21) applied to the problem (DP) (3.20) with the step length $\lambda_k > 0$, for all $k = 0, 1, 2, \ldots$ that also fulfilling

$$\lim_{k\to\infty} \lambda_k = 0, \quad \sum_{k=0}^{\infty} \lambda_k = \infty, \quad \& \quad \sum_{k=0}^{\infty} \lambda_k^2 < \infty. \tag{3.24}$$

If $\sup_k \{\|\tilde{s}^k\|\} < \infty$, then $\{u^k\}$ converges to an element of Ω^*.

Proof: Let $u^* \in \Omega^*$ and $k \geq 1$. In every iteration k we then have

$$
\begin{aligned}
\|u^{k+1} - u^*\|^2 &= \|P_\Omega(u^k + \lambda_k \tilde{s}^k) - u^*\|^2 \\
&\leq \|u^k + \lambda_k \tilde{s}^k - u^*\|^2 \\
&= \|u^k - u^*\|^2 + 2\lambda_k \tilde{s}^k(u^k - u^*) + \lambda_k^2 \|\tilde{s}^k\|^2
\end{aligned} \tag{3.25}
$$

The repeated application of (3.25) yields that

$$\|u^k - u^*\|^2 \leq \|u^0 - u^*\|^2 + 2\sum_{j=0}^{k-1} \lambda_j \tilde{s}^j(u^j - u^*) + \sum_{j=0}^{k-1} \lambda_j^2 \|\tilde{s}^j\|^2 \qquad (3.26)$$

Then from (3.22) we have

$$\tilde{s}^j(u^j - u^*) \leq 0, \qquad \forall j \geq 0 \qquad (3.27)$$

Hence, from (3.26) and (3.27), we obtain

$$\|u^k - u^*\|^2 \leq \|u^0 - u^*\|^2 + \sum_{j=0}^{k-1} \lambda_j^2 \|\tilde{s}^j\|^2. \qquad (3.28)$$

Defining $c = \sup_j \{\|\tilde{s}^j\|\}$ and $p = \sum_{j=0}^{\infty} \lambda_j^2$, we obtain

$$\|\tilde{s}^j\| \leq c \text{ for any } j \geq 0 \text{ and } \sum_{j=0}^{k-1} \lambda_j^2 < p.$$

From (3.28) we then conclude that,

$$\|u^k - u^*\|^2 \leq \|u^0 - u^*\|^2 + pc^2 \quad \text{for any} \quad k \geq 1,$$

which means that the sequence $\{u^k - u^*\}$ is bounded and, therefore, the sequence $\{u^k\}$ is bounded, too.

Assume now that there is no subsequence $\{u^t\}$ of $\{u^k\}$ with $\{\tilde{s}^t(u^t - u^*)\} \longrightarrow 0$. Then, there must exist a $\delta > 0$ with $\{\tilde{s}^k(u^k - u^*)\} \leq -\delta$ for all $k \geq K$, where K is a sufficiently large natural number, since by (3.27) the sequence is non-positive. This together with the condition $\sum_{j=0}^{\infty} \lambda_j = \infty$ imply that

$$\lim_{k \to \infty} \sum_{j=0}^{k-1} \lambda_j \tilde{s}^j(u^j - u^*) = -\infty.$$

Moreover,

$$\lim_{k \to \infty} \sum_{j=0}^{k-1} \lambda_j^2 \|\tilde{s}^j\|^2 \leq c^2 \lim_{k \to \infty} \sum_{j=0}^{k-1} \lambda_j^2 < \infty.$$

From these and (3.26) it follows that $\{\|u^k - u^*\|\} \longrightarrow -\infty$, which is impossible. The sequence $\{u^k\}$ must, therefore, contain a subsequence $\{u^t\}$ such that

$\{\tilde{s}^t(u^t - u^*)\} \longrightarrow 0$. From (3.22) it follows that $\{f(u^t)\} \longrightarrow f^*$. Moreover, since $\{u^k\}$ is bounded, there exists an accumulation point of the sequence u^t, say $\bar{u}$. From the continuity of f it follows that $f(\bar{u}) = f^*$ and hence $\bar{u} \in \Omega^*$. Now to show that the whole sequence $\{u^k\}$ converges to $\bar{u}$: Let $\epsilon > 0$ and find an $N = N(\epsilon)$ such that $\|u^N - \bar{u}\| < \frac{\epsilon}{2}$ and $\sum_{t=N}^{\infty} \lambda_t^2 < \frac{\epsilon}{2c^2}$. Then for any $k > N$, analogously to the derivation of (3.28), we obtain

$$\|u^k - \bar{u}\|^2 \le \|u^N - \bar{u}\|^2 + \sum_{j=N}^{k-1} \lambda_j^2 \|\tilde{s}^j\|^2 \; < \; \frac{\epsilon}{2} + \frac{\epsilon}{2c^2} c^2 = \epsilon.$$

Since this holds for any arbitrary small value $\epsilon > 0$, the claim of the theorem follows. $\qquad \square$

A step length satisfying the conditions of (3.24) of Theorem 3.17 is called divergent step length. If the Polyak's step length λ_k satisfying the condition of (3.23) would be chosen and $\sup_k \{\|\tilde{s}^k\|\} < \infty$ then it also holds that $f(u^k) \longrightarrow f^*$. Justification is similar to the proof Theorem 3.8 . Thus for any accumulation point $\bar{u}$ of $\{u^k\}$ we have $\bar{u} \in \Omega^*$ which follows from continuity of f. Moreover the existence of an accumulation point is guaranteed by (3.23) since $\{\|u^k - u^*\|\}$ is strictly decreasing implies that $\{u^k\}$ is bounded.

Boundedness of the sequence $\{\tilde{s}^k\}$ of conditional subgradients can be ensured by appropriate choices of v^k in the method (3.21). One possible way is to choose

$$v^k = \mathrm{P}_{\mathrm{N}_\Omega(u^k)}(s^k), \tag{3.29}$$

where $s^k \in \partial f(u^k)$. Since $\partial f(u^k)$ is a bounded convex set and a v^k chosen according to (3.29) is also bounded, it follows that the sequence of $\tilde{s}^k = s^k - v^k$ is bounded, too.

Corollary 3.18: (**Adaptive step length selection rule**)

Let η_k be an arbitrary sequence of step lengths replacing λ_k in the method (3.21). If there exists sequences $\{\underline{\lambda}_k\}$ and $\{\bar{\lambda}_k\}$ that both satisfy (3.24) and $\underline{\lambda}_k \le \eta_k \le \bar{\lambda}_k$ for all k, then the assertion of Theorem 3.17 holds. $\qquad \square$

The proof of Corollary 3.18 is immediate since the sequence $\{\eta_k\}$ under the given conditions satisfy (3.24). Since the elements of the sequence $\{\underline{\lambda}_k\}$ and $\{\bar{\lambda}_k\}$ can be made arbitrarily small and large, respectively while satisfying (3.24), for example,

$$\underline{\lambda}_k = \frac{\alpha}{\beta + k} \quad \& \quad \bar{\lambda}_k = \frac{M}{\beta + k}, \tag{3.30}$$

where $\alpha > 0$ is as small as needed, $M > 0$ is as large as required and $\beta > 0$, the condition of the Corollary 3.18 helps us to be flexible in step length selections.

The next example illustrates the effect of applying the conditional subgradient procedure to solve the problem of Example 3.2.

Example 3.3: (Enhanced convergence using the conditional subgradient)
To show the effect of the conditional subgradient method, we apply the method (3.21) using the Polyak's step length rule to the instance of Example 3.2, starting with $u^0 = (0,0)$ and with $\mu_0 = 1$. Note that the normal cone at u^0 is $N_\Omega(u^0) = \{y \in \mathbb{R}^2 : y_1 + y_2 \leq 0, \ y_1 \leq 0 \}$ and the projection of $s^0 = (-1, 2)$ onto the normal cone is $v^0 = (-3/2, 3/2)$. Hence, a conditional subgradient at u^0 is given by $\tilde{s}^0 = s^0 - v^0 = (1/2, 1/2)$. This implies $\lambda_0 = 1/\|\tilde{s}^0\|^2 = 2$. Thus, $u^1 = P_\Omega(u^0 + \lambda_0 \tilde{s}^0) = P_\Omega((1, 1)) = (1, 1)$; i.e., the optimal solution is reached in one iteration. $\qquad \square$

Note that the efficiency of the conditional subgradient method depends on a chosen conditional subgradient direction and may not eliminate zigzagging of kind II for arbitrarily chosen conditional subgradient. For instance, in the example above (-1,1) is also in $N_\Omega(u^0)$ and hence $\tilde{s}^0 = (-1, 2) - (-1, 1) = (0, 1)$ is a conditional subgradient at u^0 but it is an infeasible direction.

3.3.2 Conditional Subgradient Procedure for the Lagrangian Dual

In this section we apply the conditional subgradient procedure to solve the Lagrangian dual problem 2.3. Recall that the Lagrangian dual problem (LD) is given by

$$\phi^* = \max \ \phi(u)$$
$$u \in \Omega = \{u \in \mathbb{R}^m : \ u \geq 0\} = \mathbb{R}^m_+$$

where $\phi(u)$ is given by the subproblem SP(u):

$$\phi(u) = \min \ cx + u(b - Ax)$$
$$\text{s.t.} \ \ x \in \mathbb{X}$$

and $\mathbb{X}^*(u) = \{\bar{x} \in \mathbb{X} : \ \phi(u) = c\bar{x} + u(b - A\bar{x})\}$, the set of optimal solutions of the subproblem for a given $u \in \mathbb{R}^m_+$.

Note that the normal cone to the set $\Omega = \mathbb{R}^m_+$ at $u \in \mathbb{R}^m_+$ is given by

$$N_{\mathbb{R}^m_+}(u) = \{v \in \mathbb{R}^m : \ v \leq 0, \ v_i u_i = 0, \ \ i = 1, 2, \ldots, m\}.$$

Notation: Let $N_+(.) := N_{\mathbb{R}^m_+}(.)$, for notational convenience .

Then, the conditional subgradient of ϕ at each iterate point $\{u^k\}$ can be given by

$$\tilde{s}^k = s^k - v^k$$

where,

$$s^k = b - Ax^k \in \partial\phi(u^k), \quad \text{for} \ \ x^k \in X^*(u^k)$$

and $\quad v^k = P_{N_+}(s^k).$

Consequently, each of the *i-th* component of v^k is given by

$$v_i^k = \begin{cases} s_i^k, & \text{if} \ \ s_i^k \leq 0 \ \ \text{and} \ \ u_i^k = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Thus, denoting the *i-th* row of the coefficient matrix A by A^i, the *i-th* component of the conditional subgradient of ϕ at $u^k \in \mathbb{R}^m_+$ is given by

$$\tilde{s}^k_i = s^k_i - v^k_i = \begin{cases} 0, & \text{if } b_i - A^i x^k \leq 0 \text{ and } u^k_i = 0 \\ b_i - A^i x^k, & \text{otherwise} \end{cases} \tag{3.31}$$

where $x^k \in \mathbb{X}^*(u^k)$, which is an optimal solution of the subproblem $\text{SP}(u^k)$. Moreover, for any $u \in \mathbb{R}^m$,

$$\text{P}_\Omega(u) = \text{P}_{\mathbb{R}^m_+}(u) = u^+, \tag{3.32}$$

where the *i-th* component of u^+ is given by $u^+_i = \max\{u_i, 0\}$.

Thus, using the above results on conditional subgradient of the dual function ϕ, the corresponding conditional subgradient can be constructed as follows.

Algorithm 3.3: **Conditional Subgradient Algorithm for the Lagrangian Dual Problem:**

Step 0. Initialization:

Choose a starting point $u^0 \in \mathbb{R}^m_+$, and set $k = 0$

Step 1. Determine a conditional subgradient of ϕ at u^k:

$\tilde{s}^k = s^k - v^k$, where its *i-th* component $s^k_i - v^k_i$ is determined by (3.31).

Step 2. $u^{k+1} = \text{P}_{\mathbb{R}^m_+}(u^k + \lambda_k(s^k - v^k))$

where λ_k is chosen according to (3.24) or Corollary 3.18, and $\text{P}_{(\mathbb{R}^m_+)}(.)$ is given by (3.32).

Step 3. If a stopping condition is not yet satisfied,

let $k = k + 1$ and go to Step 1.

The convergence of the Algorithm 3.3 follows from Theorem 3.17. Note that if $v^k = 0$, then Algorithm 3.3 will reduce to the pure subgradient method. The

important property of the conditional subgradient method that enhance its performance than that of the pure subgradient is the fact that the conditional subgradient vector $s^k - v^k \geq 0$ at each iterate u^k if u^k is on the boundary of $\Omega = \mathbb{R}^m_+$. This means, for any $u^k \in \mathbb{R}^m_+$, the step direction $\tilde{s}^k = s^k - v^k$ which is determined by (3.31) is feasible in the program of (LD) and hence $u^k + \lambda_k(s^k - v^k) \in \mathbb{R}^m_+$ for a small step length λ_k and this implies $u^{k+1} = \mathrm{P}_{\mathbb{R}^m_+}(u^k + \lambda_k(s^k - v^k)) = u^k + \lambda_k(s^k - v^k)$ in the Algorithm 3.3. Hence the motion from the point u^k to u^{k+1} does not hampered by the projection and, therefore, this eliminates the phenomenon of zigzagging of kind II of the pure subgradient procedure that could be occurred due to the projection.

3.4 Bundle Method

In this section we briefly present a class of method which is closely related to subgradient methods- known as *bundle method.* A detail description can be found in Hiriart-Urruty and Lemaréchal [47], [48], Kiwiel [51] or [37]. Bundle method which have stemmed from the work of Lemaréchal [57], [58] is aimed at devising an ascent iterative procedure (in case of problem of maximization) for nonsmooth optimization problems. This requires an ascent direction d^k with the property

$$\phi(u^k + \lambda_k d^k) > \phi(u^k)$$

for all $\lambda_k \in (0, \lambda]$, where λ is some positive number, at any iterate point u^k. In order to obtain a significant improvement of the objective function ϕ at any iterate point u^k, it is required to find a step direction $d^k \in \mathbb{R}^m$ such that

$$\phi(u^k + d^k) \geq \phi(u^k) + \varepsilon, \quad \text{for some} \quad \varepsilon > 0. \tag{3.33}$$

This can be obtained by employing a concept called *ε-subdifferential*, defined as

$$\partial_\varepsilon \phi(u^k) = \{g \in \mathbb{R}^m : \phi(u) \leq \phi(u^k) + g(u - u^k) + \varepsilon, \quad \forall u \in \mathbb{R}^m\}$$

where ϕ is a concave, not necessarily differentiable function on $\mathbb{R}^m$ and $\varepsilon > 0$. Elements of $\partial_\varepsilon \phi(u^k)$ are called *ε-subgradients* of ϕ at u^k. $\partial_\varepsilon \phi(u^k)$ is a nonempty convex closed set.

Correspondingly, the ε-directional derivative of ϕ at u^k along a direction d is defined as

$$\phi'_\varepsilon(u^k; d) = \sup_{t>0} \frac{\phi(u^k + td) - \phi(u^k) - \varepsilon}{t}.$$

It has been shown that (see, for instance, [60] p. 101)

$$\phi'_\varepsilon(u^k; d) = \min\{gd : g \in \partial_\varepsilon \phi(u^k)\}. \tag{3.34}$$

Note that if a direction d can be found such that $\phi'_\varepsilon(u^k; d) > 0$, then the objective function ϕ can be increased by at least ε along the direction d. Therefore, it

is desirable to select a normalized step direction $\bar{d}^k$ such that the ε-directional derivative is maximized; i.e.,

$$
\begin{aligned}
\bar{d}^k &= \arg\{ \max_{\|d\|=1} \phi'_\varepsilon(u^k; d) \} \\
&= \arg\{ \max_{\|d\|=1} \min_{g\in\partial_\varepsilon\phi(u^k)} gd \}
\end{aligned}
$$

Hence, by the well known Min-max Theorem we can exchange the optimization order and get

$$
\bar{d}^k = \arg\{ \min_{g\in\partial_\varepsilon\phi(u^k)} \max_{\|d\|=1} gd \} \tag{3.35}
$$

We now assume that u^k is not ε-optimal (otherwise we are done), i.e., $0 \notin \partial_\varepsilon\phi(u^k)$. Then, the solution of the latter maximization problem in 3.35 is

$$
d = \frac{g}{\|g\|}.
$$

So we obtain,

$$
\bar{d}^k = \arg\{ \min_{g\in\partial_\varepsilon\phi(u^k)} \|g\| \}.
$$

$\bar{d}^k$ is known as the ε-*(steepest) ascent direction.*

Hence, an ε-steepest ascent direction at any iterate point u^k can be obtained by solving the minimum norm problem

$$
\begin{aligned}
\min \quad & \|g\| \\
\text{s.t.} \quad & g \in \partial_\varepsilon\phi(u^k).
\end{aligned} \tag{3.36}
$$

Notice that this problem demands the knowledge of the whole ε-subdifferential $\partial_\varepsilon\phi(u^k)$. Generally, since the set $\partial_\varepsilon\phi(u^k)$ is impossible or too costly to calculate, the idea of the bundle method is to accumulate subgradients s^i of the past iterates in a bundle

$$
B^t = \{s^1, s^2, \ldots, s^t\}
$$

and approximate $\partial_\varepsilon\phi(u^k)$ by the convex hull of the bundle elements given by

$$
[B_\varepsilon^t] := \{g \in \mathbb{R}^m / \ g = \sum_{i=1}^t \alpha_i s^i, \ s^i \in B^t, \ \sum_{i=1}^t \alpha_i = 1, \ \alpha_i \geq 0, \ \sum_{i=1}^t \alpha_i e_i \leq \varepsilon\}
$$

where e_i is a linearization error for element i at the point u^k, i.e.,

$$e_i := e(u^i; u^k) = \phi(u^i) + s^i(u^k - u^i) - \phi(u^k),$$

where $s^i \in \partial\phi(u^i)$. Due to the definition of subgradient, we have $e^i = e(u^i; u^k) \geq 0$ for all $u^i, u^k \in \mathbb{R}^m$. Thus, choosing a tolerance $\varepsilon_k > 0$ at each iterate point u^k and replacing $\partial_{\varepsilon_k}\phi(u^k)$ by $[B^t_{\varepsilon_k}]$, a direction $\bar{d}^k$ in $[B^t_{\varepsilon_k}]$ that has the smallest norm is used as a trial step direction. That is, at each iterate point u^k, the trial step direction is the solution of the problem

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|g\|^2 \\ \text{s.t.} \quad & g \in [B^t_{\varepsilon_k}]. \end{aligned}$$

Due to the definition of $[B^t_{\varepsilon_k}]$ this is equivalent to finding the multipliers α_i for $i = 1, 2, \ldots, t$ that solve the quadratic programming problem

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|\textstyle\sum_{i=1}^t \alpha_i s^i\|^2 \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^t \alpha_i e_i \leq \varepsilon_k \\ & \textstyle\sum_{i=1}^t \alpha_i = 1 \\ & \alpha_i \geq 0, \quad \forall i = 1, 2, \ldots, t. \end{aligned} \tag{3.37}$$

If the multipliers $\alpha_i^k, \quad i = 1, 2, \ldots, t$, solve the problem 3.37, then we obtain a trial step direction

$$\bar{d}^k = \sum_{i=1}^t \alpha_i^k s^i.$$

Line search is then performed along this trial direction. If a point, say y^{t+1}, in the trial direction yields an improvement of the objective value by at least ε_k, then we have a success and u^k is updated to the new point. such a successful step is known as *serious step.* If $[B^t_{\varepsilon_k}]$ is not adequate enough to approximate $\partial_{\varepsilon_k}\phi(u^k)$, the trial direction may not be an ascent direction. In this case u^k is not updated but a subgradient $s^{t+1} \in \partial\phi(y^{t+1})$ is added to the bundle resulting

$$B^{t+1} = \{s^1, s^2, \ldots, s^t, s^{t+1}\},$$

and the step is called a *null step.* Null steps generate more subgradients of ϕ at points near to u^k so that $[B^t_{\varepsilon_k}]$ become closer to $\partial_{\varepsilon_k}\phi(u^k)$. However, several null

steps may be required before a serious step is obtained.

We conclude this section with the following remarks. Bundle method can be used to maximize nonsmooth dual functions so as to provide ascent directions that are lacked in the subgradient methods. Moreover, bundle methods provide a better stopping criteria since a zero-vector in $\partial_{\varepsilon_k}\phi(u^k)$ can be identified while solving for the smallest norm ε-subgradient. However, there are numerous difficulties with this method. The main difficulty in this method is the choice of the tolerance ε_k in 3.37. This tolerance controls the radius of the ball in which the bundle model is thought to be a good approximation of $\partial_{\varepsilon_k}\phi(u^k)$. One can observe a conflicting situation between the sizes of ε_k, since on one hand for large ε_k the bundle model $[B_{\varepsilon_k}^t]$ gives a bad approximation of $\partial_{\varepsilon_k}\phi(u^k)$ (see, [60]) and on the other hand for small ε_k we can only expect a small increase of ϕ due to 3.33. For this reason it is difficult to derive exact rules for updating ε_k. The quadratic programming and line search involving at each iterate point can also lead to too much computations. This is a crucial problem, particularly, in the case of implementation of large scale problems. Moreover, the bundle method subject to zigzags. If we apply this method to solve, for instance, the problem in Example 3.2, then the method is reduced to the pure subgradient procedure since the subgradient of the objective function at any point is constant. Hence, the ε-steepest ascent direction turnout to be the (sub)gradient itself and the method suffers zigzagging of kind II as described in Example 3.2.

4.   A ZIGZAG-FREE SUBGRADIENT PROCEDURE

4.1   Introduction

The zigzagging behavior that arises from the step direction of the subgradient methods makes the computational performance of the procedures poor since it slows the convergence of the procedures. As mentioned earlier, the zigzagging phenomenon can be manifested itself either in the form of zigzagging of kind I or the zigzagging of kind II; and possibly the whole procedure can be exposed to both kind of zigzagging. This obviously can make the procedure quite inefficient, particularly for large sized problems.

The deflected subgradient and the conditional subgradient methods that have been discussed in Section 3.2 and 3.3, respectively, are designed with the intent of alleviating the zigzagging phenomena of the pure subgradient method. However, even though the deflected subgradient procedure can avoid zigzagging of kind I, it can not deal with zigzagging of kind II. On the other hand, the conditional subgradient procedure can eliminate zigzagging of kind II but has no way to control zigzagging of kind I. Therefore, none of these procedures can completely eliminate the zigzagging behavior.

In this chapter we design a new strategy for selection of step direction that can help to completely eliminate both kind of zigzagging. The strategy aims at combining together and extending deflected and conditional subgradient procedures while keeping the simplicity of the procedures. In Chapter 3, we have noticed that the zigzagging of kind II may arise only when the iterate points run across

the relative boundary of a feasible region and such zigzagging can be controlled by the conditional subgradient procedure if a suitable conditional subgradient is chosen for the step direction (see, Section 3.3). Moreover, the deflected subgradient procedures, with appropriate choice of a deflection parameter, can help to eliminate Zigzagging of Kind I when the iterates are moving within the relative interior of the feasible set. However, both methods cannot determine a priori a type of zigzagging that would going to arise. Here we construct the procedure that mimicking the conditional subgradient strategy while the iterates running across the boundary of the feasible set and switches to the deflected subgradient procedure whenever required. That is, we design a procedure that inspects a resulting subgradient direction at each iterate point and informs itself the type of zigzagging intended to arise. Based on the information, an appropriate step direction that can protect the appearance of a zigzag is selected.

4.2 A Generic Hybrid Subgradient Procedure

In this section we present a new algorithm with specific rules for selection of subgradient based step direction which can completely eliminate the zigzagging phenomena. Recall that a step direction d^k is said to be infeasible at an iterate point $u^k \in \mathrm{bd}(\Omega)$ if $d^k v > 0$ for some $v \in \mathrm{N}_\Omega(u^k)$; i.e., d^k forms an an acute angle with a normal vector at u^k. The more acute (i.e., the smaller) such an angle is the stronger that the projection operator would hamper the motion of the iterates and hence the worse the zigzagging phenomena of kind II. To avoid this difficulty, we would like to enable the procedure to step using the direction of an appropriately chosen tangent vector at a boundary point u^k instead of moving using the direction of the inconvenient infeasible direction. This can be accomplished by choosing a step direction Δ^k at $u^k \in \mathrm{bd}(\Omega)$ by the rule

$$\Delta^k = s^k - v^k \tag{4.1}$$

where $s^k \in \partial \phi(u^k)$ and $v^k = \mathrm{P}_{\mathrm{N}_\Omega(u^k)}(s^k)$ whenever the subgradient s^k is infeasible at u^k. Since $\Delta^k = s^k - \mathrm{P}_{\mathrm{N}_\Omega(u^k)}(s^k)$ is orthogonal to the normal cone, $\mathrm{N}_\Omega(u^k)$, Δ^k is a tangent vector of the set Ω at u^k. In particular, if Ω is a polyhedral set then there exists a $\lambda > 0$ such that

$$u^k + \lambda_k \Delta^k \in \Omega \quad \text{for all} \quad \lambda_k \in [0, \lambda],$$

in which case $u^{k+1} = \mathrm{P}_\Omega(u^k + \lambda_k \Delta^k) = u^k + \lambda_k \Delta^k$.

The vector Δ^k obtained by the rule (4.1) is a conditional subgradient of ϕ at u^k (see, Section 3.3). In this procedure, however, we use it as a step direction only when the subgradient at the iterate point is an infeasible direction, in contrast to the conditional subgradient procedure which always use it as a step direction. In other cases, i.e., if s^k is not infeasible direction or if $u^k \in \mathrm{int}(\Omega)$, we make the procedure to switch to the deflected subgradient strategy in order to avoid an occurrence of a zigzagging of kind I. Thus the generic procedure which generates the sequence of iterates $\{u^n\}$ in Ω is as follows.

Algorithm 4.1: **A generic hybrid subgradient algorithm:**

- **Step 0** (Initialization): Set $k = 0$, $\Delta^{-1} = 0$ and choose $u^0 \in \Omega$.

- **Step 1** (Determine the step direction):

$$\Delta^k = \begin{cases} s^k - v^k, & \text{if } u^k \in \mathrm{bd}(\Omega) \text{ and } s^k \text{ is infeasible direction at } u^k \\ s^k + \delta_k \Delta^{k-1}, & \text{otherwise} \end{cases}$$

$$(4.2)$$

  where $s^k \in \partial \phi(u^k)$, $v^k = \mathrm{P}_{\mathrm{N}_\Omega(u^k)}(s^k)$ and δ_k is a deflection parameter to be chosen appropriately.

- **Step 2**:

$$u^{k+1} = \mathrm{P}_\Omega(u^k + \lambda_k \Delta^k) \tag{4.3}$$

- **Step 3**: Let $k := k + 1$; and Repeat Step 1 until some stopping condition.

We call the step direction given by (4.2) the *hybrid subgradient (step) direction* for the reason that it takes either a conditional subgradient or a deflected subgradient direction. We shall determine an appropriate choices of the step length λ_k and the deflection parameter δ_k in order to enable the procedure to eliminate the zigzagging phenomena while preserving the two important properties of the subgradient methods that help the procedure to converge; namely, if $\{u^k\} \subseteq \Omega$ is the sequence of the iterates then (i) the step direction at u^k should make an acute angle with the direction leading to an optimal solution, and (ii) the sequence $\{\|u^k - u^*\|\}$ should be strictly decreasing and eventually converging to 0.

The following Lemmas will be used in the proofs of the theorems that follow.

Lemma 4.1: Suppose Ω is a non-empty closed convex subset of $\mathbb{R}^m$. If $v \in \mathbb{R}^m$ and $p_v = P_\Omega(v)$ then

(i) $\quad \|p_v - u\| \le \|v - u\| \quad$ for any $u \in \Omega$.

(ii) $\quad \alpha(v - u)(p_v - v) \le 0$, for any scalar $\alpha > 0$ and any $u \in \Omega$.

Proof: Note that $v - p_v \in N_\Omega(p_v)$. Therefore, for any $u \in \Omega$, it holds that

$$(u - p_v)(v - p_v) \le 0 \tag{4.4}$$

(i) The inequality in (4.4) directly provide $uv - up_v \le vp_v - \|p_v\|^2$. Now we use this last inequality in the following relation.

$$
\begin{aligned}
\|p_v - u\|^2 - \|v - u\|^2 &= \|p_v\|^2 - \|v\|^2 + 2(uv - up_v) \\
&\le \|p_v\|^2 - \|v\|^2 + 2vp_v - 2\|p_v\|^2 \\
&= -\|p_v - v\|^2 \\
&\le 0.
\end{aligned}
$$

Therefore the inequality in (i) holds.

(ii) Again using (4.4),

$$(u - v + v - p_v)(v - p_v) \leq 0.$$
$$\Rightarrow (u - v)(v - p_v) \leq -\|v - p_v\|^2 \leq 0.$$
$$\Rightarrow (v - u)(p_v - v) \leq 0.$$

$\square$

As consequences of Lemma 4.1 we have the followings: If $\{u^k\}$ is a sequence of iterate points generated by the rule (4.3), then

- setting $u = u^*$ and $v = u^k + \lambda_k \Delta^k$, (i) of Lemma 4.1 gives us

$$\|u^{k+1} - u^*\| \leq \|u^k + \lambda_k \Delta^k - u^*\|, \tag{4.5}$$

- setting $u = u^k$, $v = u^k + \lambda_k \Delta^k$ and $\alpha = 1/\lambda_k$, (ii) of the Lemma yields

$$\Delta^k(u^{k+1} - u^k - \lambda_k \Delta^k) \leq 0$$
$$\Rightarrow \Delta^k(u^k - u^{k+1}) \geq -\lambda_k \|\Delta^k\|^2. \tag{4.6}$$

These relations are used in the proofs of the followings.

Lemma 4.2: Suppose $\{u^k\} \subseteq \Omega$ is a sequence of iterates generated by (4.3) and the step direction Δ^k is the hybrid subgradient direction given by (4.2). If

$$0 < \lambda_k \leq \frac{\phi^* - \phi(u^k)}{\|\Delta^k\|^2} \tag{4.7}$$

Then

$$\Delta^k(u^* - u^k) \geq \phi(u^*) - \phi(u^k)$$

for any iterate point u^k.

Proof: (By induction on k)

Let $k = 0$. Then, either $\Delta^0 = s^0 - \mathrm{P}_{\mathrm{N}_\Omega(u^0)}(s^0) \in \partial^\Omega \phi(u^0)$ or $\Delta^0 = s^0 \in \partial \phi(u^0)$.

For both cases, the assertion is true.

Suppose the claim holds for $k \in \mathbb{N}$. Then, in case $u^{k+1} \in \mathrm{bd}(\Omega)$ and s^{k+1} is infeasible at u^{k+1}, we obtain

$$\Delta^{k+1} = s^{k+1} - \mathrm{P}_{\mathrm{N}_\Omega(u^{k+1})}(s^{k+1}) \in \partial^\Omega \phi(u^{k+1})$$

From this it follows,

$$\Delta^{k+1}(u^* - u^{k+1}) \geq \phi(u^*) - \phi(u^{k+1}).$$

In the remaining cases, $\Delta^{k+1} = s^{k+1} + \delta_{k+1}\Delta^k$. Hence,

$$\Delta^{k+1}(u^* - u^{k+1}) = s^{k+1}(u^* - u^{k+1}) + \delta_{k+1}\Delta^k(u^* - u^{k+1}) \tag{4.8}$$

Next to show $\Delta^k(u^* - u^{k+1}) \geq 0$:

$$
\begin{aligned}
\Delta^k(u^* - u^{k+1}) &= \Delta^k(u^* - u^k + u^k - u^{k+1}) \\
&= \Delta^k(u^* - u^k) + \Delta^k(u^k - u^{k+1}) \\
&\geq \phi(u^*) - \phi(u^k) + \Delta^k(u^k - u^{k+1}) \quad [\text{ induction hypothesis}] \\
&\geq \phi(u^*) - \phi(u^k) - \lambda_k\|\Delta^k\|^2 \quad\quad [\text{ followed from (4.6)}] \\
&\geq 0 \quad\quad\quad\quad\quad\quad\quad\quad [\text{ from the given condition on } \lambda_k.]
\end{aligned}
$$

Using this in (4.8), we obtain

$$\Delta^{k+1}(u^* - u^{k+1}) \geq s^{k+1}(u^* - u^{k+1}) \geq \phi(u^*) - \phi(u^{k+1}.$$

$\square$

The following theorem extends important properties of subgradient methods.

Theorem 4.3: Suppose $\{u^k\}$ is a sequence of iterates generated by (4.3) and the step direction Δ^k is the hybrid subgradient direction given by (4.2). If

$$0 < \lambda_k \leq \frac{\phi^* - \phi(u^k)}{\|\Delta^k\|^2}, \tag{4.9}$$

then,

(i) $\Delta^k(u^* - u^k) > 0$ for all $u^k \notin \Omega^*$.

(ii) $\|u^{k+1} - u^*\| < \|u^k - u^*\|$ for all $u^k \notin \Omega^*$.

(iii) If $\Delta^n = 0$ at some iterate point u^n, then u^n is an optimal solution.

Proof: (i) Follows obviously from Lemma 4.2 since for all $u^k \notin \Omega^*$ it holds that

$$\Delta^k(u^* - u^k) \geq \phi(u^*) - \phi(u^k) > 0.$$

(ii) Using Lemma 4.1 we obtain

$$\begin{aligned}
\|u^{k+1} - u^*\|^2 &\leq \|u^k + \lambda_k \Delta^k - u^*\|^2 \\
&= \|u^k - u^*\|^2 + \lambda_k[\lambda_k \|\Delta^k\|^2 - 2\Delta^k(u^* - u^k)].
\end{aligned}$$

Therefore, we need only to show $\lambda_k\|\Delta^k\|^2 - 2\Delta^k(u^* - u^k) < 0$. Using the condition on λ_k and Lemma 4.2 we have

$$\lambda_k\|\Delta^k\|^2 \leq \phi^* - \phi(u^k) \leq \Delta^k(u^* - u^k) < 2\Delta^k(u^* - u^k).$$

Hence, $\lambda_k\|\Delta^k\|^2 - 2\Delta^k(u^* - u^k) < 0$. This completes the prove of (ii).

(iii) $\phi^* = \max\{\phi(u) : u \in \Omega\} \geq \phi(u^n)$. On the other hand, using Lemma 4.2 and the given condition on Δ^n, it holds that $\phi^* - \phi(u^n) \leq \Delta^n(u^* - u^n) = 0$. Hence, $\phi^* \leq \phi(u^n)$. Putting these together, we have $\phi(u^n) = \phi^*$. $\square$

Theorem 4.3 establishes important properties that (i) at each iterate point the hybrid subgradient direction Δ^k forms an acute angle with the direction leading from u^k to an optimal solution u^*, and (ii) the sequence $\{\|u^* - u^k\|\}$ is monotonic decreasing. Moreover, Theorem 4.3(iii) provides a sufficient condition for optimality.

Theorem 4.4: Suppose a sequence $\{u^k\} \subseteq \Omega$ is constructed by the hybrid subgradient procedure using a step length

$$\lambda_k = \mu_k \frac{\phi^* - \phi(u^k)}{\|\Delta^k\|^2}, \quad 0 < \varepsilon_0 \leq \mu_k \leq 1. \tag{4.10}$$

If $\sup\{\|\Delta^k\| : k = 0, 1, 2, \ldots\} < \infty$ and Ω^* is nonempty, then $\phi(u^k) \to \phi^*$ and the sequence $\{u^k\} \to u^*$ for some $u^* \in \Omega^*$.

Proof: Using Lemma 4.1 one obtains

$$
\begin{aligned}
\|u^{k+1} - u^*\|^2 &\leq \|u^k + \lambda_k \Delta^k - u^*\|^2 \\
&= \|u^k - u^*\|^2 + \lambda_k^2 \|\Delta^k\|^2 - 2\lambda_k \Delta^k(u^* - u^k) \\
&= \|u^k - u^*\|^2 + \mu_k^2 \frac{(\phi^* - \phi(u^k))^2}{\|\Delta^k\|^2} - 2\mu_k \frac{\phi^* - \phi(u^k)}{\|\Delta^k\|^2} \Delta^k(u^* - u^k) \\
&\leq \|u^k - u^*\|^2 - \mu_k(2 - \mu_k)\frac{(\phi^* - \phi(u^k))^2}{\|\Delta^k\|^2} \quad\quad (4.11) \\
&\leq \|u^k - u^*\|^2 - \frac{\varepsilon_0}{c^2}(\phi^* - \phi(u^k))^2 \quad\quad (4.12)
\end{aligned}
$$

where $c = \sup\{\|\Delta^k\| : k = 0, 1, 2, \ldots\}$. Note that the relation (4.11) follows by using Lemma 4.2 in the previous relation and the relation (4.12) holds since $2 - \mu_k \geq 1$ and $\mu_k \geq \varepsilon_0$. From (4.12) we conclude that $(\phi(u^k) - \phi^*)^2 \to 0$, or equivalently, $\phi(u^k) \to \phi^*$ (otherwise, $\exists \delta > 0$ and $K \geq 0$ such that $(\phi(u^k) - \phi^*)^2 > \delta \ \forall k \geq K$. Let, with out loss of generality, $K = 0$. Using this inequality in (4.12), we obtain $\|u^{k+1} - u^*\|^2 \leq \|u^k - u^*\|^2 - \alpha$ where $\alpha = \frac{\varepsilon_0 \delta}{c^2}$ is positive. The last inequality written recursively yields $\|u^{k+1} - u^*\|^2 \leq \|u^0 - u^*\|^2 - \alpha k \ \to \ -\infty$ as $k \to \infty$, which is a contradiction).

Now it remains only to show that $u^k \to u^* \in \Omega^*$. Observe that $\{\|u^k - u^*\|\}$ is monotonic decreasing implies the boundedness of $\{u^k - u^*\}$ and this again implies that $\{u^k\}$ is bounded. Hence an accumulation point $\bar{u} \in \Omega$ of $\{u^k\}$ exists; i.e., there is a subsequence $\{u^{k_n}\}$ which converges to $\bar{u}$. It follows that $\phi(u^{k_n}) \to \phi(\bar{u})$ since ϕ is continuous. Hence we have $\phi(\bar{u}) = \phi^*$; i.e., $\bar{u} \in \Omega^*$. We now show that the entire sequence $\{u^k\}$ converges to $\bar{u}$. Since $\bar{u} \in \Omega^*$, the sequence $\|u^k - \bar{u}\|$ is bounded and monotonic decreasing with a subsequence converging to 0. Hence the sequence $\|u^k - \bar{u}\| \to 0$. Thus, $\{u^k\} \to \bar{u}$. $\quad\square$

The next theorem describes the most important property of the hybrid subgradient procedure.

Theorem 4.5: Suppose the hybrid step direction Δ^k is given by (4.2), the step length $\lambda_k > 0$, and the deflection parameter δ_k is given by either

(i)

$$\delta_k = \begin{cases} -\tau_k \frac{s^k \Delta^{k-1}}{\|\Delta^{k-1}\|^2}, & \text{if} \quad s^k \Delta^{k-1} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.13}$$

where $1 < \tau_k < 2$ and $s^k \in \partial\phi(u^k)$,

or

(ii)

$$\delta_k = \frac{\|s^k\|}{\|\Delta^{k-1}\|}, \tag{4.14}$$

then the hybrid subgradient procedure, Algorithm 4.1, is free of zigzagging of kind I and kind II.

Proof: Suppose s^k is infeasible; i.e., $u^k \in \text{bd}(\Omega)$ and s^k forms an acute angle with a normal vector of Ω at u^k. Then $\Delta^k = s^k - \text{P}_{\text{N}_\Omega(u^k)}(s^k)$ is orthogonal to a normal vector and hence $\Delta^k \in \text{T}_\Omega(u^k)$. Thus, zigzagging of kind II can not arise. Moreover, if s^k is not infeasible then the problem of concern is only a zigzagging of kind I. But in this case $\Delta^k = s^k + \delta_k \Delta^{k-1}$.

(i) Now if δ_k is chosen according to (4.13), then δ_k either 0 or a positive scalar. If $s^k \Delta^{k-1} \geq 0$, then the claim of the theorem follows since in this case $\Delta^k = s^k$ as $\delta_k = 0$. Thus consider, the case $s^k \Delta^{k-1} < 0$. Then,

$$\begin{aligned} \Delta^k \Delta^{k-1} &= s^k \Delta^{k-1} + \delta_k \|\Delta^{k-1}\|^2 \\ &= s^k \Delta^{k-1} - \tau_k s^k \Delta^{k-1} \\ &= (1 - \tau_k) s^k \Delta^{k-1} \\ &> 0, \end{aligned}$$

since $1 - \tau_k < 0$ and also $s^k \Delta^{k-1} < 0$. Thus, also zigzagging of kind I does not arise.

(ii) If δ_k is given by (4.14), then $\Delta^k = s^k + \frac{\|s^k\|}{\|\Delta^{k-1}\|}\Delta^{k-1}$

$$
\begin{aligned}
\Rightarrow \quad \Delta^k\Delta^{k-1} = \quad & s^k\Delta^{k-1} + \|s^k\|\|\Delta^{k-1}\| \\
\geq \quad & \|s^k\|\|\Delta^{k-1}\| - |s^k\Delta^{k-1}| \qquad [\text{ since } s^k\Delta^{k-1} \geq -|s^k\Delta^{k-1}| \,] \\
\geq \quad & 0 \qquad\qquad\qquad\qquad\qquad [\text{ Cauchy-Schwarz inequality }]
\end{aligned}
$$

$\square$

4.3 Hybrid Subgradient Procedure for the Lagrangian Dual

In this section we apply the hybrid subgradient procedure to solve the Lagrangian dual problem 2.3. Following our previous notations, let

$$
\begin{aligned}
\mathbb{X}^*(\bar{u}) \quad &:= \quad \{\bar{x} \in \mathbb{X} : \ \bar{x} \text{ is an optimal solution of the subproblem SP}(\bar{u}) \,\} \\
\mathrm{N}_+(.) \quad &:= \quad \mathrm{N}_{\mathbb{R}^m_+}(.), \ \text{ normal cone of the positive octant at a point} \\
\Omega \quad &:= \quad \mathbb{R}^m_+ = \{u \in \mathbb{R}^m : \ u \geq 0 \,\}
\end{aligned}
$$

and $s^k = b - Ax^k \in \partial\phi(u^k)$, for $x^k \in \mathbb{X}^*(u^k)$.

Note that for all such s^k, $\{\|s^k\|\}$ is bounded and hence $\sup\{\|s^k\| : k \in \mathbb{N}\} < \infty$. Therefore, a hybrid subgradient vector Δ^k obtained from s^k by (4.2) is bounded, too. As a result, Theorem 4.4 is valid for this problem. Furthermore, the hybrid step direction is given by $\Delta^k = s^k + \delta_k\Delta^{k-1}$ wherever the subgradient s^k is not infeasible at the iterate point u^k. Now we describe how one can identify an infeasibility of s^k and find a rule that can be used to determine an appropriate step direction when s^k is infeasible.

Obviously, $u^k \in \mathrm{bd}(\mathbb{R}^m_+)$ only if $u^k_j = 0$ for some of its *j-th* component. Furthermore, a subgradient s^k at boundary point is infeasible if and only if $s^k_i < 0$ for some of its *i-th* component for which $u^k_i = 0$; i.e, if $\mathcal{I}^k_0 = \{i : u^k_i = 0\}$, then $s^k \in \partial(u^k)$ is infeasible at u^k if and only if $s^k_i < 0$ for some of its components $i \in \mathcal{I}^k_0$. In such case the hybrid step direction Δ^k is determined by $\Delta^k = s^k - v^k$

where $s^k = b - Ax^k$ for $x^k \in \mathbb{X}^*(u^k)$ and $v^k = \mathrm{P}_{\mathrm{N}_+}(s^k)$ so that each of the i-th component v^k is given by

$$v_i^k = \begin{cases} s_i^k, & \text{if } u_i^k = 0 \text{ and } s_i^k < 0 \\ 0, & \text{otherwise.} \end{cases}$$

Thus, in case s^k is infeasible at $u^k \in \mathrm{bd}(\mathbb{R}_+^m)$ the hybrid step direction can be expressed as

$$\bar{\Delta}^k = (\bar{\Delta}_1^k, \bar{\Delta}_2^k, \ldots, \bar{\Delta}_m^k) \tag{4.15}$$

where

$$\bar{\Delta}_i^k = \begin{cases} 0, & \text{if } u_i^k = 0 \text{ and } b_i - A^i x^k < 0 \\ b_i - A^i x^k, & \text{otherwise.} \end{cases}$$

for $x^k \in \mathbb{X}^*(u^k)$ and $b_i - A^i x^k$ is the i-th component of $s^k = b - Ax^k$.

Therefore, using the above results, the corresponding hybrid subgradient for the Lagrangian dual problem can be constructed as follows.

Algorithm 4.2: **Hybrid subgradient algorithm for the Lagrangian dual:**

- **Step 0:** (Initialization) Set $k := 0$, and choose $u^0 \in \mathbb{R}_+^m$.
    Let $\Delta^{-1} = 0$, $\bar{u}^{-1} = u^0$, and $\phi(\bar{u}^{-1}) = -\infty$.

- **Step 1:** Solve the subproblem $\mathrm{SP}(u^k)$ to get $x^k \in \mathbb{X}^*(u^k)$ and $\phi(u^k)$.
    $s^k = b - Ax^k$, a subgradient of ϕ at u^k.
    If $\phi(u^k) \geq \phi(\bar{u}^{k-1})$, then
        $\bar{u}^k = u^k$ and $\phi(\bar{u}^k) = \phi(u^k)$.
      else
        $\bar{u}^k = \bar{u}^{k-1}$ and $\phi(\bar{u}^k) = \phi(\bar{u}^{k-1})$.

- **Step 2:** Determine the hybrid subgradient direction:
    Let $\mathcal{I}_0^k = \{ i : u_i^k = 0 \}$

$$\Delta^k = \begin{cases} \bar{\Delta}^k, & \text{if } s_i^k < 0 \text{ for some } i \in \mathcal{I}_0^k. \\ s^k + \delta_k \Delta^{k-1}, & \text{otherwise} \end{cases}$$

where $\bar{\Delta}^k$ and δ_k are given by 4.15 and 4.13, respectively.

- **Step 3:** $u^{k+1} = (u_1^{k+1}, u_2^{k+1}, \ldots, u_m^{k+1})$,

  where, $u_i^{k+1} = \max\{\, 0, \ u_i^k + \lambda_k \Delta_i^k \}$ for each $i = 1, 2, \ldots, m$.

  Let $k := k + 1$, repeat Step 1, (until a stopping condition).

Note that in Step 1 $\bar{u}^k$ is updated only if ϕ has a greater value (up to the current iteration), so that ϕ is monotonic increasing on the subsequence $\{\bar{u}^k\}$. At termination, the last iterate $\bar{u}^n$ would be an output as a solution of the Lagrangian dual (LD) with the objective value $\phi(\bar{u}^n)$.

Finally, the comment on a termination criterion of the hybrid procedures is similar to that of any other subgradient methods. Indeed, the procedure can be stopped when $\Delta^k = 0$ at some iterate point; but there is no guarantee for occurrence of this. Hence a possible stopping rule is either to stop after a sufficiently large number of iterations or if the value of ϕ has not increased for a number of consecutive iterations. The procedure may also made to terminate if Δ^k or λ_k is below certain threshold. A combination of these may also be used to terminate the procedure.

5. PRIMAL SOLUTIONS WITHIN THE SUBGRADIENT PROCEDURES

In this chapter, we analyze two important extensions of the subgradient methods which have been developed recently in order to construct primal solution within the subgradient schemes without a significant additional computational effort. Originally, the procedures were developed either for problems arising from linear programming or convex programming problems. We adopt the procedures to find a near-optimal solution for integer programming problems.

5.1   Introduction

This chapter deals with a solution method of the original integer programming problem, i.e, the primal problem IP. While the subgradient optimization procedures that have been described in the previous sections can be quite powerful in providing a good lower bound (for the minimization problem) to IP via the solution of its Lagrangian dual LD, the disadvantage is that a primal optimal solution (or even a feasible solution) to IP is not usually available using these schemes. Such a primal solution is important not only as the final solution of IP but also in the branch and bound procedure which is usually applied to search for an integral solution since a primal solution provides an upper bound on the IP. Moreover, the availability of a primal solution can provide a natural stopping criterion of a subgradient procedure.

One may attempt to obtain a primal solution after enforcing complementary slackness on the derived optimal solution of LD. Note that if $\hat{x} \in \mathbb{X}$ and $\hat{u} \in \mathbb{R}^m_+$

satisfy the optimality condition (Theorem 2.9), then $\hat{x}$ is an optimal solution in the primal problem and $\hat{u}$ is an optimal solution in its Lagrangian dual problem. Thus, the possible strategy for the application of Lagrangian relaxation techniques could be to find first an optimum $\hat{u}$ in LD and once this has been done, then try to find a complementary $\hat{x} \in \mathbb{X}$ for which the optimality conditions hold by calculating one or more solutions $\hat{x}$ satisfying

$$
\begin{cases}
c\hat{x} + \hat{u}(b - A\hat{x}) = \phi(\hat{u}) \\
b - A\hat{x} \leq 0 \\
\hat{u}(b - A\hat{x}) = 0.
\end{cases}
$$

But this might involve a significant additional computational burden, particularly in the case of a large scale problem. Beside the computational burden, there is no guarantee that this strategy will succeed because (a) there may be no $\hat{u}$ feasible in LD for which the optimality conditions can be made to hold for some $\hat{x} \in \mathbb{X}$ because of the possible duality gap; or (b) the specific optimal $\hat{u}$ we obtained may not admit the optimality conditions for any $\hat{x} \in \mathbb{X}$, although some other $\bar{u} \in \mathbb{R}_+^m$ which maximize the Lagrangian dual may satisfy the optimality condition.

On the other hand, branch and bound is a method guaranteed to find an optimal solution to an integer programming problem. Usually an optimal solution of a linear programming, LP, relaxation is used to provide a lower bound (for a minimization problem) in a branch and bound algorithm and a feasible (integral) solution has to be searched by repeated re-optimizations using a simplex-based algorithms and additional extra constraint in order to get a solution or an upper bound ([64],[72], [84]). However, the search for an integral solution has been a challenging task in such a procedure, particularly, in the case of a large-scale integer programming problem since such a simplex-based algorithms are slow and requires large storage area.

Alternatively, the Lagrangian relaxation and the Lagrangian dual procedures can thus be used to provide bounds in a branch and bound algorithm with relative

easiness and quickness. The issues involved in designing a branch and bound algorithm that uses the Lagrangian relaxation are essentially the same as those that arise when a LP relaxation is used. The difference is, however, the Lagrangian relaxation is used and the subgradient method is applied to solve the resulting Lagrangian dual instead of a simplex-based algorithm of LP problems. This technique provides a lower bound on the IP problem relatively easily and quickly and it has lead to dramatically improved algorithms for a number of important particular problems in the area of such as Travelling Salesman Problem([44], [45]), Scheduling ([33], [34]), Set Covering([32], [13]), Location ([25], [30], [23], [14]) and general integer programming ([39],[74], [35], [36]). Lagrangian Relaxation was successful for these applications because the subgradient optimization method which was used to solve its Lagrangian dual is more powerful than the simplex-based methods available for solving the (generally large) LP relaxation of IP. The important message of these applications is that combinatorial optimization problems frequently can be formulated as large integer programming whose Lagrangian dual can be solved relatively easily and quickly by the subgradient methods.

The drawbacks of a subgradient algorithm is, however, that it does not produce values for the primal variables. Moreover, it has no well defined stopping criteria. Hence, at termination, one does not know how far the value obtained is from the optimal value. In order to deal with these drawbacks, Shor ([80], pp.116-118) proposed to use a sequence of points generated by convex combination of solutions of the subproblems of the Lagrangian dual problem within the subgradient scheme in such a way that it eventually converges to a primal solution by adopting certain restricted step size strategies. He applied this idea within a pure subgradient scheme of a Lagrangian dual problem that came from a linear programming problem to construct a sequence of primal iterates by convex combination of solution of subproblems. Sherali and Choi [78] have used this idea within the deflected subgradient methods that solve Lagrangian duals of

linear programming problem. Larsson, Patriksson and Strömberg [54] also used
the idea to the case of general convex programs within the conditional subgra-
dient scheme. More recently, Barahona and Anbil [6] developed an extension of
a subgradient method which produce an approximate primal solution with in a
deflected subgradient method and presented a successful numerical experiments
with large scale linear programming relaxations arising from set partitioning, set
covering and plant location problems.

In this chapter, we present extensions of the subgradient algorithms to a class
of algorithms that can produce an approximate to a primal solution of integer
programming problem directly from the information generated in the process of
the deflected subgradient or the conditional subgradient optimization methods.
Insights and solution methods for the primal problem are derived from the La-
grangian dualization and convexfication of the IP problem. Recall that (Theorem
2.4), given an integer programming problem

$$
\text{(IP)} \qquad\qquad \min \quad cx \qquad\qquad\qquad (5.1)
$$
$$
\text{s.t.} \quad Ax \geq b
$$
$$
x \in \mathbb{X} \;=\; \{\, x \in \mathbb{Z}_+^n : \; Dx \geq d \,\},
$$

then it holds that

$$
\phi^* \;=\; \min \quad cx \qquad\qquad\qquad (5.2)
$$
$$
\text{s.t.} \quad Ax \geq b
$$
$$
x \in \text{conv}(\mathbb{X})
$$

where ϕ^* is the maximum value of the Lagrangian dual LD of the IP. Note that
(5.2) is a convex programming problem. From these facts, we may say that the
Lagrangian dual of IP is equivalent to convexfication of it. Moreover, there is
no duality gap for problem (5.2). Hence, we aimed at calculating $\hat{x} \in \text{conv}(\mathbb{X})$
which satisfies the optimality condition (Theorem 2.9) together with some $\hat{u} \in \mathbb{R}_+^m$
which maximizes the LD of the convex programming relaxation (5.2) of IP. We call

such a solution $\hat{x}$ a *near-optimal* solution of the IP. Thus, we have the following definition:

Definition 5.1: (**Near-optimal solution**)

Suppose $\hat{u} \in \Omega^* = \{\hat{u} \in \mathbb{R}_+^m : \ \phi(\hat{u}) \geq \phi(u), \ \ \forall u \in \mathbb{R}_+^m\}$

and let

$$\mathbb{X}(\hat{u}) = \{\bar{x} \in \text{conv}(\mathbb{X}) : c\bar{x} + \hat{u}(b - A\bar{x}) \leq cx + \hat{u}(b - Ax), \ \ \ \forall x \in \text{conv}(\mathbb{X})\}.$$

An $\hat{x} \in \mathbb{X}(\hat{u})$ is said to be a *near-optimal* solution of the IP, (5.1), if

$A\hat{x} \geq b \ \ \text{and} \ \ \hat{u}(A\hat{x} - b) = 0.$

$\square$

In Section 5.2, we extend the deflected subgradient procedure so as both a dual and a near-optimal primal solution are generated within the scheme. Section 5.3 deals with such extension of the conditional subgradient procedure. Additionally, while these extended procedures are maintaining the same low computational cost per iteration, they provide a much better stopping criterion. In general, the extensions produce a near-optimal primal vector as well as a dual vector that can be used by themselves to devise a simple heuristic that can convert the near-optimal primal solution into an integral solution, which is usually the case ([8], [7], [6], [50]), or as the starting points of a more exact method such as, for instance, branch-and-bound method.

5.2   The Volume Algorithm

We describe below an extension of the subgradient algorithm which produces a near-optimal primal solution of IP directly from information generated within the process of the deflected subgradient algorithm. Given an IP problem (5.1), we have seen that the optimal objective value of its Lagrangian dual can be obtained from (5.2) which is itself a convex programming (or LP) relaxation problem of

the IP. That is,

$$\phi^* = \min \quad cx$$
$$\text{s.t.} \quad Ax \geq b$$
$$x \in \text{conv}(\mathbb{X}) = \{ \ x : \ x = \sum_i \alpha_i x^i, \sum_i \alpha_i = 1, \alpha_i \geq 0, \ x^i \in \mathbb{X}\}$$

which can be written as

$$\phi^* = \min \quad \sum_i (cx^i)\alpha_i \qquad (5.3)$$
$$\text{s.t.} \quad \sum_i (Ax^i - b)\alpha_i \geq 0$$
$$\sum_i \alpha_i = 1$$
$$\alpha_i \geq 0 \quad \forall i.$$

This is a linear programming problem (generally large-scale) whose dual is

$$\phi^* = \max \quad z$$
$$\text{s.t.} \quad z + u(Ax^i - b) \leq cx^i, \quad \forall i$$
$$u \in \mathbb{R}_+^m, \quad z \in \mathbb{R}$$

or,

$$\phi^* = \max \quad z \qquad (5.4)$$
$$\text{s.t.} \quad z \leq ub + (c - uA)x^i \quad \forall i$$
$$u \in \mathbb{R}_+^m, \quad z \in \mathbb{R}.$$

Setting $z = \phi(u)$, then (5.4) is equivalent to the Lagrangian dual

$$(\text{LD}) \qquad \phi^* = \max\{\phi(u) : \ u \in \mathbb{R}_+^m\}$$

where $\phi(u)$ is the dual function determined pointwise by the subproblem

$$\phi(u) = ub + \min\{(c - uA)x^i : \ x^i \in \mathbb{X}, \ \forall i\}. \qquad (5.5)$$

We have seen already that the Lagrangian dual (LD) can be solved by a subgradient optimization procedure. Beside solving (LD), we want to determine a primal

solution of (5.2) or, equivalently, (5.3) within the subgradient procedures. Such solution yields a convex combination of points in $\mathbb{X}$, and hence is a near-optimal solution of the IP. To generate such solution via a convex combination of points in $\mathbb{X}$, the set of points $\{x^1, x^2, \ldots, x^i, \ldots\} \subseteq \mathbb{X}$ are generated from the solutions of the subproblem (5.5). The basic idea to be used to determine the corresponding weights α_i in the convex combination comes from applying the theorem on volume and duality due to Barahona and Anbil [6], where the α_i are estimating certain volumes associated to active faces at an optimal dual solution as shown in the following theorem.

Theorem 5.1: **(Volume and Duality)**
Consider the problem

$$\max \quad z \tag{5.6}$$
$$\text{s.t} \quad z + a^i u \le b_i, \quad \text{for} \ \ i = 1, 2, \ldots, m$$

where $z \in \mathbb{R}$, and $a^i, u \in \mathbb{R}^n$. Let $(\hat{z}, \hat{u}) \in \mathbb{R} \times \mathbb{R}^n$ be an optimal solution, and suppose that constraints $1, 2, \ldots, m', \ m' \le m$ are active at this point.
Let $\bar{z} < \hat{z}$ and assume that

$$\left. \begin{array}{l} z + a^i u \le b_i, \quad \text{for} \ \ i = 1, 2, \ldots, m' \\ z \ge \bar{z}, \end{array} \right\} \tag{5.7}$$

define a bounded polyhedron. For $1 \le i \le m'$, let γ_i be the volume between the face defined by $z + a^i u \le b_i$ and the hyperplane defined by $z \ge \bar{z}$. Then an optimal dual solution is given by

$$\alpha_i = \frac{\gamma_i}{\sum_{j=1}^{m'} \gamma_j}.$$

Proof: The shaded region, P, in the Figure 5.1 illustrates such a volume in a plane. Consider the polyhedral P defined by (5.7). Since it is a bounded polyhedral it is a polytope and full dimensional. Let F_0 be the face defined by $z \ge \bar{z}$, and F_i the face defined by $z + a^i u \le b_i, \ i = 1, 2, \ldots m'$. Gauss' divergence
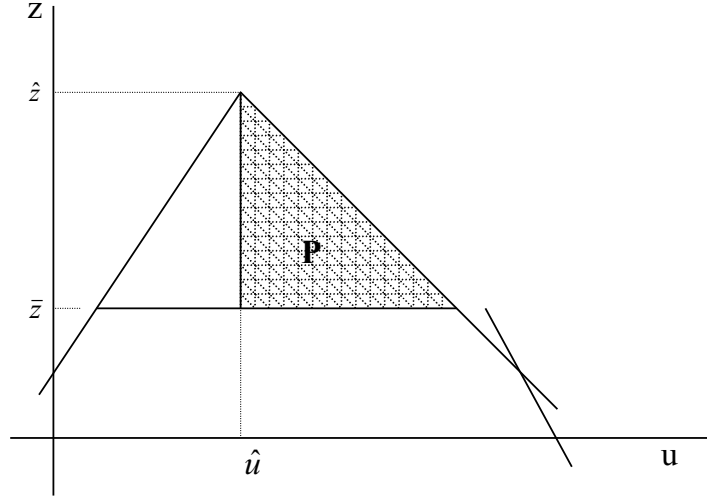
Fig. 5.1: The shaded region P is an example of the required volume.

theorem says that for a closed bounded region P whose boundary is a piecewise smooth orientable surface S, and vector field $\vec{F}$ defined on P,

$$\int_S \vec{F}.\vec{n}dS = \int_P div(\vec{F})dv,$$

where $\vec{n}$ is the outer unit normal vector of the surface S. Taking $\vec{F} = v$, a constant vector, we get

$$\int_S v.\vec{n}dS = 0, \tag{5.8}$$

since $div(\vec{F}) = 0$ for the constant $\vec{F} = v$. By taking $v = e_j$ (the *j-th* unit vector) for $1 \leq j \leq n$, we obtain $v.\vec{n} = n_j$ and hence (5.8) yields $\int_S n_j dS = 0$ for each $j = 1, 2, \ldots, n$. From this, it follows

$$\int_S \vec{n}dS = \vec{0}, \tag{5.9}$$

where $\vec{0}$ denotes a zero vector. Since $S = F_1 \dot\cup F_2 \dot\cup \ldots \dot\cup F_{m'} \dot\cup F_0$, disjoint unions, (5.9) gives us

$$\sum_{i=1}^{m'} \int_{F_i} \vec{n}(F_i)dS + \int_{F_0} \vec{n}(F_0)dS = 0$$

where $\vec{n}(F_i)$ is an outer unit normal to the face F_i. Notice that

$$\vec{n}(F_0) = -(1, 0, \ldots, 0) \quad \text{and} \quad \vec{n}(F_i) = \frac{(1, a^i)}{\|(1, a^i)\|}, \quad \text{for each} \quad i = 1, 2, \ldots m'.$$

Thus, using this in the last integral equation we obtain,

$$\sum_{i=1}^{m'} \left[\frac{(1, \ a^i)}{\|(1, \ a^i)\|} \int_{F_i} dS \right] - (1, 0, \dots, 0) \int_{F_0} dS \ = \ 0$$

This implies,

$$\sum_{i=1}^{m'} \frac{\delta_i}{\|(1, \ a^i)\|} (1, \ a^i) - \delta_0 (1, 0, \dots, 0) \ = \ 0$$

where $\delta_i = \int_{F_i} dS$, the surface area of F_i, for each $i = 0, 1, 2, \dots, m'$. Thus,

$$(1, 0, \dots, 0) = \sum_{i=1}^{m'} \frac{\delta_i}{\delta_0 \|(1, \ a^i)\|} (1, \ a^i),$$

so at this point we have the gradient of the objective function written as a non-negative linear combination of the gradients of the constraints that are active at the optimum. This provides, therefore, an optimal dual solution

$$\alpha_i = \frac{\delta_i}{\delta_0 \|(1, \ a^i)\|} \tag{5.10}$$

for $i = 1, 2, \dots m'$ (where $\alpha_i = 0$, for each $i = m' + 1, m' + 2, \dots, m$). Now we shall see that

$$\gamma_i = c \frac{\delta_i}{\|(1, \ a^i)\|}$$

where c is a constant, and γ_i, as defined earlier, is the volume between F_i and the hyperplane F_0.

If $\delta_i = 0$, then $\gamma_i = 0$. So we have to consider the case when $\delta_i > 0$. For that we apply Gausses theorem again as follows.

Let Q_i be the convex hull of F_i and $(\bar{z}, \hat{u})$ (see Figure 5.2).

Let $\bar{F}_0 = F_0 \cap Q_i$. Notice that the faces of Q_i other than F_0 and F_i are defined by inequalities like $cu \leq d$ where the variable z is absent. This means that the normal to this faces are orthogonal to the z-axis. Therefore, if we use formula (5.8) with $v = e_1 = (1, 0, 0, \dots, 0)$ on Q_i, the only faces with $v.\vec{n} \neq 0$ are F_i and $\bar{F}_0$. Thus applying (5.8) on Q_i with $v = e_1$ we get

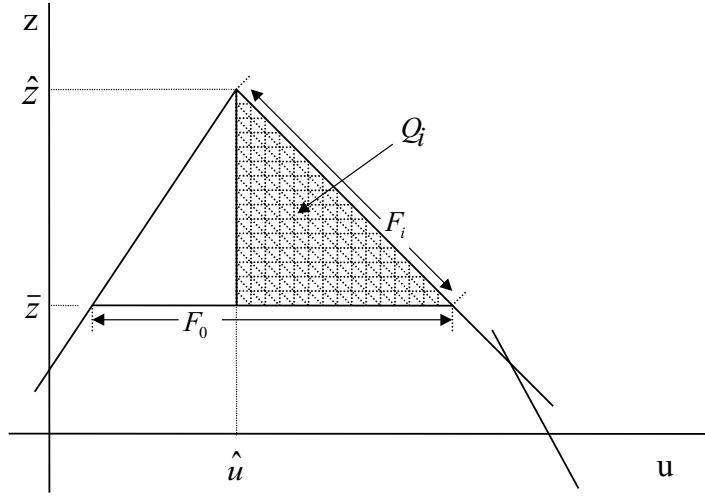$$0 = \int_{Q_i} e_1 . \vec{n} dS = \int_{F_i} n_1(F_i) dS + \int_{\bar{F}_0} n_1(F_0) dS,$$

Fig. 5.2:

where $\vec{n} = (n_1, n_2, \ldots, n_n)$ and $n_1(F_i) = \frac{1}{\|(1,\ a^i)\|}$, $\quad n_1(F_0) = -1$. Thus, it follows

$$\frac{1}{\|(1,\ a^i)\|} \int_{F_i} dS_i - \int_{\bar{F}_0} dS \;=\; 0.$$

This implies

$$A_i = \frac{\delta_i}{\|(1,\ a^i)\|},$$

where $A_i = \int_{\bar{F}_0} dS$, the surface area of $\bar{F}_0$.

Let $h = \hat{z} - \bar{z}$. Then the volume of Q_i is $\gamma_i = \frac{1}{2} h A_i$ or $A_i = \frac{2}{h} \gamma_i$. Since $\delta_0 = \sum_{j=1}^{m'} A_j$, we have (see (5.10))

$$\alpha_i = \frac{\delta_i}{\delta_0 \|(1,\ a^i)\|} = \frac{A_i}{\delta_0} = \frac{A_i}{\sum_{j=1}^{m'} A_j} = \frac{\frac{2}{h}\gamma_i}{\frac{2}{h}\sum_{j=1}^{m'} \gamma_j} = \frac{\gamma_i}{\sum_{j=1}^{m'} \gamma_j}.$$

This completes the proof. $\qquad \square$

The Volume and Duality Theorem suggests that given a vector $(\bar{z}, \bar{u})$ one should look at the active faces, and compute the volume of their projection over the hyperplane $z = \bar{z} - \epsilon$, on a neighborhood of $\bar{u}$, and for some small positive value of ϵ. Let α_i be the ratio of the volume below F_i to the total. Then one should

compute

$$(1, \underline{0}) - \sum_{i=1}^{m'} \alpha_i(1, a^i), \qquad \text{where } \underline{0} = (0, 0, \ldots, 0).$$

If this is zero we have a proof of optimality, otherwise we have a step direction and the procedure needs to continue. The Volume Algorithm which will be presented in this section computes approximations to these volumes within the subgradient scheme.

Example 5.1:

$$\begin{aligned}
\max \quad & x_2 \\
\text{s.t} \quad & x_2 - x_1 \leq 5 \\
& x_2 + \tfrac{1}{2}x_1 \leq 14 \\
& x_2 + 2x_1 \leq 32 \\
& x_2 \in \mathbb{R}, \quad x_1 \geq 0
\end{aligned}$$

Solution: Clearly, the solution of this problem is $(\hat{x}_1, \hat{x}_2) = (6, 11)$. We would like to solve the dual problem using the Volume and Duality Theorem. (See Figure 5.3). Let $\alpha_1, \alpha_2, \alpha_3$ be the dual variables. Note that $\alpha_3 = 0$ since the third constraint is not active at the optimal primal solution. But

$$\alpha_i = \frac{\gamma_i}{\gamma_1 + \gamma_2}, \qquad i = 1, 2$$

where $\gamma_1 = \tfrac{1}{2}\epsilon^2 = \tfrac{1}{2}\epsilon^2$ and $\gamma_2 = \tfrac{1}{2}\epsilon(2\epsilon) = \epsilon^2$. Hence, $\alpha_1 = 1/3$, $\alpha_2 = 2/3$ and $\alpha_3 = 0$ is the optimal dual solution.
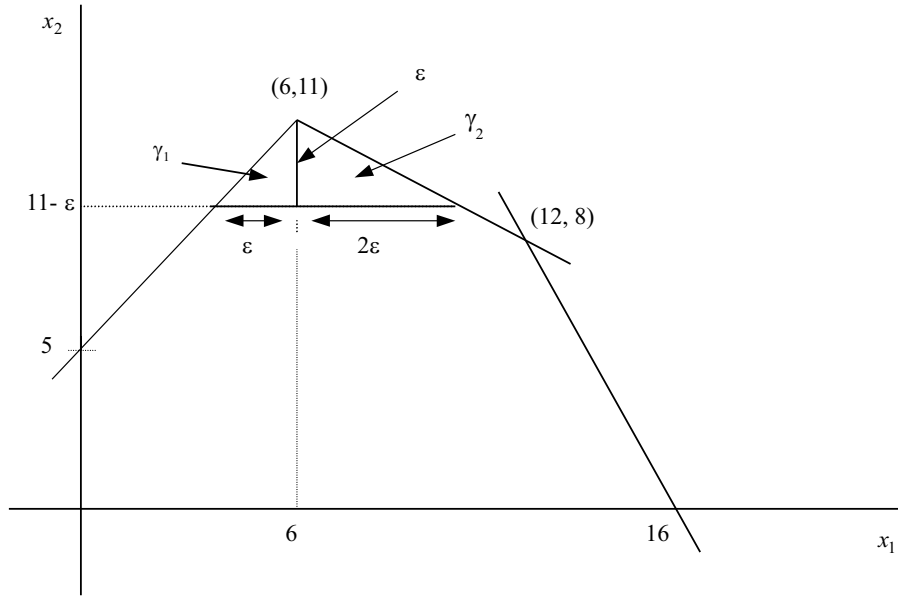
$\square$

Fig. 5.3: graphical illustration of Example 5.1.

Next we describe an extension of the subgradient algorithm, due to Barahona and Anbil [6], which produces both the solution of the Lagrangian dual and an approximation to a near-optimal primal solution based on the Volume and Duality Theorem.

Algorithm 5.1: **Volume Algorithm**

Step 0. (Initialization): Start with a vector $\bar{u} := u^0$ and solve the corresponding subproblem (5.5). Let $\bar{x}$ be an optimal solution of this subproblem. Set $x^0 = \bar{x}, \ \bar{\phi} := \phi(u^0), \ k = 1$.

Step 1. Determine a direction of motion, $d^k = b - A\bar{x}$; and

$u^k = P_{(\mathbb{R}^m_+)}(\bar{u} + \lambda_k d^k)$, for a step-size λ_k given by (5.13).

Solve the subproblem (5.5) with u^k, and let x^k and $\phi(u^k)$ be the solutions obtained.

Then $\bar{x}$ is updated as

$$\bar{x} := \alpha x^k + (1 - \alpha)\bar{x} \tag{5.11}$$

for some $\alpha \in (0, 1)$, to be discussed later.

Step 2. If $\phi(u^k) > \bar{\phi}$, update $\bar{u}$ and $\bar{\phi}$ as

$$\bar{\phi} := \phi(u^k), \quad \bar{u} := u^k$$

Let $k := k + 1$, and goto Step 1.

At termination of the Volume Algorithm, $\bar{u}$ would be an output for the solution of the Lagrangian dual with the objective value $\bar{\phi}$. Notice that in Step 2 we update only if $\phi(u^k) > \bar{\phi}$, so that this is an ascent method. The other difference with an ordinary subgradient algorithm is the formula (5.11). If $x^0, x^1, \ldots, x^k$ is the sequence of vectors produced by solving the subproblem (5.5), then

$$\bar{x} = (1 - \alpha)^k x^0 + (1 - \alpha)^{k-1} \alpha x^1 + \ldots + (1 - \alpha) \alpha x^{k-1} + \alpha x^k.$$

For $\alpha \in (0, 1)$, obviously each of the coefficients of x^i, $i = 0, 1, \ldots, k$ is between 0 and 1 and their sum is equal to 1, i.e,

$$\alpha \sum_{i=0}^{k-1} (1 - \alpha)^i + (1 - \alpha)^k = 1, \tag{5.12}$$

which can be shown by induction on k, for $k = 1, 2, \ldots$. Thus, $\bar{x}$ is a convex combination of $\{x^0, x^1, \ldots, x^{k-1}, x^k\} \subseteq \mathbb{X}$. The assumption that this sequence estimates an optimal solution of (5.2) or the corresponding coefficients $(1 - \alpha)^k, (1 - \alpha)^{k-1} \alpha, \ldots, (1 - \alpha) \alpha, \alpha$ approximate the solution of (5.3) is based on Theorem 5.1. Notice the exponential decrease of the coefficients of this convex combination, thus the later vectors receive a much larger weight than the earlier once.

At each iteration of the Volume Algorithm, the direction of move is determined as a convex combination of a current subgradient and the previous direction. That is, if d^k is the direction of move at the *k-th* iteration, then

$$d^k = \alpha s^k + (1 - \alpha) d^{k-1},$$

where $s^k \in \partial\phi(u^k)$, $k = 1, 2, \ldots$, $d^0 = s^0$, since (for $\bar{x}^k := \bar{x}$, at the k-th iteration)

$$
\begin{aligned}
d^k &= b - A\bar{x}^k \\
&= b - A[\alpha x^k + (1 - \alpha)\bar{x}^{k-1}] \\
&= \alpha(b - Ax^k) + (1 - \alpha)(b - A\bar{x}^{k-1}) \\
&= \alpha s^k + (1 - \alpha)d^{k-1}.
\end{aligned}
$$

Therefore, at every iteration the direction is updated as in the deflected subgradient method, so this is a method that would not have the same zigzagging behavior of the pure subgradient and it is in this sense that we consider the Volume Algorithm as an extension of the deflected subgradient procedure.

As in the case of the subgradient methods, the Polyak's (relaxation) step length rule

$$
\lambda_k = \mu_k \frac{(UB - \bar{\phi})}{\|d^k\|^2} \tag{5.13}
$$

is used where $0 < \mu_k < 2$. In order to set the value of μ_k, Barahona and Anbil [6] proposed the followings: Define three types of iterations as follows.

- Each time that we do not find an improvement of the objective value we call this iteration *red*. A sequence of red iterations suggests the need for a smaller step-size.

- If $\phi(u^k) > \bar{\phi}$, we compute

$$
\theta = d^k s^k
$$

  where d^k is a determined direction of move and $s^k = b - Ax^k \in \partial\phi(u^k)$.

  - If $\theta < 0$ it means that a longer step in the direction of d^k would have given a smaller value for $\phi(u^k)$, we call this iteration *yellow*.

  - If $\theta \geq 0$ we call this iteration *green*. A green iteration suggests the need for a larger step size.

Then, the strategy to determine μ_k is

$$\mu_k = \begin{cases} (0.66)\mu_{k-1}, & \text{after a sequence of 20 red iterations} \\ (1.1)\mu_{k-1}, & \text{if } k\text{-}th \text{ iteration is green.} \\ \mu_{k-1}, & \text{otherwise.} \end{cases}$$

In order to determine the value of α in (5.11), Barahona and Anbil [6] suggested to set it to a fixed value for a number of iterations and decreases afterwards; or to set it by solving the following 1-dimensional problem:

$$\min \quad \|b - A(\alpha x^k + (1-\alpha)\bar{x})\| \qquad (5.14)$$
$$\text{s.t} \quad \frac{a}{10} \leq \alpha \leq a$$

The value of a is originally set to 0.1 and its value would be decreased near the end. In [8], it is reported that a good success can be obtained by setting the value of a originally to 0.1 and then after every 100 iterations check if $\phi(u^k)$ had increased by at least 1%, if not divide a by 2. When a becomes less than 10^{-5}, it is kept constant. This choice of α bears a great similarities with the one proposed in [83].

At any iteration if $d^k = b - A\bar{x} \leq 0$, then $\bar{x}$ is a feasible solution of the primal problem (5.2) since $\bar{x} \in \text{conv}(\mathbb{X})$ and the dualized constraints are also satisfied. If $d^k = b - A\bar{x} \leq 0$ and $\bar{u}(b - A\bar{x}) = 0$, then by Theorem 2.9 $\bar{x}$ is an optimal solution of (5.2) and $\bar{u}$ is an optimal solution of the Lagrangian dual and we stop the procedure. Otherwise, if it becomes difficult to reach that, the algorithm should stop when $|b - A\bar{x}|$ and $|c\bar{x} - \bar{\phi}|$ are both below a certain threshold.

5.2.1   Similarities with the Bundle Method

We have seen that the Volume Algorithm (VA) constructs two sequences:

1. sequence of primal variables $\{\bar{x}^k\}$,   and

2. sequence of dual variables $\{\bar{u}^t\}$,   which are special subsequence of $\{u^k\}$, where the dual function ϕ get improved.

If we look at only the way dual variables are constructed, the VA shows some similarities with the bundle method (BM). As mentioned above, the VA is an ascent procedure (see Step 2) in the same way as a BM. In the VA Step 1 is referred as a *minor iteration* and Steep 2 a *major iteration*. The major iteration does update $\bar{u}^t$ only when there is an improvement in objective value. Otherwise, using the minor iteration, the procedure moves in the neighborhood of $\bar{u}^t$ to find a preferable direction or a tentative iterate point. In other word, the minor iteration provides a new subgradient vector at some point u^k in the neighborhood of $\bar{u}^t$, say $s^k \in \partial\phi(u^k)$, and constructs a trial step direction $d^k = \alpha s^k + (1-\alpha)d^{k-1}$. Indeed,

$$d^k = (1-\alpha)^k s^0 + (1-\alpha)^{k-1}\alpha s^1 + \ldots + (1-\alpha)\alpha s^{k-1} + \alpha s^k.$$

where $s^k \in \partial\phi(u^k)$, for each k, and the coefficients are all nonnegative with sum equal to 1 (see, 5.12). Thus, also the VA is utilizing all subgradients assembled during the previous iterations as convex combination in order to determine its current trial step direction. If this trial direction d^k has lead from $\bar{u}^t$ to a tentative dual point u^{k+1} where $\phi(u^{k+1}) > \phi(\bar{u}^t)$, then $\bar{u}^t$ is updated to u^{k+1}; otherwise the major iteration does nothing. Hence, we might have a sequence of minor iterations before finding an improvement (or update at the major iteration) occurs. The minor iteration in the VA resembles the null step in BM while the major iteration corresponds to the serious step of the BM.

There is also differences between the VA and BM. In a typical BM, we have a measure of improvement of objective value (see, (3.33)). This is a crucial distinction since the major iteration in the VA does not measure the gain obtained. Specifically, when passing from $\bar{u}^t$ to $\bar{u}^{t+1}$, it is not known how much "better" the serious step is - we only know that $\phi(\bar{u}^{t+1}) > \phi(\bar{u}^t)$. Recently, Bahiense et. al. have addressed this issue in more details in their paper on a revised version of the VA [5].

5.3 Ergodic Primal Solution within the Conditional Subgradient Method

We shall now extend the conditional subgradient method to find a near-optimal primal solution to the IP. The conditional subgradient algorithm for the Lagrangian dual problem, Algorithm 3.3, produces a sequence $\{x^k\}$ of "primal" solutions to the subproblem. As mentioned above, there is no guarantee for convergence of this sequence to an optimal (or near-optimal) solution of IP. We utilize an ergodic (averaged) sequence to find a near-optimal solution to IP. The ergodic sequence which we will consider is a sequence with elements that are weighted averages of those of sequence of solutions of the subproblems.

Definition 5.2: (**Ergodic Sequence**)

Suppose a sequence $\{y^k\}$, $k = 0, 1, 2, ...$, is given. A sequence $\{\bar{y}^t\}$, $t = 1, 2, 3, ...$, is called an *ergodic sequence* of the given sequence if each element $\bar{y}^t$ is a weighted average of the first t terms of the given sequence; i.e.,

$$\bar{y}^t = \sum_{k=0}^{t-1} \omega_{tk} y^k, \quad \sum_{k=0}^{t-1} \omega_{tk} = 1, \quad \omega_{tk} \geq 0, \quad \forall k, t. \tag{5.15}$$

Example: If

$$\begin{aligned} \{y^k\} &= \{(1/2)^k\}, \quad k = 0, 1, 2, 3... \\ &= \{1, 1/2, 1/4, \ldots, (1/2)^k, \ldots\} \end{aligned}$$

is a given sequence, then its ergodic sequence with equal weights is given by

$$\begin{aligned} \{\bar{y}^t\} &= \{\tfrac{1}{t} \sum_{k=0}^{t-1}(1/2)^k\}, \quad t = 1, 2, 3..., \\ &= \{1, 3/4, 7/12, \ldots, \tfrac{2}{t}(1 - 1/2^t), \ldots\}. \end{aligned}$$

To construct a near optimal solution of the IP we generate an ergodic sequence, with a choice of special weights, from the sequence of the solutions of the Lagrangian subproblems so that it converges to a near optimal solution. Larsson-Patriksson-Strömberg [54] proposed two schemes for generating such weighted averages of the sequence of solutions of the subproblems with in the conditional

subgradient procedure for Lagrangian dual of a convex programming problem and also proved the convergence of each of these sequences to an optimal solution of the primal problem. We will utilize this idea to construct a near-optimal solution to IP within the conditional subgradient procedure. An ergodic sequence $\{\bar{x}^t\}$ of the solutions of the Lagrangian subproblem is defined by

$$\bar{x}^t = l_t^{-1} \sum_{k=0}^{t-1} \lambda_k x^k, \qquad l_t = \sum_{k=0}^{t-1} \lambda_k, \quad t = 1, 2, ... \tag{5.16}$$

where $x^k \in \mathbb{X}(u^k)$, i.e., a solution to the Lagrangian subproblem (2.16) at u^k, and λ_k is a step length used in the conditional subgradient scheme (Algorithm 3.3). Note that the weights $w_{tk} = l_t^{-1}\lambda_k, \quad \forall k, t$ are in accordance with the weights in the definition of ergodic sequence, (5.15).

We will show that each of the accumulation point of the ergodic sequence $\{\bar{x}^t\}$ defined by (5.16) is an optimal solution to the primal problem (5.2). The next lemma, whose proof can be found in, e.g., ([52], Theorem 2, p. 35), (presented in [54]), will be used in the following discussion.

Lemma 5.2: Assume the sequence $\{w_{tk}\} \subseteq \mathbb{R}$ fulfills the conditions

$$\omega_{tk} \geq 0, \quad k = 0, 1, \ldots, t-1; \quad \sum_{k=0}^{t-1} \omega_{tk} = 1, \quad t = 1, 2...;$$
$$\text{and} \quad \lim_{t\to\infty} \omega_{tk} = 0, \quad k = 0, 1, 2, ...$$

If the sequence $\{y^k\} \subseteq \mathbb{R}^m$ is such that $\lim_{k\to\infty} y^k = y$, then

$$\lim_{t\to\infty} \left(\sum_{k=0}^{t-1} \omega_{tk} y^k\right) = y.$$

$\square$

Lemma 5.2 means that if the weights of the ergodic sequence of a given sequence satisfy the given conditions, then the ergodic sequence converges to the limit point of the original sequence, provided such limit exists.

Lemma 5.3: Suppose the sequence $\{\bar{x}^t\}$ be the ergodic sequence given by the definition (5.16). If $\bar{x}$ is its accumulation point, then $\bar{x} \in \text{conv}(\mathbb{X})$.

Proof: Since $\omega_{tk} = l_t^{-1}\lambda_k \geq 0$, $\sum_{k=0}^{t-1} \omega_{tk} = 1$ and $x^k \in \mathbb{X}(u^k) \subseteq \text{conv}(\mathbb{X})$, it holds that $\bar{x}^t = \sum_{k=0}^{t-1} \omega_{tk} x^k \in \text{conv}(\mathbb{X})$, $\forall t$. Hence, from closeness of $\text{conv}(\mathbb{X})$ it follows that $\bar{x} \in \text{conv}(\mathbb{X})$. $\qquad\square$

Recall that under the Slater constraint qualification (or else from the fact that the strong duality holds for problem (5.2)) the solution to the primal problem (5.2), which is the convex relaxation of the IP (5.1), may be expressed as (see, Theorem 2.9)

$$\mathbb{X}_{cp}^* = \{x \in \mathbb{X}(u): \ b - Ax \leq 0, \ \ u(b - Ax) = 0 \} \qquad (5.17)$$

irrespective of the choice of $u \in \Omega^*$ and the primal-dual optimality conditions may be expressed as(see, Theorem 3.15(b))

$$(x, u) \in \mathbb{X}_{cp}^* \times \Omega^* \ \Leftrightarrow \ s(x) \in \partial\phi(u) \cap \text{N}_{\mathbb{R}_+^m}(u) \qquad (5.18)$$

where $s(x) = b - Ax$ is a subgradient of the dual function at u.

In the next theorem, the convergence of the ergodic sequence $\{\bar{x}^t\}$ to the set $\mathbb{X}_{cp}^*$ is established in terms of the fulfilment of the optimality conditions of Theorem 2.9.

Theorem 5.4: If the step size λ_k in the conditional subgradient procedure (Algorithm 3.3) applied to the Lagrangian dual problem (LD) is given by (3.24) and the ergodic sequence $\{\bar{x}^t\}$ is given by the definition (5.16), then

$$\lim_{t \to \infty} \min_{x \in \mathbb{X}_{cp}^*} \|\bar{x}^t - x\| = 0.$$

Proof: We want to show, for any limit point $\bar{x}$ of the ergodic sequence $\{\bar{x}^t\}$, $\bar{x} \in \mathbb{X}(\bar{u})$, $b - A\bar{x} \leq 0$, and $\bar{u}(b - A\bar{x}) = 0$ for some $\bar{u} \in \Omega^*$. Note that for any limit point $\bar{x}$ of $\{\bar{x}^t\}$, it holds that $\bar{x} \in \text{conv}(\mathbb{X})$ (Lemma 5.3). Let $\bar{u} \in \Omega^*$ be the limit of the sequence of iterates generated by the conditional subgradient scheme (Algorithm 3.3), whose existence is guaranteed by Theorem 3.17. Note that the

inequalities

$$0 \leq \text{dist}(\bar{x}^t, \mathbb{X}(\bar{u})) \leq \sum_{k=0}^{t-1} \omega_{tk} \text{dist}(x^k, \mathbb{X}(\bar{u})) \tag{5.19}$$

where $\omega_{tk} = l_t^{-1} \lambda_k$, for all t, hold since the first inequality follows from nonnegativity of the distance function and the second inequality follows from the fact that the distance function $\text{dist}(., \mathbb{X}(\bar{u}))$ is a convex function. By Theorem 2.8 and the convergence of $\{u^t\}$ to $\bar{u}$ we obtain

$$\{\text{dist}(x^k, \mathbb{X}(\bar{u}))\} \longrightarrow 0, \quad \text{as} \quad k \longrightarrow \infty. \tag{5.20}$$

Observe also that

$$\lim_{t \to \infty} \omega_{tk} = \lim_{t \to \infty} \frac{\lambda_k}{\sum_{k=0}^{t-1} \lambda_k} = 0, \quad \sum_{k=0}^{t-1} \omega_{tk} = \sum_{k=0}^{t-1} \frac{\lambda_k}{l^t} = 1, \quad \omega_{tk} \geq 0, \quad \forall t, k$$

and hence, using Lemma 5.2, with $y^k = \text{dist}(x^k, \mathbb{X}(\bar{u}))$ and $y = 0$, it is then follows from (5.19)and (5.20) that

$$0 \leq \text{dist}(\bar{x}^t, \mathbb{X}(\bar{u}) \leq \sum_{k=0}^{t-1} \omega_{tk} \text{dist}(x^k, \mathbb{X}(\bar{u})) \longrightarrow 0, \quad \text{as} \quad t \longrightarrow \infty,$$

It follows from the convexity and closeness of $\mathbb{X}(\bar{u})$ that for any limit point $\bar{x}$ of $\{\bar{x}^t\}$ we have

$$\bar{x} \in \mathbb{X}(\bar{u}). \tag{5.21}$$

Next we show that for any limit point $\bar{x}$ of the ergodic sequence $\{\bar{x}^t\}$, $\bar{x}$ is feasible in the primal problem (5.2). From Lemma 5.3, $\bar{x} \in \text{conv}(\mathbb{X})$. So it remains to show that $s(\bar{x}) = b - A\bar{x} \leq 0$. Since s is affine function we have,

$$s(\bar{x}^t) = l_t^{-1} \sum_{k=0}^{t-1} \lambda_k s(x^k) \quad \forall t,$$

and from the iteration formula in the conditional subgradient procedure,

$$\begin{aligned} u^{k+1} = {} & P_{\mathbb{R}_+^m}[u^k + \lambda_k(s(x^k) - v(x^k))], \quad \text{where} \quad v(x^k) \in N_{\mathbb{R}_+^m}(x^k) \\ \geq {} & u^k + \lambda_k(s(x^k) - v(x^k)) \quad \text{from (3.32).} \\ \geq {} & u^k + \lambda_k s(x^k), \quad \text{since, from (3.31)}, s(x^k) - v(x^k) \geq s(x^k) \end{aligned}$$

From which follows that

$$s(x^k) \leq \frac{u^{k+1} - u^k}{\lambda_k}, \quad \forall k. \tag{5.22}$$

From this and the expression for $s(\bar{x}^t)$ we get

$$s(\bar{x}^t) \leq l_t^{-1} \sum_{k=0}^{t-1} \lambda_k \frac{u^{k+1} - u^k}{\lambda_k} = \frac{u^t - u^0}{l_t}, \quad \forall t.$$

Theorem 3.17 implies that the sequence $\{u^t - u^0\}$ is bounded, and therefore

$$\limsup_{t \to \infty} s_i(\bar{x}^t) \leq \lim_{t \to \infty} \frac{u^t - u^o}{l_t} = 0, \quad \forall i \in I = \{1, 2, \dots, m\},$$

since $\lim_{t \to \infty} l_t = \sum_{k=0}^{\infty} \lambda_k = \infty$. Therefore,

$$b - A\bar{x} = s(\bar{x}) \leq 0. \tag{5.23}$$

Now we want to show $\bar{u}s(\bar{x}) = \bar{u}(b - A\bar{x}) = 0$. Consider an $i \in I(\bar{u}) = \{i : \bar{u}_i > 0\}$. As $\bar{u}$ is a limit point of u^k, it follows that, for some fixed but sufficiently large k_1, $u_i^k > 0 \;\; \forall k \geq k_1$, where $\{u^k\}$ is the sequence of dual solutions generated by the conditional subgradient procedure (Algorithm 3.3). Therefore, from (3.31), it follows that

$$\tilde{s}_i^k = b_i - A^i x^k = s_i^k$$

which implies

$$u_i^{k+1} = \max\left\{ 0, \; u_i^k + \lambda_k \tilde{s}_i^k \right\} = \max\left\{ 0, \; u_i^k + \lambda_k s_i^k \right\}, \quad \forall k \geq k_1.$$

Moreover, as $u_i^k > 0 \;\; \forall k \geq k_1$, one can find a sufficiently small $\lambda_k > 0$ such that $u_i^k + \lambda_k s_i^k > 0$. Thus, since $\lambda_k \longrightarrow 0$ as $k \longrightarrow \infty$, there exists a sufficiently large $k_2 > 0$ such that $\forall k \geq k_2$ we get $u_i^k + \lambda_k s_i^k \geq 0$. That is,

$$
\begin{aligned}
u_i^{k+1} &= u_i^k + \lambda_k s_i^k \quad \forall k \geq \max\{k_1, k_2\} \\
\Rightarrow \quad s_i^k &= \frac{u_i^{k+1} - u_i^k}{\lambda_k} \quad \forall k \geq \max\{k_1, k_2\}
\end{aligned}
\tag{5.24}
$$

Now

$$s_i(\bar{x}^t) = l_t^{-1} \sum_{k=0}^{t-1} \lambda_k \frac{u_i^{k+1} - u_i^k}{\lambda_k} = \frac{u_i^k - u_i^0}{l_t}.$$

Therefore, by taking account of boundedness of the sequence $\{u^k - u^0\}$ and the $\lim_{t\to\infty} l_t = \infty$, we obtain $\lim_{t\to\infty} s_i(\bar{x}^t) = 0$. Since this result holds for all $i \in I(\bar{u})$, and by definition $\bar{u}_i = 0 \;\; \forall i \in I \setminus I(\bar{u})$, it follows that as $t \longrightarrow \infty$,

$$\{\bar{u}s(\bar{x}^t)\} \longrightarrow 0. \tag{5.25}$$

Thus, from (5.21), (5.23), and (5.25) it follows that at the limit the ergodic sequence is feasible and satisfies the optimality conditions together with any $\bar{u} \in \Omega^*$. $\qquad\square$

Corollary 5.5: ($\bar{x}^t$ verifies the optimality in the limit):

Under the assumption of Theorem 5.4,

$$\{\text{dist}(s(\bar{x}^t), \partial\phi(\bar{u}) \cap N_{\mathbb{R}^m_+}(\bar{u}))\} \longrightarrow 0.$$

The proof of the corollary follows from Theorem 3.17, Theorem 5.4, and the relation (5.18). $\qquad\square$

An alternative ergodic sequence, with equal weights on all subproblem solutions, is given by

$$\hat{x}^t = \frac{1}{t} \sum_{k=0}^{t-1} x^k, \quad t = 1, 2, \ldots \tag{5.26}$$

where, as before, $x^k \in \mathbb{X}(u^k)$. Analogously to Theorem 5.4, one can show the convergence of the sequence $\{\hat{x}^t\}$ to the optimal solution set of the primal problem (5.2) or the near-optimal solution of the IP, given that the step size λ_k in the conditional subgradient scheme is chosen according to the adaptive step size selection rule (Corollary 3.18), i.e.,

$$\lambda_k \in [\frac{\alpha}{\beta + k}, \frac{M}{\beta + k}], \tag{5.27}$$

where $\beta > 0, \;\; 0 < \alpha \leq M < \infty, \quad k = 0, 1, 2, \ldots$.

The next theorem whose detailed proof is found in ([54], Theorem 2) can be justified with arguments analogous to the proof of Theorem 5.4 with $\omega_{tk} = 1/t$, for each $t = 1, 2, 3, \ldots$.

Theorem 5.6: (**convergence of** $\{\hat{x}^t\}$ **to the solution set** $\mathbb{X}_{cp}^*$)

Suppose the conditional subgradient scheme (Algorithm 3.3) is applied to solve the Lagrangian dual problem of IP (5.1) with a step size determined as (5.27), the set $\mathbb{X}_{cp}^*$ and the sequence $\{\hat{x}^t\}$ are given by the definitions (5.17) and (5.26), respectively, and suppose that $\{v^t\}$ is bounded. Then,

$$\{ \mathrm{dist}(\hat{x}^t, \mathbb{X}_{cp}^*) \} \longrightarrow 0.$$

$\square$

The above theorem implies directly the following corollary.

Corollary 5.7: ($\hat{x}^t$ **verifies the optimality in the limit**)

Under the assumption of Theorem 5.6,

$$\{\mathrm{dist}(s(\hat{x}^t), \partial\phi(\bar{u}) \cap \mathrm{N}_{\mathbb{R}_+^m}(\bar{u}))\} \longrightarrow 0.$$

$\square$

As a consequence of the results of this section, one can directly design a primal-dual iterative procedure that can provide a near optimal primal solution to the IP as well as an optimal solution of its Lagrangian dual by incorporating the ergodic sequence of solutions of subproblems given by either (5.16) or (5.26) into the conditional subgradient procedure (Algorithm 3.3). Note that the ergodic sequence $\bar{x}^t$ in (5.16) can be written recursively using the following formula:

Given u^t, any $x^t \in \mathbb{X}(u^t)$, and step length $\lambda_t, \quad t = 0, 1, 2, \ldots$, let

$$
\begin{aligned}
\bar{x}^1 &:= \quad x^0, \quad l_1 := \lambda_0; \\
\bar{x}^{t+1} &:= \quad \frac{l_t}{l_{t+1}}\bar{x}^t + \frac{\lambda_t}{l_{t+1}}x^t, \quad \text{where } l_{t+1} := l_t + \lambda_t, \quad t = 1, 2, \ldots
\end{aligned}
\tag{5.28}
$$

Similarly, the ergodic sequence $\hat{x}^t$ in (5.26) can be expressed recursively using the following formula:

$$
\begin{aligned}
\hat{x}^1 &:= \quad x^0; \\
\hat{x}^{t+1} &:= \quad \frac{t}{t+1}\hat{x}^t + \frac{1}{t+1}x^t, \quad t = 1, 2, \ldots
\end{aligned}
\tag{5.29}
$$

where $x^t \in \mathbb{X}(u^t)$.

6. MINIMIZATION OF TREATMENT TIME OF A RADIATION THERAPY

Radiation therapy refers to the use of radiation as a means of treating a cancer patient. In the process of radiotherapy, a beam from a linear accelerator is modulated using a multileaf collimator (MLC) in order to define a series of beam shapes, known as *segments*, which are superimposed to produce a desired fluency pattern (or intensity function) on a target area. After discretization, an intensity function that been given on a cross-sectional target areas can be described as an $m \times n$ intensity matrix. The implementation of the intensity matrix, i.e, delivery of the radiation dose, within a short possible time is one of the important goals in a radiotherapy planning. The optimization problem we consider in this thesis is to determine a suitable sequence of leaf settings of MLC in order to minimize total delivery time. Two main important objectives to accomplish this are to minimize the total beam-on-time and the number of segments. Boland, Hamacher and Lenzen [16] have developed an exact method for the problem of minimizing the total beam-on-time using a side-constrained network flow model. However, minimizing the beam-on-time alone cannot minimize the total treatment time since the procedure under consideration requires set-up time during each beam-off time, i.e., between each consecutive segments of the beam. Hence, we need also to minimize the number of segments. Unfortunately, the problem of minimizing the number of segments is NP-hard [20]. The heuristic algorithm of Xia and Verhey [85] is the best available thus far for the problem of minimizing the number of segments.

The optimization problem of total delivery time, where delivery time depends

on both beam-on time and number of segments, is NP-hard. We introduce a new fast and efficient algorithm which combines exact and heuristic procedures with the objective to minimize the total delivery time. In particular, with the objective of minimizing the number of segments, we construct a heuristic algorithm which involves minimizing beam-on-time as a subproblem. We solve the subproblem using the Hamacher-Boland side constrained network flow model. This side-constrained network flow model usually becomes a large scale problem and requires a large amount of computational time since it involves a large number of arcs (variables) to model a practical problem. We use the subgradient optimization methods in order to overcome the difficulties with the problem of large instances. In particular, the Lagrangian dual technique is used to remove the complicating side constraints of the network so that the relaxed network subproblem becomes a pure minimum cost network flow problem. We solve this pure minimum cost network flow subproblem, using negative cycle cancelling method. Solutions of this pure minimum cost network flow subproblem provide either a solution to the original problem or a subgradient direction at each subgradient iteration point.

This chapter is organized as follows. We present some backgrounds in Section 6.1. Section 6.2 deals with the decomposition of intensity matrix (radiation dose) into segments (deliverable radiation beams). In this section, we also describe associated physical constraints of the MLC which would be used to deliver the radiation to a target area. In Section 6.3 we present the Hamacher-Boland network flow model of the MLC. A new algorithm intended to minimize the total delivery time of the radiation therapy problem is introduced in 6.4. In section 6.5 we present some numerical tests. Application of subgradient methods in the numerical implementation of large scale-scale problems show a tremendous reduction of computational time as well as much less requirement of memory space as compared to other currently known methods.

6.1 Introduction

Radiation therapy concerns with the delivery of the proper dose of radiation to a tumor without causing damage to a surrounding healthy tissue and critical organs. Since a sufficiently high dose of radiation can kill cancer cells (tumor), external beam of radiation is often used to treat a cancer patient. However, difficulties and risks are associated with this technique since such a high dose of radiation can kill also a normal or healthy tissue surrounding the tumor. Therefore, it is highly required to plan a treatment carefully so that the radiation beams are focused in such a way that they deposit enough dose of radiation energy into a tumor but do not deposit an abundance of radiation into organs at risk or normal tissues. Thus, it is a crucial task in a clinical radiation treatment planning to realize on one hand a high level dose of radiation in the cancer tissue in order to obtain a maximum tumor control; and on the other hand, it is absolutely necessary to keep the radiation into the tissue outside of the tumor as low as possible in order to spare the health and functions of the organs after the treatment. Because of such conflicting objectives in the radiation treatment planning, Hamacher and Küfer [43] have dealt with the problem using a multiple objective optimization approach. Such an approach usually starts with a given or desired dose distribution and determine the radiation field that can provide the specified dose distribution in the patient field - thus called the *inverse treatment planning method*. A number of other solution methods have also been proposed to solve the inverse treatment planning problem. Depending on the nature of a chosen objective function, determination of a solution of the inverse treatment planning problem has been performed using various optimization techniques including minimizing deviations from given bounds using least square or quadratic programming approach (Burkard et al. [19], Shepard et al. [76]), linear programming (Rosen et al. [71], Morill et al. [62], Holder [49]), mixed integer programming (Lee et al.[56]) and multiple objective optimization (Hamacher and Küfer [43]).

The output of an inverse treatment formalism is usually an $m \times n$ matrix with non-negative entries, called *intensity matrix*. Subsequently, the resulting intensity matrix has to be implemented, i.e., delivered to the cancer patient treatment field by using a medical accelerator which are put into a gantry that can be rotated about a patient, who is positioned and fixed on a couch (see Fig. 6.1).

Fig. 6.1: A medical linear accelerator with beam head and a couch. A patient is being treated.
Source: http://www.varian.com/onc/prod55.html

The modern device used for this purpose is a multileaf collimator (MLC), that consists of metal pieces that can totally block any radiation. These pieces of metals are called leaves. Every two of these leaves are placed opposite to each other where each of them is connected to a linear motor by a metal band and can move in the direction towards the other leaf or away from it. Such two leaves are called a *channel* and corresponds to a row of an intensity matrix. By placing several leaf pairs one can shape a rectangular irradiation field where only the cross-sectional areas corresponding to the openings between a pair of leaves

receive radiation. For instance, given a 5×6 intensity matrix

$$
I = \begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0
\end{pmatrix},
$$

the corresponding leaves configuration of the MLC can be set as shown in the Fig. 6.2.

Fig. 6.2: The leave setting corresponding to the intensity matrix I.

Fig. 6.3 demonstrates the actual leave settings (configurations) of a MLC mounted into the head of a linear accelerator from the patient treatment field eye's view.

The beam modulation can be accomplished in two different ways by using the MLC: (1) *the dynamic mode* as described by Convey and Rosenbloom [24] and Svensson et al. [82] whereby the leaves move with a calculated, not necessarily constant, speed while the beam remains switched on in order to create the desired intensity profile; and (2) *the static mode* as described by Bortfeld et al. [17], Galvin et al. [38], Xia and Verhey [85], Siochi [81], Lenzen [59] and Boland et al. [16]. With the static mode, which is also called the "stop and shoot", the beam is switched off while the leaf pairs are being moved to the desired position. Then, keeping all of the leaf pairs at this position, the beam is switched on for a certain time in order to irradiate the sites which are not blocked by any of the MLC leaf pairs. This procedure is repeated until the required intensity profile
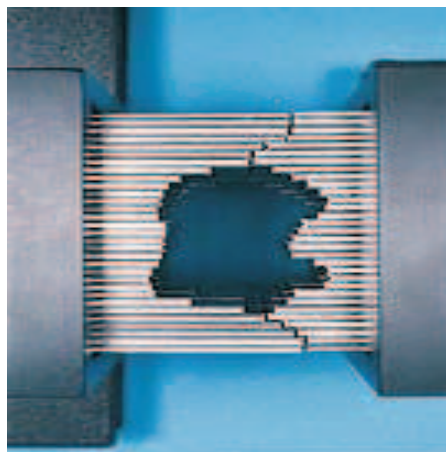
Fig. 6.3: A configuration of leaves of a MLC (from a patient treatment field eye's view).
Source: http://www.radionics.com/products/rt/mmlc.shtml

has been delivered.

For the dynamic method the goal is to minimize beam on time which is equivalent to minimizing the total treatment time. For the static method, however, minimizing beam on time alone cannot minimize the total treatment time since the procedure requires set-up time during each beam off time, i.e., between every consecutive segments of the beam. Hence, one has also to strive to minimize the total number of segments for the case of the static delivery method in order to minimize the total treatment time. Here, we consider the static delivery method and address the problem of minimizing the total number of segments.

6.2   Decomposition of Intensity Matrix into Segments

Given an intensity matrix $I \in \mathbb{Z}^{m \times n}$, which describes the radiation intensity level that the corresponding cross-sectional areas in the patient treatment filed should receive from a fixed gantry angle, one cannot directly deliver it since the MLC can produce only a uniform radiation beam at a time and also subjected to physical constraints described in this section. Therefore, we need to decompose

the intensity matrix into a deliverable binary matrices $\{S^1, S^2, \ldots, S^N\} \subseteq \mathbb{Z}^{m \times n}$ such that

$$I = \sum_{k=1}^{N} \alpha_k S^k$$

where each of the coefficients $\alpha_k > 0$ is proportional to the delivery time of the corresponding segment (shape matrix) S^k, so that

$$S_{ij}^k = S^k(i,j) = \begin{cases} 1, & \text{if the site } (i,j) \text{ is irradiated.} \\ 0, & \text{otherwise.} \end{cases}$$

Each of these binary matrices is required to be constructed in such away that it fulfills the following Physical constraints of the MLC:

1) *Consecutive 1's property.* Since only the sites between a pair of left and right leaves in a channel can be irradiated, the 1's in a row should occur consecutively. i.e., for any row i and two columns j_1, j_2 such that $j_1 < j_2$

   if $S^k(i, j_1) = 1 = S^k(i, j_2)$, then $S^k(i,j) = 1$, for all j such that $j_1 \leq j \leq j_2$. (See Fig. 6.4 (a) and (b)).

2) *Interleaf collusion constraints.* Due to the problem of collusion of neighboring leaves from opposite banks, interleaf motion is prohibited; i.e., no leaf can be extended beyond the end of its nearest neighbors in the opposite bank. That is,

   for any row i suppose $l_i \in \{0, 1, \ldots, n\}$, $r_i \in \{1, 2, \ldots, n+1\}$, $l_i < r_i$ such that

   $$S^k(i,j) = \begin{cases} 1, & \text{if } l_i < j < r_i \\ 0, & \text{otherwise} \end{cases}$$

   Then, for any $i \in \{1, 2, \ldots, m-1\}$ it should hold that

   $$l_i \leq r_{i+1} - 1 \quad \text{and} \quad r_i \geq l_{i+1} + 1. \tag{6.1}$$

Note that l_i and r_i are the columns in which the left and right leaves corresponding to the *i-th* row are positioned, respectively. The interleaf collusion

constraints are also known as the *interleaf motion constraints.* (See Fig. 6.4 (c) and (d)). In fact, there are some technologies which do not demand this physical constraints. Ahuja and Hamacher [1], for instance, are currently working on the problem of minimizing beam-on time without considering the interleaf motion constraints of a MLC.
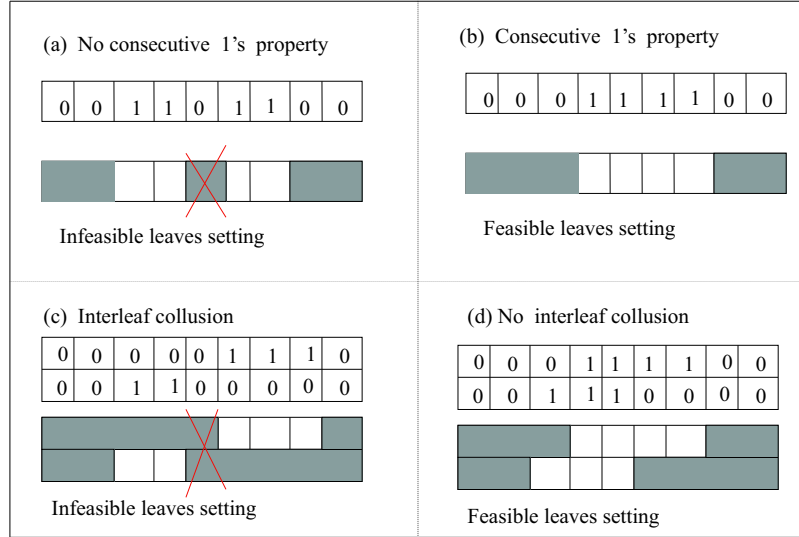

Fig. 6.4: Example of feasible and infeasible leaf settings.

Definition 6.1:

- An $m \times n$ binary matrix S, with at least one nonzero entry, satisfying both the consecutive 1's property and the interleaf motion constraints is called a *segment* or a *shape matrix.*

- A *decomposition* of an intensity matrix I is a representation of I as a positive linear combination of shape matrices (segments), i.e,

$$I = \sum_{k=1}^{N} \alpha_k S^k. \tag{6.2}$$

where S^k are segments $(k = 1, 2, \ldots, N)$ and α_k $(k = 1, 2, \ldots, N)$ are positive integers which are proportional to the required beam-on-time of the segment S^k, respectively.

- The scalars $\alpha_1, \alpha_2, \ldots, \alpha_N$ in the decomposition (6.2) are called *delivery coefficients* or *beam-on time* of the corresponding segments $S^1, S^2 \ldots, S^N$.

- Given a decomposition 6.2 of an intensity matrix I, the sum $\sum_{k=1}^{N} \alpha_k$ is said to be the *total beam-on-time* associated with the decomposition.

Example 6.1: Let

$$I = \begin{pmatrix} 0 & 3 & 1 & 4 & 1 \\ 2 & 5 & 6 & 3 & 0 \\ 2 & 5 & 3 & 1 & 0 \end{pmatrix}. \tag{6.3}$$

Then I can be decomposed, for instance, into

$$I = \sum_{k=1}^{5} \alpha_k S^k \tag{6.4}$$

where

$$S^1 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad S^2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad S^3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$S^4 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}, \quad \text{and} \quad S^5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

with the corresponding delivery coefficients $\alpha_1 = 4$, $\alpha_2 = 2$, $\alpha_3 = 2$, $\alpha_4 = 1$, and $\alpha_5 = 1$. That is, this decomposition yields 5 segments with total beam-on-time $\sum_{k=1}^{5} \alpha_k = 10$. Other possible decomposition of I (6.3) can be

$$S^1 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad S^2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}, \quad S^3 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

with the corresponding delivery coefficients $\alpha_1 = 3$, $\alpha_2 = 2$, and $\alpha_3 = 1$. In this case we have obtained only 3 segments with also smaller total beam-on-time $\sum_{k=1}^{3} \alpha_k = 6$ which is desirable in the radiation treatment processes.

The main concern of a MLC problem is to find an optimal feasible leaf settings of the MLC that can generate a suitable sequence of beam segments given by the set of binary matrices $\mathcal{S} = \{S^1, S^2, \ldots, S^N\}$ and a corresponding beam-on-time, called *delivery sequence*, $\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ such that equation 6.2 holds and each S^k satisfies both the consecutive 1's property as well as the interleaf motion constraints 6.1 with the objective to minimize the total delivery time. In general, if I is decomposed into N segments $S^1, S^2, \ldots, S^N$ with corresponding beam-on time $\alpha_1, \alpha_2, \ldots, \alpha_N$, the total delivery time (TDT) depends on the total beam on time, $\sum_{k=1}^{N} \alpha_k$, and the number of segments N. Indeed, the total delivery time is

$$\text{TDT}(\alpha_1, \ldots, \alpha_N, N) = \sum_{k=1}^{N} \alpha_k + \sum_{k=1}^{N-1} c(S^k, S^{k+1})$$

where $c(S^k, S^{k+1})$ is the time it takes to change from S^k to S^{k+1}, called *set-up time*. We assume that the set-up time is a positive constant so that $c(S^k, S^{k+1}) = \tau$. Consequently, the total set-up time will be

$$\sum_{k=1}^{N-1} c(S^k, S^{k+1}) = (N-1)\tau.$$

Hence, the optimization problem we consider here is to determine a sequence of deliverable shape matrices $S^1, S^2, \ldots, S^N$ of I and their corresponding beam-on time $\alpha_1, \alpha_2, \ldots, \alpha_N$ so as:

$$
\begin{aligned}
T^* = \min \quad & \sum_{k=1}^{N} \alpha_k + (N-1)\tau \\
\text{s.t.} \quad & \sum_{k=1}^{N} \alpha_k S^k = I \\
& S^k \text{ is deliverable shape matrices}, \alpha_k > 0, \quad \forall k.
\end{aligned}
$$

Beside being NP-hard, another difficulty associated with this problem is that we do not have a closed formula for the number of segments N which expresses its mathematical relation with S^k or α_k's − we have it only as part of the output of the problem. Different approaches have been proposed in order to tackle with this challenging problem. Two main important objectives of these are:

1. Minimize $\sum_{k=1}^{N} \alpha_k$: The total beam-on-time problem.

2. Minimize $|\mathcal{S}|$, i.e, Minimize the cardinality of $\mathcal{S}$: The problem of mini-
   mizing the total number of segments.

Various researches have been done thus far in order to achieve these objectives. Earlier heuristic methods of such works were by Bortfeld et al. [17] and Galvin et al. [38]. Comparing the early works, while the Bortfeld-Boyer algorithm provides a smaller total beam-on-time at a price of increasing the number of segments, the Galvin method provides smaller number of segments but with a larger value of total beam-on-time. Siochi [81] has aimed at finding the trade-off between the two situation in order to minimize the total delivery time. Thus far, the most effective heuristic method that can provide a smaller number of segments, in average, is that of Xia and Verhey [85] as justified by the work of Que [69]. In the Xia and Verhey procedure, each intensity level of the entries of the matrix that has a value larger than powers of two is "stripped off" first in order to extract a binary intensity matrix with the delivery coefficients(beam on time) of powers of 2. Moreover, the extracted binary intensity matrix would be further decomposed into segments using a greedy algorithm. The extraction reduces the intensity level at each iteration and the extraction procedure continues till the desired decomposition is found.

Other approach is to develop an exact method in order to use mathematical programming methods of optimization. The first work in this line was accomplished by Lenzen [59]. The mathematical programming models constructed by Lenzen for this problem involves several thousands of constraints and millions of binary variables which lead into a very large integer programming problem. More recently Boland-Hamacher-Lenzen [16] have formulated the MLC problem as a side-constrained network flow problem. In this setting, they solve a constrained minimum cost circulation problem in order to determine an exact optimal solution of the problem of the total beam-on-time. They have also proved the polynomial time solvability of the problem of minimizing the total beam-on-time.

Here we address the problem of minimizing the total number of segments, without "paying much price" of increasing the total-beam-on time, using the combination of heuristic and exact method in order to obtain smaller delivery time as much as possible. In particular, we follow the idea of the level reduction technique of Xia and Verhey to extract a binary matrix from a given intensity matrix, find an appropriate corresponding delivery coefficient which is not necessarily powers of 2, and further determine the exact minimum number of segments of the extracted binary intensity matrix using the Hamacher-Boland network model. To this end, we next introduce the Hamacher-Boland network model of the MLC problem.

6.3 The Hamacher-Boland Network Flow Model of the MLC Problem

In this section we introduce the Hamacher-Boland network flow formulation of the MLC problem in which a shape matrix (segment) can be represented by a path in the network.

In the sequel, the intensity matrix $I = (I_{ij})$ is an $m \times n$ matrix with $I_{ij} \in \mathbb{Z}$ for all $(i, j) \in \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$, unless stated otherwise. Corresponding to each row (also called a *channel*) of an intensity matrix I, a pair of metal leaves of a MLC called *left and right leaves* , are associated.

Definition 6.2: (Representation of a pair of leaves of a MLC)
For any $i \in \{1, 2, \ldots, m\}, \quad l_i \in \{0, 1, \ldots, n\},$ and $r_i \in \{1, 2, \ldots, n+1\}$, a triple (i, l_i, r_i) is said to represent a position of a pair of left and right leaves in the row (channel) i if

1. $l_i < r_i,$ and

2. the left leaf blocks radiation into the entry (cross-sectional area) (i, j) for all $j \leq l_i$ and the associated right leaf blocks radiation into the entry (i, j) for all $j \geq r_i$.

Thus, the representation (i, l_i, r_i) corresponds to a binary intensity matrix $M = (M_{ij})$ where

$$M_{ij} = \begin{cases} 1, & \text{if } l_i < j < r_i \\ 0, & \text{otherwise.} \end{cases}$$

For instance, if I is an $m \times 9$ intensity matrix, then $(i, 3, 7)$ represents the position of the left and right leaves in the row i where only the 4-th, 5-th, and 6-th columns in the row can be irradiated (see Fig. 6.5).
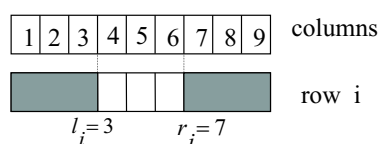

Fig. 6.5: Leaves representation where $(i, l_i, r_i) = (i, 3, 7)$.

The representation of positions of a pair of leaves given above satisfies the consecutive 1's property. In order to satisfy the interleaf motion constraints we need to have the following relations between the representations of pairs of leaves in consecutive rows.

Definition 6.3: (Admissible Representations)
Let (i, l_i, r_i) be a representation of the pair of leaves in a row i. The representations of the positions of pairs of leaves in the rows $i = 1, 2, \ldots m$ are said to be admissible if the representations (i, l_i, r_i) and $(i + 1, l_{i+1}, r_{i+1})$ of any two consecutive rows satisfy the following relations:

$$l_{i+1} \leq r_i - 1 \quad \text{and} \quad r_{i+1} \geq l_i + 1 \tag{6.5}$$

for all $i \in \{1, 2, \ldots, m - 1\}$.

The admissible representations of the pairs of leaves of a MLC satisfy the consecutive 1's property and the interleaf motion constraints of a MLC. The Hamacher-Boland network consists of nodes that basically correspond to the representations (i, l_i, r_i) and arcs that correspond to the admissibility relation 6.5, a relation which

has to be satisfied in order to fulfil the interleaf motion constraints. In particular, the network is a layered digraph $G_s = (V, E)$ consists of m layers which correspond to the m rows of the intensity matrix, where the set of main nodes V_s is given by

$$V_s = \{(i, l, r) \ : \ i \in \{1, 2, \ldots, m\}, \ l \in \{0, 1, \ldots, n\}, \ r \in \{1, 2, \ldots, n+1\}, \ l < r \ \}$$

and the set of main arcs E_s is given by

$$E_s = \{((i, l, r), (i+1, l', r')) \ : \ (i, l, r), (i+1, l', r') \in V_s, \ l' \le r - 1, \ r' \ge l + 1\}.$$

Two dummy nodes D and D' which serve as start and end nodes, respectively, be added so that

$$V = V_S \ \cup \ \{D, D'\}.$$

Moreover, the set of arcs from the start node D to the first layer, from the last layer to the end node D', and the return arc from the end node to the start node are also included so that

$$E = E_s \ \cup \ \{ \ (D, (1, l, r) \ ) \ : \ (1, l, r) \in V_s \ \}$$
$$\cup \ \{ \ (\ (m, l, r), \ D' \ ) \ : \ (m, l, r) \in V_s\} \ \cup \ \{(D', D)\}.$$

The graph G_s is called the *shape matrix graph*. An example of the shape matrix graph associated with a 3×2 intensity matrix is given in Fig. 6.6.

Note that the cycles $C_1 = D - 102 - 212 - 303 - D' - D$ and $C_2 = D - 113 - 203 - 302 - D' - D$ in the graph (Fig. 6.6) correspond to the shape matrices

$$S^1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}, \quad \text{and} \quad S^2 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix},$$

respectively.

The next Lemma given in ([16], Lemma 4.1) states some properties of the shape matrix graph G_s. The proof of the lemma is an immediate consequence of the definition of G_s.
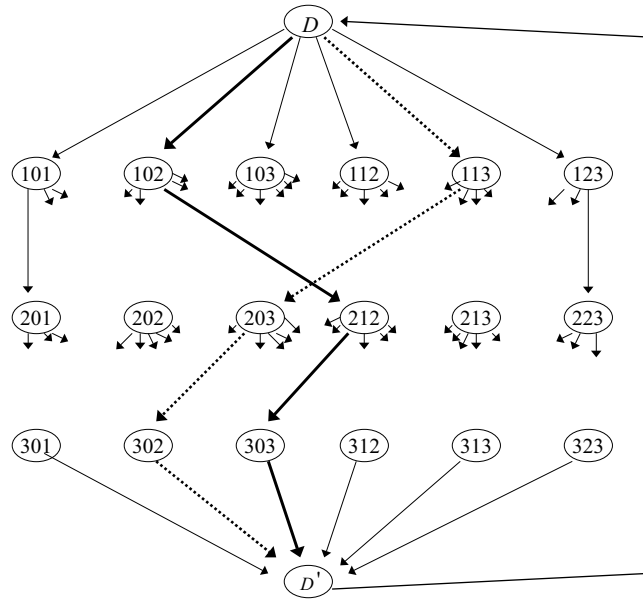
Fig. 6.6: Shape matrix graph associated with a 3×2 matrix with its nodes and some of its arcs (the remaining arcs indicated by the small arrows) including two cycles $C_1 = D - 102 - 212 - 303 - D' - D$ and $C_2 = D - 113 - 203 - 302 - D' - D$.

Lemma 6.1:

1. $G_s - \{(D', D)\}$ is an acyclic digraph.

2. Any directed cycle in G_s contains the arc (D', D).

3. Every directed cycle in G_S (or every directed path from D to D') corresponds to a shape matrix and vice versa.

$\square$

In the sequel, a cycle means a directed cycle and similarly a path means a directed path, unless stated otherwise. The shape matrix graph G_s provides all the feasible shape matrices. Indeed, the set of all paths from the start node D to the end node D' (or the set of all cycles in G_s) represents the set of all feasible shape matrices.

The shape matrix graph G_s is still needed to be expanded so as to construct a complete network flow structure of the problem. In order to complete the network, the intensity levels are included so as a flow through the network represents the required irradiation intensity level. In particular, the value of a circulation flow through a cycle represents the beam-on-time (delivery coefficient) of the shape matrix represented by the cycle. Observe that a flow entering a node (i, l, r) for some $l < r$ represents irradiation of all sites (i, j) such that $l < j < r$. In order to reflect this situation, each node $(i, l, r) \in V_s$ is splitting in to two nodes $(i, l, r)^1$ and $(i, l, r)^2$ with the idea that a flow could enter a layer (row) i only via a node of the form $(i, l, r)^1$ and may leave the layer i via the node $(i, l, r)^2$ after going through all the sites (i, j) such that $l < j < r$; i.e., $j \in \{l+1, l+2, \ldots, r-1\}$. Thus the network consists of also another set of nodes

$$IN = \{(i, j) \ : \ i = 1, 2, \ldots, m, \quad j = 0, 1, 2, \ldots, n\} \tag{6.6}$$

The elements of the set IN (6.6) are called *intensity nodes*. Thus the Hamacher-Boland network, denoted by $\mathcal{N}_{HB}$, consists of the set of nodes

$$\mathcal{N} = \{\ (i, l, r)^1,\ (i, l, r)^2\ :\ (i, l, r) \in V_s\} \ \cup\ IN \cup\ \{D, D'\}. \tag{6.7}$$

The arcs in $\mathcal{N}_{HB}$ are also defined as follows:

- The set of copies of all edges (arcs) E in G_s, i.e,

$$\hat{E} = \{\ (\ D,\ (1, l, r)^1\ )\ :\ (D,\ (1, l, r)\ ) \in E\ \}$$
$$\cup\ \{\ (\ (i, l, r)^2,\ (i+1, l', r')^1\ )\ :\ (\ (i, l, r),\ (i+1, l', r')\ ) \in E\}$$
$$\cup\ \{\ (\ (m, l, r)^2,\ D'\ )\ :\ (\ (m, l, r),\ D'\ ) \in E\ \}$$
$$\cup\ \{(D', D)\}. \tag{6.8}$$

- Other arcs incident to the nodes in IN are:

  - the set of arcs, referred as *entering arcs*:

$$EA = \{\ (\ (i, l, r)^1, (i, l)\ )\ :\ (i, l, r)^1 \in \mathcal{N},\ (i, l) \in IN\ \} \tag{6.9}$$

– the set of arcs, called *intensity arcs*:

$$IA = \{ \, (\, (i, j-1), (i, j) \,) \; : i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n \, \} \quad (6.10)$$

– the set of arcs, referred as *out-going arcs*:

$$OA = \{ \, (\, (i, r-1), (i, l, r)^2 \,) \; : (i, r-1) \in IN, \; (i, l, r)^2 \in \mathcal{N} \} \quad (6.11)$$

Thus, the set of all arcs in $\mathcal{N}_{HB}$ is given by

$$\mathcal{A} = \hat{E} \; \cup \; EA \; \cup \; IA \; \cup \; OA. \qquad (6.12)$$

The capacity of each arc is set as follows:

Let $\underline{u}_e$ and $\bar{u}_e$ be the lower and upper capacity of an arc $e \in \mathcal{A}$. Then

$$\underline{u}_e = 0 \; \text{ and } \; \bar{u}_e = \infty, \; \text{ for all } e \in \mathcal{A} - IA. \qquad (6.13)$$

$$\underline{u}_e = I_{ij} = \bar{u}_e, \; \text{ for each } \; e = (\, (i, j-1), (i, j) \,) \in IA \, . \qquad (6.14)$$

The Hamacher-Boland network flow model is aimed at formulating the problem of minimizing beam-on-time of the MLC problem as a minimum cost circulation problem. we noted that each cycle contains the arc (D', D) once and only once. As a consequence, the cost of an arc e, $c(e)$, is set to be

$$c(e) = \begin{cases} 1, & \text{if } e = (D', D). \\ 0, & \text{otherwise.} \end{cases} \qquad (6.15)$$

The other important point in the model is that a flow entering a layer (row) i through a node $(i, l, r)^1$ should leave out of the layer through the node $(i, l, r)^2$ to ensure feasibility of a cycle as a representation of a shape matrix. Hence for any

$$((i, l, r)^1, (i, l) \,) \in EA \quad \text{and} \quad (\, (i, r-1), (i, l, r)^2 \,) \in OA$$

it must hold that

$$x((i, l, r)^1, (i, l) \,) = x(\, (i, r-1), (i, l, r)^2 \,) \qquad (6.16)$$

where x is a circulation flow (in-flow equals out-flow at each node) in the network.

Thus, the *Hamacher-Boland Network* for the MLC problem is given by the network

$$\mathcal{N}_{HB} = (\ \mathcal{N}, \ \mathcal{A}, \ \underline{u}_e, \ \bar{u}_e, \ c(e)\ )$$

together with the side constraint 6.16.

Example 6.2: Let $I = \begin{pmatrix} 3 & 5 \\ 4 & 7 \\ 2 & 6 \end{pmatrix}$.

The Hamacher-Boland network corresponding to I is shown in the Fig. 6.3.

Using the network $\mathcal{N}_{HB}$ and the fact that the total beam-on-time

$$\sum_{k=1}^{N} \alpha_k = x(D', D),$$

Boland-Hamacher-Lenzen established the following important result.

Theorem 6.2: The "Minimize beam-on-time" problem of the MLC is equivalent to the network flow problem

$$\min \ x(D', D) \tag{6.17}$$

subject to x a circulation in $\mathcal{N}_{HB}$ lying between the lower and upper capacity limits $\underline{u}_e$ and $\bar{u}_e$ defined by (6.13) and (6.14), and satisfying the side constraint

$$x((i, l, r)^1, (i, l)\ ) = x(\ (i, r-1), (i, l, r)^2\ ) \qquad \text{for all } (i, l, r) \in V_s. \tag{6.18}$$

Moreover, if the intensity matrix $I \in \mathbb{Z}^{m \times n}$, and thus the capacity of arcs in $\mathcal{N}_{HB}$ is integer, this side constrained network flow problem has an integer optimal solution, which can be found in time polynomial in m and n. In particular, the beam-on-time problem of the MLC can be solved in $O(m^3 n^{10} log(mn^2))$ time.

Proof: See Theorem 4.3 in [16]. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \square$
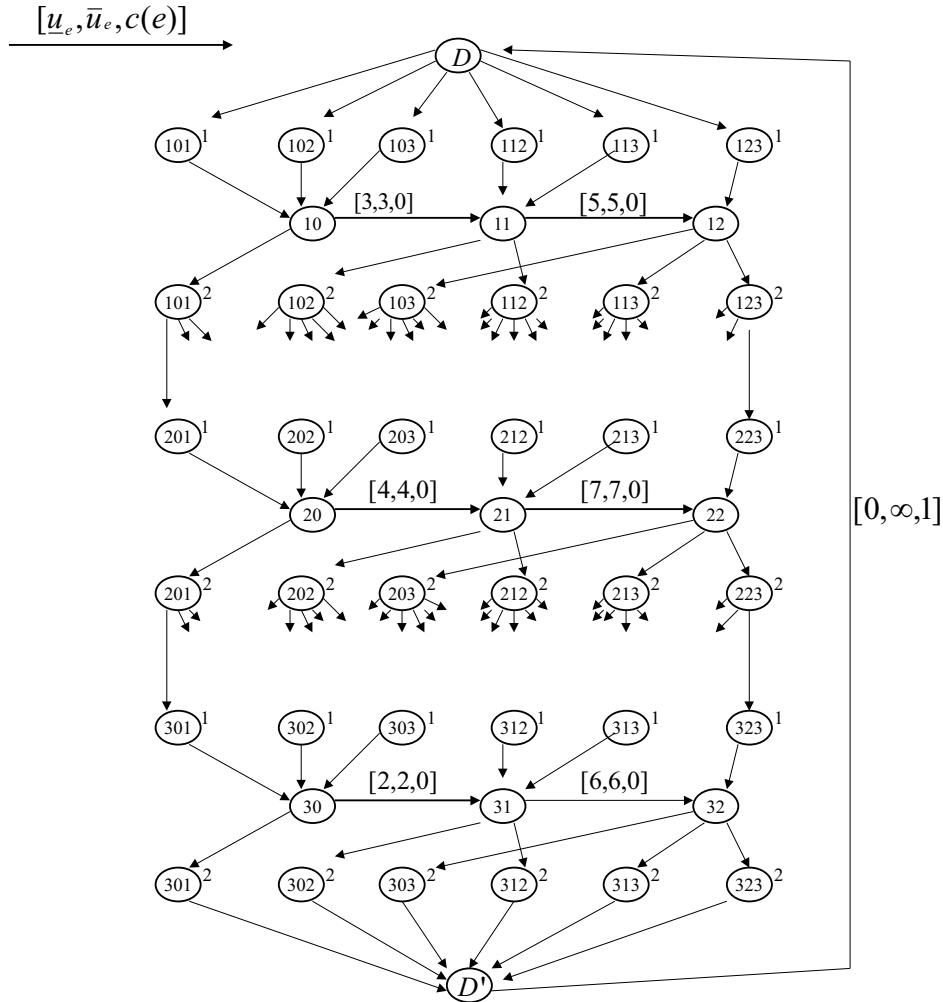
Fig. 6.7: The Hamacher-Boland network corresponding to the intensity matrix of Example 6.2, with all entering arcs, intensity arcs, out-going arcs, and some other arcs (the remaining arcs are indicated by the small arrows). The capacities and cost value $[\underline{u}_e, \bar{u}_e, c(e)]$ of all intensity arcs and the return arc are indicated in the network. For all the other arcs $[\underline{u}_e, \bar{u}_e, c(e)] = [0, \infty, 0]$.

Despite the polynomial solvability of the beam-on-time problem, still it requires a large amount of computational time in order to find an optimal solution if, in particular, the problem size is large. Note that even for a small sized 10×10 intensity matrix, the solving time is of $O(10^{13})$ and practical problems can be a lot more than this size.

One way to deal with this problem is the subgradient optimization method in which the subproblem is constructed via Lagrangian relaxation where the side constraints (6.18) are removed and included into the objective function. In this case, given a Lagrangian multiplier $u \in \mathbb{R}^{|V_s|}$, where $|V_s|$ is the cardinality of the set V_s, the Lagrangian subproblem is given as

$$\min \ x(D', D) + \sum_{(i,l,r) \in V_s} u_{ilr}[x((i,l,r)^1, (i,l)\ ) - x(\ (i,r-1), (i,l,r)^2\ )] \qquad (6.19)$$

$$\text{subject to} \quad x \ \ is \ \ a \ \ circulation \ \ in \ \ \mathcal{N}_{HB}.$$

The subproblem (6.19) is a pure minimum cost network flow (circulation) problem and its solution provides us a step direction (subgradient vector) at each point $u = (u_{ilr})$ for a subgradient procedure. In particular, given $u = (u_{ilr}) \in \mathbb{R}^{|V_s|}$ and a calculated optimal circulation x^* of the subproblem (6.19), the vector $s = (s_{ilr}) \in \mathbb{R}^{|V_s|}$ where

$$s_{ilr} = x^*((i,l,r)^1, (i,l)\ ) - x^*(\ (i,r-1), (i,l,r)^2), \qquad (6.20)$$

is a subgradient of the Lagrangian dual function at $u = (u_{ilr})$. Note that a duality gap does not occur for this problem since the associated side-constrained network flow problem has an integer optimal solution (Theorem 6.2). This is a helpful property in the computational procedure of a subgradient optimization methods. In Section 6.5, we use this method to determine the minimum number of segments of a binary intensity matrices for large instances of a MLC problem.

6.4 Minimizing Total Delivery Time for Radiation Therapy

In this section we deal with the problem of minimizing delivery time of a radiation therapy by a MLC. We construct a new algorithm which will use a combination of heuristic and exact methods in order to find a smaller total delivery time for an intensity matrix $I \in \mathbb{Z}^{m \times n}$ as much as possible. In particular, we use a heuristic level reduction technique similar to that of Xia and Verhey to extract a binary matrix from a given intensity matrix. However, unlike the algorithm of Xia and Verhey, a delivery coefficient of an extracted binary intensity matrix will not be fixed to powers of 2 but it is a calculated value based on certain rules which help to keep the total beam-on-time as small as possible. Further, we use the Hamacher-Boland network model to determine the exact minimum number of segments of an extracted binary intensity matrix.

6.4.1 Max-Level Halving Algorithm

Here we construct an algorithm which extracts binary intensity matrix from a given intensity matrix iteratively. The extracted intensity matrix will not necessarily be a shape matrix. But its exact minimum number of segments can be found as a consequence of the following theorem.

Theorem 6.3: Suppose $M \in \{0,1\}^{m \times n}$ is a binary intensity matrix. The MLC problem "Minimizing the number of segments of M" is equivalent to the problem "Minimizing the total beam-on-time of M".

Proof: Consider a decomposition of a binary matrix M into segments $S^1, S^2, \ldots, S^N$; i.e., $M = \sum_{k=1}^{N} \alpha_k S^k$, where α_k is a positive integer for each k. Then, obviously, $\alpha_k = 1$ for all k. Therefore,

$$\sum_{k=1}^{N} \alpha_k = N, \quad \text{the number of segments of M.}$$

Hence, a minimizer of the total beam-on-time $\sum_{k=1}^{N} \alpha_k$ minimizes also the number of segments of M and vice versa. □

As a consequence of Theorem 6.3, a binary intensity matrix can be decomposed into its exact minimum number of segments using a model whose objective is to minimize beam-on time, for instance, the network flow model [16] due to Boland, Hamacher, and Lenzen or the LP model [3] due to Baatar and Hamacher. We would like to determine a corresponding delivery coefficient α_k, beam-on time of the extracted binary intensity matrix, so as the extraction process reduces the intensity matrix iteratively into zero-matrix with a few number of iterations as much as possible in order to obtain a smaller possible number of segments while also trying to keep each of the α_k's to a minimum possible values in order to produce a reasonable amount of total beam-on time. Such result can be obtained by choosing α_k in such a way that it is close to half of the maximum level of the matrix element. Such delivery coefficients can be constructed based on the difference between maximum intensity levels of consecutive residual intensity matrices, where a *k-th* residual intensity matrix is defined as follows.

Definition 6.4: Let I be an intensity matrix and $M^1, M^2, \ldots, M^{k-1}$ be binary matrices with a corresponding delivery coefficients $\alpha_1, \alpha_2, \ldots, \alpha_{k-1}$. The *k-th residual intensity matrix* I^k after the $k-1$ matrices are extracted is given by

$$I^k = I - \sum_{t=1}^{k-1} \alpha_t M^t. \tag{6.21}$$

Note that the residual intensity matrices can be expressed recursively as

$$I^1 = I, \quad I^{k+1} = I^k - \alpha_k M^k, \quad k = 1, 2, \ldots \tag{6.22}$$

Next, we introduce a rule for determining the set of delivery coefficients. The rule is based on the intension to reduce the maximum intensity level of a residual intensity matrix by at least its half at each iteration so that the maximum intensity level of a *k-th* residual intensity matrix I^k is not more than half of the maximum intensity level of the previous residual intensity matrix I^{k-1}. Thus we choose α_k to be close to half of the maximum level of I^k as described below.

Given an intensity matrix $I \in \mathbb{Z}^{m \times n}$, suppose I^k is its *k-th* residual matrix. Let

- $H_k = \max_{i,j}\{I^k_{ij}\}$ be the highest intensity level of I^k.

- $d_k = 2^m$, where $m = \text{nint}(\log_2 H_k)$, where nint denotes the nearest integer.

Clearly, $\frac{1}{2}d_k < H_k$ since $m - 1 < \log_2 H_k$. Then, the binary matrix $M^k = (M^k_{ij}) \in \{0,1\}^{m \times n}$ can be extracted as:

$$M^k_{ij} = \begin{cases} 1, & \text{if } I^k_{ij} \geq \frac{1}{2}d_k \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, let
- $\bar{H}_k = \max_{i,j}\{I^k_{ij} : I^k_{ij} < \frac{1}{2}d_k\}$.

- $L_k = \min_{i,j}\{I^k_{ij} : I^k_{ij} \geq \frac{1}{2}d_k\}$.

Then, the delivery coefficient of M^k is taken to be

$$\alpha_k = \min\{ H_k - \bar{H}_k, \ L_k \}. \tag{6.23}$$

and the procedure is repeated until $I^k = 0$. Hence the *max-level halving* algorithm can be formulated as follows:

Algorithm 6.1: **Max-Level Halving Algorithm**
- Input: Intensity matrix $I \in \mathbb{Z}^{m \times n}$;
- Initialization: let $k = 1$, $I^1 = I$;
- **Repeat**

  Step 1: let $H_k = \max_{i,j}\{I^k_{ij}\}$;

  let $d_k = 2^{m_k}$, where $m_k = \text{nint}(\log_2 H_k)$;

  let $\bar{H}_k = \max_{i,j}\{I^k_{ij} : I^k_{ij} < \frac{1}{2}d_k\}$;

  let $L_k = \min_{i,j}\{I^k_{ij} : I^k_{ij} \geq \frac{1}{2}d_k\}$;

  Step 2: Let $M^k = M^k_{ij} \in \{0,1\}^{m \times n}$, where

  $$M^k_{ij} = \begin{cases} 1, & \text{if } I^k_{ij} \geq \frac{1}{2}d_k \\ 0, & \text{otherwise.} \end{cases}$$

  let $\alpha_k = \min\{ H_k - \bar{H}_k, \ L_k \}$;

  let $I^{k+1} = I^k - \alpha_k M^k$;

Step 3: Find a decomposition of M^k with exact minimum number of
segments;

let $k = k + 1$;

Until $\max_{i,j}\{I_{ij}^k\} = 0$

The output of the algorithm is the decomposition

$$I = \sum_k \alpha_k \left(\sum_{t=1}^{T_k^*} S^{kt} \right), \tag{6.24}$$

where $S^{k1}, S^{k2}, \ldots, S^{k,T_k^*}$ are minimal segments of the binary matrix M^k, for each k, obtained by an exact method. We use the Hamacher-Boland network model to determine the minimum number of segments of a binary intensity matrix.

Remarks:- From the construction of the algorithm, Algorithm 6.1, one can see by a direct calculation that

1. $\alpha_k > 0$, for each delivery coefficient α_k.

2. $I_{ij}^k \geq 0$, for each residual matrix $I^k = (I_{ij}^k)$.

3. for every consecutive residual matrices $I^k = (I_{ij}^k)$ and $I^{k+1} = (I_{ij}^{k+1})$, it holds that

$$\max_{i,j}\{I_{ij}^{k+1}\} = \max_{i,j}\{I_{ij}^k\} - \alpha_k. \tag{6.25}$$

The property at equation 6.25 can help to reduce the total beam-on-time as compared to that obtained by the Xia and Verhey algorithm; and in some cases the Max-level halving algorithm can produce a decomposition with a minimum total beam-on-time (See, for instance, Theorem 6.5 and Corollary 6.6).

Example 6.3: Suppose

$$I = \begin{pmatrix} 1 & 0 \\ 10 & 3 \\ 28 & 27 \\ 26 & 21 \\ 3 & 2 \end{pmatrix}.$$

Application of Algorithm 6.1 gives us 4 segments; namely,

$$S^1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 0 \end{pmatrix}, \; S^2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}, \; S^3 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}, \; \text{and} \; S^4 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

with the corresponding delivery coefficients $\alpha_1 = 18, \quad \alpha_2 = 7, \quad \alpha_3 = 2, \quad \alpha_4 = 1$. Therefore, the total beam-on-time of the decomposition is $\sum_{k=1}^{4} \alpha_k = 28$.

The next theorem provides a lower bound to a total beam-on-time of a decomposition of an intensity matrix whose proof is trivial.

Theorem 6.4: Suppose $I = (I_{ij})$ is an intensity matrix and $\{S^1, S^2, \ldots, S^N\}$ is a set of its segments with corresponding delivery coefficients $\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$. Then

$$\sum_{k=1}^{N} \alpha_k \geq \max_{i,j}\{I_{ij}\}.$$

$\square$

Theorem 6.5: If each of the binary matrix M^k constructed by the max-level halving algorithm (Algorithm 6.1) is a segment (shape matrix), then the algorithm provides a decomposition with a minimum total beam-on-time.

Proof: Let $M^1, M^2, \ldots, M^N$ be the N binary matrices constructed by Algorithm 6.1 and suppose each of them is a segment (shape matrix). The algorithm constructs a set of residual matrices $\{I^1, I^2, \ldots, I^N\}$ where $I^1 = I$ and $I^{N+1} = 0$ such that

$$I^{k+1} = I^k - \alpha_k M^k, \quad k = 1, 2, \ldots, N.$$

Moreover, from (6.25), we have

$$\alpha_k = \max_{i,j}\{I_{ij}^k\} - \max_{i,j}\{I_{ij}^{k+1}\}$$

for each $k = 1, 2, \ldots, N$ where $\max_{i,j}\{I_{ij}^{N+1}\} = 0$. Therefore,

$$
\begin{aligned}
\sum_{k=1}^{N} \alpha_k &= \sum_{k=1}^{N} \left[\max_{i,j}\{I_{ij}^k\} - \max_{i,j}\{I_{ij}^{k+1}\} \right] \\
&= \max_{i,j}\{I_{ij}^1\} - \max_{i,j}\{I_{ij}^{N+1}\} \\
&= \max_{i,j}\{I_{ij}\}.
\end{aligned}
$$

Thus, by Theorem 6.4, $\sum_{k=1}^{N} \alpha_k$ is equal to its lower bound; which means the minimum value has been reached. $\qquad\square$

Corollary 6.6: If I is an $m \times 2$ intensity matrix, then the max-level halving algorithm, Algorithm 6.1, constructs its decomposition with the minimum total beam-on-time.

Proof: Any two column binary matrix M^k constructed by the algorithm is a shape matrix (segment). Hence, the claim of the theorem follows directly from Theorem 6.5. $\qquad\square$

One can observe that Theorem 6.5 and the above corollary provide a rationale for choosing the rule of α_k as given in the Algorithm 6.1.

6.4.2   Worst-Case Bound on the Number of Segments

Next we determine a worst-case bound on the total number of segments if the max-level halving algorithm is used. Note that from the definition of H_k, L_k, $\bar{H}_k$, and α_k we have the following relations.

$$ H_{k+1} = \max\{\bar{H}_k, H_k - \alpha_k\}, \tag{6.26} $$

and from equation (6.25) it follows that

$$ H_{k+1} = H_k - \alpha_k. \tag{6.27} $$

Hence (6.26) and (6.27) implies that

$$H_{k+1} \geq \bar{H}_k. \tag{6.28}$$

Now, let $m = \text{nint}(\log_2 H_k)$.

Then, H_k is "close to" 2^m. In fact,

$$2^{m-1} < H_k \leq 2^m \tag{6.29}$$

or,

$$2^m \leq H_k < 2^{m+1}. \tag{6.30}$$

To derive the upper bound to the total number of segments, let us first consider a 2-column intensity matrix. From Corollary 6.6, one can conclude that the number of segments equals to the number of elements in the sequence

$$\mathcal{H} = \{H_1, H_2, \ldots, H_t\},$$

where $0 < H_k = \max_{i,j}\{I_{ij}^k\}$, the highest intensity level of *k-th* nonzero residual intensity matrix I^k. Thus, we determine the maximum number of elements in the finite sequence $\mathcal{H}$.

Case 1: Suppose $H_k \leq 2^m$ where $m = \text{nint}(\log_2 H_k) > 0$. (See 6.29).

Claim: $H_{k+1} \leq 2^{m-1}$.

To show this, note that from the definition of α_k, $\bar{H}_k$ and L_k, either

$$\alpha_k = H_k - \bar{H}_k \quad \text{or} \quad \alpha_k = L_k,$$

where $\bar{H}_k \leq 2^{m-1}$ and $2^{m-1} \leq L_k \leq H_k$.

If $\alpha_k = H_k - \bar{H}_k$, then from (6.27) we obtain directly

$$H_{k+1} = H_k - \alpha_k = \bar{H}_k \leq 2^{m-1}. \tag{6.31}$$

If $\alpha_k = L_k$, then

$$
\begin{aligned}
H_{k+1} =\ & H_k - \alpha_k \\
\leq\ & H_k - 2^{m-1} && (\text{since } \alpha_k = L_k \geq 2^{m-1}.) \\
\leq\ & 2^m - 2^{m-1} && (\text{since } H_k \leq 2^m). \\
=\ & 2^{m-1}.
\end{aligned}
$$

This relation together with (6.31) imply that

$$H_{k+1} \leq 2^{m-1} \tag{6.32}$$

as claimed. From this we conclude that the sequence $\mathcal{H}$ can have at most m_1 nonzero elements, where $m_1 = \mathrm{nint}(\log_2(\max_{i,j} I_{ij}))$.

Case 2: If H_k has the property (6.30), i.e., $2^m < H^k < 2^{m+1}$, then replacing m by $m + 1$ in the above reasoning (Case 1), we conclude that $\mathcal{H}$ can have at most $m_1 + 1$ elements, where m_1 is defined as above.

Hence for any two-column intensity matrix I with the highest intensity level $H_1 = \max_{i,j}\{I_{ij}\}$, the max-level halving algorithm provides in the worst-case a maximum of

$$N_{max} = \mathrm{nint}(\log_2 H_1) + 1 \tag{6.33}$$

segments. Comparing this result with that of Xia and Verhey algorithm, one can observe that the number of segments obtained by the max-level halving algorithm equals in the worst case to that obtained by the Xia and Verhey algorithm since the number of segments of a 2-column intensity matrix is always either $\mathrm{nint}(\log_2 H_1)$ or $\mathrm{nint}(\log_2 H_1) + 1$ if the Xia and Verhey algorithm is used. Indeed, the max-level halving algorithm can produce smaller number of segments. For instance, suppose $I = \begin{pmatrix} 1 & 7 \\ 0 & 6 \end{pmatrix}$. If max-level halving algorithm is used, then $H_1 = 7$, $L_1 = 6$, $\bar{H}_1 = 1$, and $m_1 = 3$. Consequently, $\alpha_1 = \min\{H_1 - \bar{H}_1, L_1\} = 6$ and the corresponding segment $S^1 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$.

Thus, the residual matrix $I^2 = I - \alpha_1 S^1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$, which implies, $\alpha_2 = 1$ and $S^2 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$. Therefore, $I = \alpha_1 S^1 + \alpha_2 S^2$, i.e., I is decomposed into two segments. However, if Xia and Verhey algorithm is applied, then it produces three

segments with sequence of delivery coefficients $\{4, 2, 1\}$.

For general $m \times n$ intensity matrix, where $n > 2$, in the worst-case one can split the matrix in to several two column sub-matrices and apply the method described above to the sub-matrices. Hence, the maximum number of segments, N_{max}, for such an n column intensity matrix can be,

$$N_{max} = \lfloor \frac{n+1}{2} \rfloor (m_1 + 1) \tag{6.34}$$

where $m_1 = \text{nint}(\log_2(\max_{i,j} I_{ij}))$ and $\lfloor . \rfloor$ is the greatest integer less or equal to the number.

6.5   Implementation

In this section we address the issues regarding implementation of the max-level halving algorithm and compare its result with that of others. Note that the crucial point in the implementation of the algorithm is to determine a decomposition of a binary matrix with exact minimum number of segments. This can be accomplished by using the Hamacher-Boland network flow model of a MLC problem since, by Theorem 6.3, minimizing the number of segments of a binary intensity matrix is equivalent to minimizing its total beam-on-time. Recall that, beside the large size of the problem, the model is a side-constrained network flow problem. In the cases of large instance, we use the subgradient optimization methods in order to solve the problem by "removing" the side constraints as described by equations 6.19 and 6.20. That is, for any $u = (u_{ilr}) \in \mathbb{R}^{|V_s|}$, we need to solve the Lagrangian subproblem SP(u):

min $x(D', D) + \sum_{(i,l,r) \in V_s} u_{ilr}[x((i,l,r)^1, (i,l)) - x((i, r-1), (i,l,r)^2)]$
s.t. x is a circulation in $\mathcal{N}_{HB}$.

$$\tag{6.35}$$

We solve this problem using a negative cycle cancelling procedure. The negative cycle cancelling algorithm starts with a feasible circulation x in the network.

At every iteration, the algorithm constructs a corresponding residual network $G(x)$ and finds a negative cost cycle W in $G(x)$. If such cycle is identified, then the algorithm augments the maximum possible flow along W, updates $G(x)$ and repeats the process. If $G(x)$ contains no negative cycle, then x is a minimum cost circulation, so the algorithm terminates. For details, one can refer to the standard text books by Ahuja et al. [2], Hamacher and Klamroth [42], and Murty [63]. This algorithm maintains feasibility of the solution of a circulation problem at every step. Moreover, a minimum cost circulation problem has integrality property; namely, if all arcs capacities are all integer, the minimum cost circulation problem always has an integer solution. Therefore, the algorithm we implement mainly incorporates the following subroutines.

Given a binary intensity matrix M and HB-network :

1. Find starting feasible circulatiuon in HB-network with $u = 0$.

   Observe that we need to find the set of paths from the source node D to the sink node D' each of which carries flow of 1 unit and covers all nonzero intensity arcs. An intensity arc should occur on one and only one of the paths. So the problem is reduced to find the set of all such covering paths where each nonzero intensity arc is covered by one and only one path. We accomplish this by a set of longest paths from D to D' by defining the following arc length l_1: for any arc $e \in \mathcal{N}_{HB}$ let

$$
l_1(e) = \begin{cases} 1, & \text{if } e \text{ is a nonzero intensity arc} \\ 0, & \text{otherwise.} \end{cases} \tag{6.36}
$$

   Thus, we have the following.

   - Subroutine 1: Feasible circulation in $\mathcal{N}_{HB}$, $(u = 0)$
     Repeat:

     – Determine the longest path from D to D'.

     – If the length of this path is not zero, then augment a flow of 1 unit along the path.

     – Delete all intensity arcs included in the path from the network.

Until: The network contains no nonzero intensity arc.

The rationale for the longest path is to cover as many intensity arcs as possible by the path. This provides us with a "good" starting feasible circulation.

2. Negative cost cycles in the residual network, $u = 0$:
   Once we have a feasible circulation x, we construct a corresponding residual network $G(x)$ as usual and search for negative cycle(s) in $G(x)$. Observe that any path from node D' to node D in $G(x)$ together with the reverse of the return arc, i.e., (D, D'), gives us a negative cost cycle and non of such path consisting an intensity arc. Therefore, the problem of finding cycles of negative cost in $G(x)$ is reduced to finding the set of paths from the node D' to the node D in $G(x)$. This can be accomplished by determining the set of shortest paths from the node D' to node D in $G(x)$. In this case, however, the arc length is defined as 1 unit for each arc in $G(x)$. That is, for each arc e in $G(x)$, let its arc length to be $l_2(e) = 1$. Then,

   - Subroutine 2: Negative cost cycles in $G(x)$:
     Repeat:

       - Determine the shortest path from D' to D.

       - Augment a flow of 1 unit along the path.

       - Update $G(x)$.

     Until: No more path from D' to D in $G(x)$.

   At termination of subroutine 2, we obtain a solution, say x_0^*, of the subproblem SP(0). Given u, if x_u^* is an optimal circulation of the subproblem SP(u) and the subgradient of the dual function at u is a zero vector then x_u^* is an optimal solution for the constrained network flow problem. Otherwise, if there is a nonzero component of the subgradient, say

$$s_{ilr} = x_u^*(i, l, r, 2, i, r - 1) - x_u^*(i, l, r, 1, i, l) > 0,$$

then u is updated in the subgradient procedure and solve the corresponding subproblem SP(u). This can be solved also by the subroutine 2 with a modification that at each iteration we set $D' := (i, r-1)$ as a source node and $D := (i, l, r, 2)$ as a sink node in G(x_u^*). A path from $(i, r-1)$ to $D := (i, l, r, 2)$ together with the arc $(i, l, r, 2) \rightarrow (i, r-1)$ is the desired cycle in G(x_u^*). Similarly, if the subgradient has a negative component, say

$$s_{ilr} = x_u^*(i, l, r, 2, i, r-1) - x_u^*(i, l, r, 1, i, l) < 0,$$

then we set in the subroutine 2 the start node to be $D' := (i, l)$ and the sink node to be $D := (i, l, r, 1)$. A path from the node (i, l) to the node $(i, l, r, 1)$ together with the arc $(i, l, r, 1) \rightarrow (i, l)$ is the desired cycle in G(x_u^*).

Other main difficulty regarding implementation is that the practical problems are usually large-sized and the model requires a large number of variables to describe even a medium-sized problem since the network needs a large number of arcs to represent such problems. In particular, the network corresponding to an $m \times n$ intensity matrix involves $m(n+1)(n+3) + 2$ nodes and

$$(m-1)[\frac{1}{2}(n+2)(n+1)]^2 + (m+1)(n+2)(n+1) + m.n + 1$$

arcs, which can be shown by straightforward calculations (see, [40]). That is, the number of nodes and arcs of the Hamacher-Boland network corresponding to an $m \times n$ intensity matrix is of size $O(mn^2)$ and $O(mn^4)$, respectively. One way to overcome this difficulty is to reduce the number of variables in the model via pre-processing techniques based on some special structure of a particular intensity matrix.

6.5.1   Some Preprocessing Techniques

As described above the network model associated with an intensity matrix is usually very large with, particularly, large number of arcs. However, in the case of binary intensity matrices that been constructed by the above algorithms, we usually obtain a sparse matrix. As a consequence, the associated network

involves a large number of arcs with zero flows. The removal of these unnecessary arcs substantially reduces the variables of an instance and therefore the over all computation time. Our algorithm incorporates the following techniques to reduce the size of an instance in the corresponding network.

1. **Removal of nodes**: Consider a representation (i, l, r) such that $l < r - 1$. If a radiation "enter" into this node, then every site (entry) (i, j) where $l < j < r$ would be irradiated. Thus, one can observe that for any j such that $l < j < r$ if the intensity $I_{ij} = 0$, then there should be no radiation going into the node (i, l, r). Hence such nodes can be removed. That is, all of the nodes in the set

$$Zeronodes = \{(i, l, r)^1, (i, l, r)^2 : (i, l, r) \in V_s \text{ and}$$
$$\text{exists } j \text{ such that } l < j < r \text{ and } I_{ij} = 0 \} \quad (6.37)$$

   are removed form the network.

2. **Removal of arcs** : Obviously, any arc incident with a node belonging to the set $Zeronodes$ can carry only zero flow and is removed from the model.

3. **Removal of first and last nodes:** One can observe that the first node $(i, 0, 1)$ and the last node $(i, n, n + 1)$ in each layer i have no significant role and their removal does no affect a final solution. Therefore, such nodes and all arcs incident with any of these nodes are all removed.

4. **Removal of unnecessary intensity nodes and arcs**: Suppose (i, j) and $(i, j + 1)$ are two consecutive intensity nodes such that $I_{ij} = 0 = I_{j,j+1}$. Then no positive flow can come into the node (i, j). Therefore, the set of all nodes

$$ZI = \{(i, j) : (i, j) \text{ is intensity node and } I_{ij} = 0 = I_{i,j+1}\}$$

   is removed. We remove also all arcs incident to a node in ZI.

5. **Redirecting flows from a closed node**: By closed node we mean a node belonging to a set

$$CN = \{(i, l, r)^1, (i, l, r)^2 : (i, l, r) \in V_s \text{ and } r = l + 1\}.$$

Such nodes represent position of a pair of leaves where the channel is closed, i.e, no cite in the row receive radiation. Therefore, a flow entering a channel i through a node $(i, l, r)^1$ can move directly to $(i, l, r)^2$ without going through any intensity node in the channel. Therefore, for every row i, each of the set of arcs

$$\{\big((i, l, r)^1, (i, l)\big), \big((i, r - 1), (i, l, r)^2\big) :$$
$$\big((i, l, r)^1, (i, l)\big), \big((i, r - 1), (i, l, r)^2\big) \in \mathcal{A} \text{ and } r = l + 1\}$$

is removed from the model and replaced by the arc $((i, l, r)^1, (i, l, r)^2)$.

6. **Fixing variables:** For any intensity arc $((i, j - 1), (i, j)) \in \mathcal{A}$, the lower and upper capacities are both equal to I_{ij}. Consequently, the value of a flow on such arc is equal to I_{ij} in the final solution; and, hence, can be fixed from the beginning. Therefore, for any intensity arc $((i, j - 1), (i, j)) \in \mathcal{A}$,

$$x((i, j - 1), (i, j)) = I_{ij}$$

is fixed, where x is a flow in the network.

Example 6.4: Consider representing a binary intensity matrix

$$M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

by the Hamacher-Boland network. In general, the network requires 142 nodes and 842 arcs to represent a 4×4 intensity matrix. However, after removing unnecessary nodes and arcs by using the above techniques, we can represent the matrix by a smaller sized network which involves only 55 nodes and 80 arcs (see, Fig. 6.8).
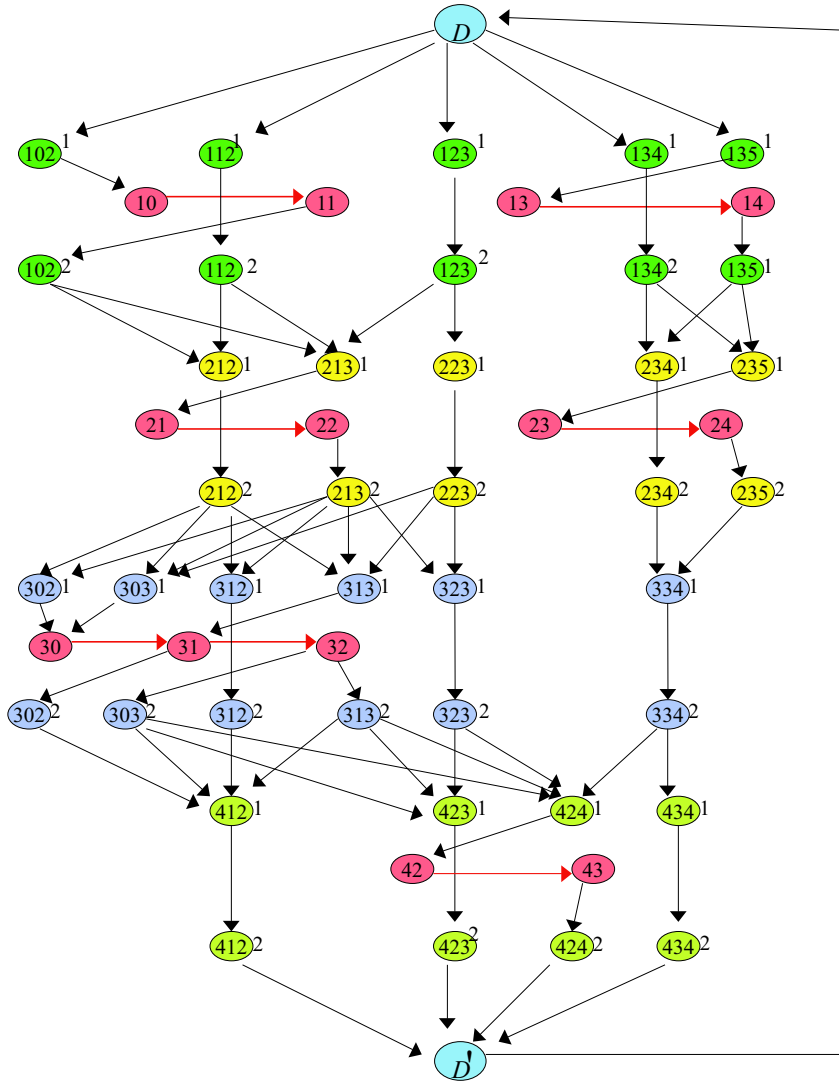
Fig. 6.8: Hamacher-Boland network representation of the matrix M of Example 6.4 after the removal of unnecessary nodes and arcs .

In this section, we present some numerical results of our computational experiments. Here, conversion of units of measurement is needed since the units of delivery time is seconds while MLC measures the intensity of radiation with monitor units (MU). Therefore, we determine the time (in sec) required by a MLC to delivery a unit of intensity level based on the report of Siochi [81]. This is done by setting the number of MU per intensity level to be 2. Moreover, it is given that the linear accelerator can deliver 200 MU per minute; i.e, delivering a single MU takes 0.3 sec. Hence, we obtain $2 \times 0.3 = 0.6$ sec/(intensity level). i.e.,

$$\text{Delivery time} = (0.6 \times \text{BOT}) + (N - 1)\tau$$

where BOT is beam-on time, N is number of segments and τ is the set-up time in seconds.

Now, we compare first the results obtained by our new algorithm (MLH) with that of four other existing algorithms; namely, the network flow (NF) procedure due to Boland , Hamacher and Lenzen [16], the Xia and Verhey (XV) algorithm [85], the agorithm of Siochi [81], and the very recent algorithm by Baatar and Hamacher (BH) [3]. Figure 6.9 shows the comparison with set-up time 18 seconds tested on 15 randomly generated 10×10 intensity matrices with 15 intensity levels, ranging from 1 to 15. Here, we applied the numerical test on the same test set used in [59], [16] and [3] in order to facilitate the comparison.

In this case, our new algorithm (MLH) and the recent algorithm by Baatar and Hamacher (BH) yield always the smallest delivery time alternatively as compared to the other three. In fact the average total delivery time (in sec) obtained by MLH, BH, NF, XV, and Siochi is 334, 337, 388, 432, and 499, respectively.
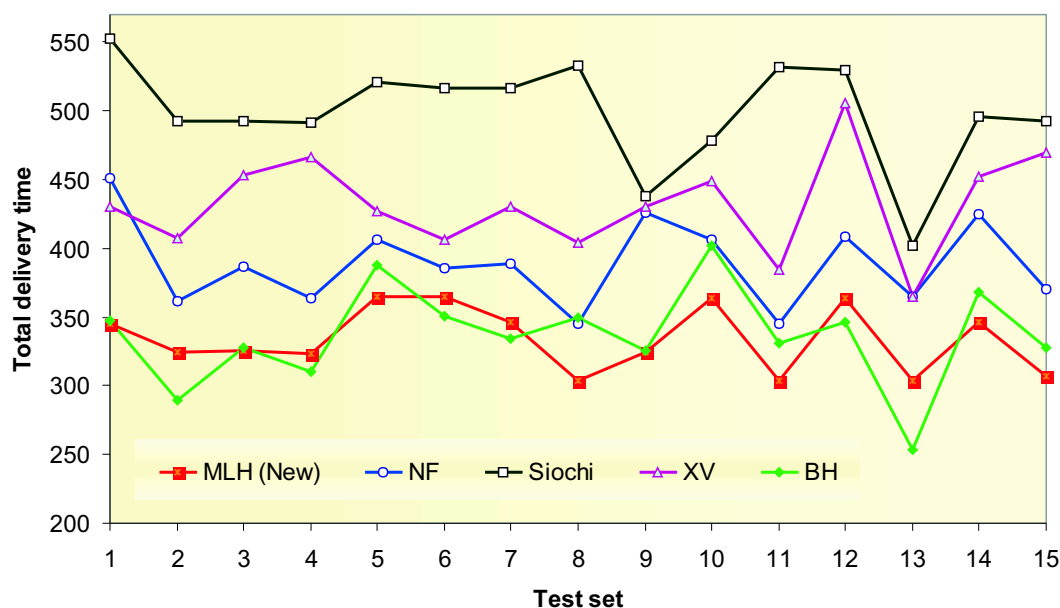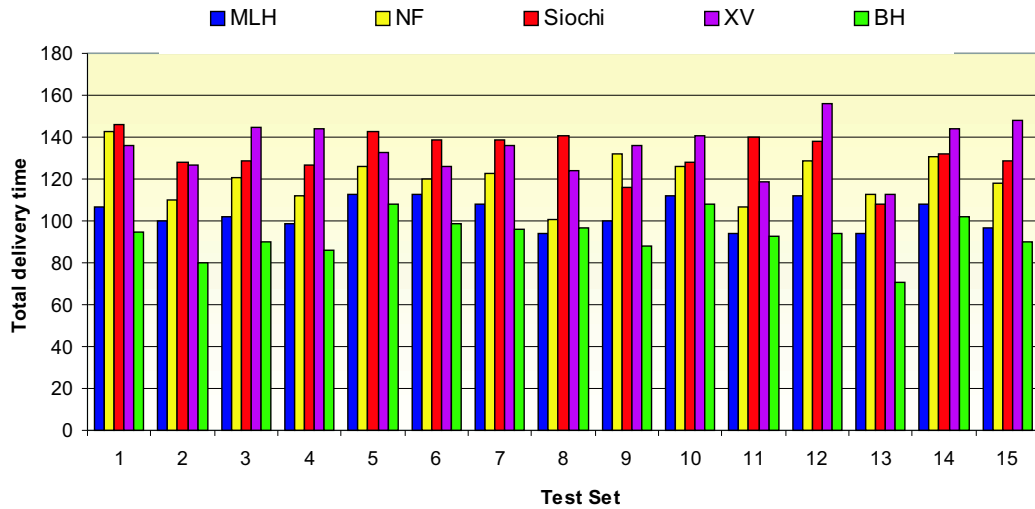
Fig. 6.9: Comparison of the results(total delivery time in sec) obtained by the five procedures applied on 15 different 10×10 intensity matrices with 15 maximum intensity level and constant set up time equals 18 sec.

If st-up time is very small, then BH yields the smallest total delivery time as can be seen from Figure 6.10 for the case with the set-up time equals 4 sec. The reason for this is that BH always finds optimal beam-on time. However, if the comparison with respect to the number of segments is considered then the number of segments obtained by our new algorithm (MLH) is almost always smaller than the number of segments obtained by BH and the others. This is shown in Figure 6.11.

Fig. 6.10: Comparison of the results(total delivery time in sec) obtained by the five procedures applied on 15 different 10×10 intensity matrices with 15 maximum intensity level and constant set up time equals 4 sec.
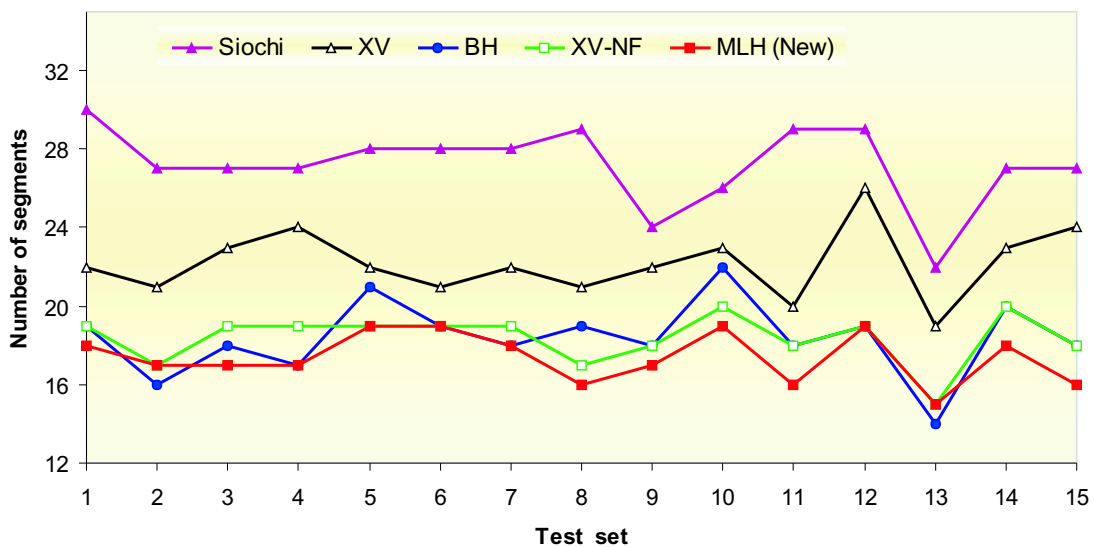
Fig. 6.11: Comparison : total number of segments obtained by the five procedures applied on 15 different 10×10 intensity matrices with 15 maximum intensity level.

Our solution procedure also enabled us to solve large-sized practical problems within a reasonable time. To check this, we compare the performance of MLH with NF. The reason for restriction of the comparison to only these two is that, only MLH and NF are using the same model having the same variables, i.e., the Hamacher-Boland network flow model. Thus, in Table 6.1, we compare the results obtained by MLH and NF applied on 40 clinical cases (practical data)[1] with 10 maximum intensity level. Some of these cases are of large size. The NF unable to deal with the large instances. In fact, for some of the large-sized instances, NF using CPLEX could not produce a result within three hours and for some others, the problem size exceeded the available memory. However, our new algorithm (MLH) together with the subgradient solution method solved these large-sized instances within at most 95 seconds and without any complain about the memory space.

---

[1] Dr. Alfredo Siochi is gratefully acknowledged for providing us the data.

	Number of segments		Beam-on time		Delivery time		Computation time (sec)	
Size	MLH	NF	MLH	NF	MLH	NF	MLH	NF
14×10	14	23	33	23	72	102	22	46
10×9	13	18	31	18	66	79	6	9
10×19	13	18	30	18	66	79	8	12
9×11	14	19	34	19	72	83	10	140
14×10	16	21	36	22	82	93	12	13
15×11	16	21	39	22	83	93	12	15
15×11	16	23	38	23	83	102	16	220
14×10	16	23	38	23	83	102	17	290
14×10	17	24	42	24	89	106	15	110
15×11	17	23	42	23	89	102	16	170
14×10	18	30	46	30	96	134	17	46
15×11	18	29	46	30	96	130	16	77
15×10	19	22	47	22	100	97	18	42
15×11	19	22	47	22	100	97	18	43
15×11	14	23	33	23	72	102	22	46
10×10	12	16	29	16	61	70	4	15
11×9	13	18	31	18	67	79	5	17
11×11	12	15	29	16	61	66	6	22
9×9	14	20	31	20	71	88	4	6
10×19	15	16	35	16	77	70	4	4
11×9	15	16	35	16	71	70	5	5
11×9	13	18	30	18	66	79	7	10
9×10	14	19	34	19	72	83	7	5
9×9	13	15	31	17	67	66	4	3
29×41	19	**	34	**	92	**	36	**
20×25	19	*	34	*	92	*	32	*
16×28	16	*	33	*	80	*	28	*
20×23	10	*	20	*	48	*	40	*
16×28	18	*	35	*	89	*	50	*
15×28	21	*	36	*	102	*	90	*
16×30	19	*	40	*	96	*	80	*
16×27	17	*	35	*	85	*	38	*
16×29	19	*	39	*	95	*	47	*
29×41	16	**	33	**	80	**	35	**
29×41	10	**	20	**	48	**	32	**
29×41	18	**	35	**	89	**	55	**
29×41	21	**	36	**	102	**	95	**
29×41	19	**	40	**	96	**	92	**
29×41	17	**	35	**	85	**	43	**
29×41	18	**	37	**	90	**	52	**

Tab. 6.1: Comparison of results produced by the MLH and the NF applied on 40 clinical cases (practical data), with 10 max intensity level and set-up time 4 sec.

* In this case, NF using the CPLEX could not solve the problem within 3 hours.

** In this case, the problem size exceeded the available memory.

7. CONCLUSIONS AND FUTURE RESEARCH

In this thesis, different versions of subgradient methods have been investigated and a unified presentation has been provided. Furthermore, a new strategy which can completely eliminate the main drawback, namely, the zigzagging phenomena, of the subgradient methods has been established. We have also employed the subgradient methods to solve effectively a large-scale practical problem in the radiation therapy planning.

The foundation of our work is the methodology of Lagrangian relaxation and dualization of an integer programming problem. The methods help us to remove a set of complicating constraints of an integer programming problem and give us a frame work to exploit any available "nice" mathematical structure embedded in the problem in order to solve the problem, perhaps, approximately. However, this is not without a cost. In fact, this transforms the problem into a problem of nonsmooth optimization which has to be solved iteratively. These procedures as well as the consequences have been discussed in Chapter 2.

In Chapter 3, different versions of subgradient methods was investigated and a unified presentation has been provided. We have also investigated the zigzagging phenomena of the subgradient methods in detail and formalized the definition of zigzagging in terms of mathematical expression. The formalization served to distinguish two different kinds of zigzagging and aided us to identify corresponding subgradient procedures that are better suited to control respective kind of zigzagging. Furthermore, the formalized definition of zigzagging has also helped us to determine suitable values of parameters in some of the existing subgradient

procedures in order to strengthen their capability to control zigzagging.

One of the central points of our research was in Chapter 4. There, we established a new procedure which can completely eliminate the zigzagging phenomena. We called the procedure the *hybrid subgradient method* and its step direction the *hybrid subgradient.* Obviously, this tremendously increases the speed of convergence of the subgradient methods. For a general problem, the computation of the hybrid subgradient might demand difficult works since it involves determination of the normal cone at each iterate point and the projection of the subgradient vector onto the cone. However, for the problem of maximization of the Lagrangian dual of integer programming, we prescribed an explicit rule which computes the hybrid subgradient. In this case, the computation does not demand a significant additional work than the computation of a subgradient vector.

Further extensions of the zigzag-free hybrid subgradient method would be useful. For instance, as briefly described in Chapter 3, the bundle method is closely related with the subgradient methods. In fact, the bundle method also uses a step direction based on the subgradient vectors and subjected to zigzagging. An interesting subject for further research would be therefore the employment of the hybrid subgradient method to eliminate the zigzagging behavior of the bundle method.

Chapter 5 examined the issues of directly obtaining the primal solutions within the subgradient schemes while solving the Lagrangian dual formulations. We reviewed classes of procedures which deal with the construction of primal solutions directly from the information generated in the process of deflected or conditional subgradient optimization methods. We deduced that these methods can produce a near-optimal primal solution for an integer programming problem. Note that since these procedures are based on either the deflected or a conditional subgradient methods, they are not completely free of zigzagging. Naturally, therefore, it is

desirable to use the newly constructed zigzag-free hybrid subgradient method for the purpose of generating both the dual and primal solutions. However, though we have proved that the zigzag-free hybrid subgradient scheme can give dual convergence, whether or not this scheme would also yield primal convergence is an open question.

The goal of this thesis has been to employ the subgradient optimization techniques to solve a large-scale optimization problem that originated from a radiation therapy planning problem. In Chapter 6, we have formulated the problem of minimization of total delivery time of a given radiation dose for a cancer patient. We have also described in detail the Hamacher-Boland network flow model which represents important physical constraints of the Multileaf Collimator. The model is used to minimize the total beam-on time to optimality. However, the problem of minimization of the total delivery time, which is the sum of total beam-on time and total set-up times, is NP-hard and thus far there exists no method for solving the problem to optimality. In this thesis, Chapter 6, we have established also a new, fast and efficient algorithm which combines exact and heuristic procedures with the objective to minimize the total delivery time. The exact subproblem in our procedure deals with the problem of minimizing a beam-on time of a binary intensity matrix generated by the algorithm. We used the Hamacher-Boland network flow model to solve this problem. Since the model represents a side-constrained network flow problem and consists of large number of nodes and arcs, its direct applicability has been limited only to problems of small size. Here, we successfully overcame this difficulty by using subgradient methods. In particular, using the Lagrangian dual technique, we dualized all the side constraints and obtained a pure minimum cost network flow (circulation) problem as a subproblem in the subgradient procedure. We solved the resulting circulation problem using the negative cycle cancelling method. In this procedure, we obtained a "good" starting feasible circulation using a set of distinct longest paths in the network which partition the intensity arcs. Once we have a

feasible circulation, we use shortest paths in the residual network to determine cycles with negative cost. The numerical implementations show that our solution method dramatically reduces computational time as well as memory space requirement of large-sized practical problems. Furthermore, we presented numerical tests which show that our new algorithm yields almost always the smallest number of segments without a cost of unreasonable increment of the beam-on time. Consequently, our algorithm is a good alternative for the problem of minimizing the total delivery time of radiation.

One can observe that the problem of minimization of the total delivery time of radiation can be treated as an optimization problem with two objectives; namely, minimization of total beam-on time and minimization of number of segments (shape matrices). A possible future research work is therefore to examine whether or not it would be advantageous to employ multiple objective optimization methods to solve the problem.

BIBLIOGRAPHY

[1] Ahuja, R.K. and Hamacher, H.W.: *Linear time network flow algorithm to minimize beam-on time for unconstrained multileaf collimator problems in cancer radiation therapy.* Working Paper, 2003. Submitted to Networks.

[2] Ahuja, R.K., Magnanti, T.L., and Orlin, J.B.: *Network flows: Theory, algorithms and applications.* Prentice-Hall, 1993.

[3] Baatar, D. and Hamacher, H.W.: *Fast LP models for constrained multileaf collimators in radiation therapy.* Report, Fachbereich Mathematik, Universitaet Kaiserslautern, 2003.

[4] Baatar, D.: *New models for multileaf collimators* Master Thesis, University of Kaiserslautern, 2002.

[5] Bahiense L., Maculan N., and Sagastizábal C. : *The volume algorithm revisited: relation with bundle methods.* Mathematical Programming, Ser. A, Vol. 94, 41 – 69, 2002.

[6] Barahona, F. and Anbil, R.: *The volume algorithm: producing primal solutions with a subgradient method.* Mathematical Programming, 87(3),385-400 , 2000.

[7] Barahona, F. and Anbil, R.: *On some difficult linear programs coming from set partitioning.* Research Report, RC 2140, IBM Watson Research Center, 1999.

[8] Barahona, F. and Chudak, F.: *Solving large scale uncapacitated location problems.* Research Report, RC 21515, IBM Watson Research Center, 1999.

[9] Bazaraa, M.S.: *A survey of various tactics for generating Lagrangian multipliers in the context of Lagrangian duality.* European Journal of Operational Research, 3, 322 – 338, 1979.

[10] Bazaraa, M.S. and Sherali, H.D.: *On the choice of step size in subgradient optimization.* European Journal of Operational Research, 7, 1981, pp. 380 - 388.

[11] Bazaraa, M.S., Jarvis, J.J. and Sherali, H.D.: *Linear Programming and Network Flows.* John Wiley & Sons, Inc., pp. 320 - 369, 1977.

[12] Bazaraa, M.S., Sherali, H.D. and Shetty, C.M.: *Non Linear Programming: Theory and Algorithms.* John Wiley & Sons, Inc, 1993.

[13] Beasley, J.E: *An algorithm for set covering problem.* European Journal of Operational Research, 31, 1987, pp. 85 - 93.

[14] Beasley, J.E.: *An algorithm for solving large capacitated warehouse location problems.* European Journal of Operational Research, 33, 1988, pp. 314 - 325.

[15] Beasrley, J.E.: *Lagrangian relaxation.* Chapter 6 in Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publication, Oxford, 1993.

[16] Boland, N., Hamacher, H.W. and Lenzen, F.: *Minimizing Beam-On Time in Cancer Radiation Treatment Using Multileaf Collimators.* Technical report, University of Kaiserslautern, Department of Mathematics, 2002. Report in Wirtschaftsmathematik Nr. 78/2002.

[17] Bortfeld T., Kahler D.L., Waldron, T.J. and Boyer, A.L.: *X-ray field compensation with multileaf collimators.* Int. J. Radiation Oncology Biol. Phys, 28(3), 1994, 723– 730.

[18] Braennlund, U.: *A generalized subgradient method with relaxation step.* Mathematical Programming 71, 1995, pp. 207-219.

[19] Burkard, R.E., Leitner, H., Rudolf, R., Siegel, T., and Tabbert, E.: *Discrete optimization models for treatment planning in radiation therapy.* In Hutten, H., editor, Science and Technology for Medicine, Biomedical Engineering in Graz. Pabst, Lengerich, 1994.

[20] Burkard,R.E.: *Open Research section of Oberwolfach conference*, November 2002.

[21] Camerini, P.M., Fratta, L. and Maffioli, F.: *On improving relaxation methods by modified gradient techniques.* Mathematical Programming Study 3, 1975, pp. 26-34.

[22] Caprara, A., Fishetti, M. and Toth, P.: *A heuristic method for set covering problem.* Operations Research, 47(5), 1999, pp. 730-743.

[23] Christofides, N. J. and Beasley, E.: *extensions to a Lagrangian relaxation approach for the capacitated warehouse location problem.* European Journals of Operational Research, 12, 1983, pp. 19 - 28.

[24] Convey, D.J and Rosenbloom, M.E.: *The generation of intensity modulated fields of conformal radiation therapy by dynamic collimation.* Phys. Med. Biol. 37(6), 1992, 1359 – 1374.

[25] Cornuejolis, G., Fisher, M.L. and Nemhauser, G.L.: *Location of bank accounts to optimize float: An Analytic Study of Exact and Approximate Algorithms.* Management Science, 23, 1977, pp. 789 - 810.

[26] Dantzig, G.B. and Wolfe, P.: *Decomposition principle for linear programs.* Operations Research 8, 1960, pp. 101 - 111.

[27] Dem'yanov, V.F. and Shomesova, V.K.: *Conditional subdifferentials of convex functions.* Soviet Mathematics Doklady, 19, 1978, pp. 1181-1185.

[28] Dem'yanov, V.F. and Shomesova, V.K.: *Conditional subdifferentials of functions on sets.* Cybernetics 16(1), 1980, 24-31.

[29] Dem'yanov, V.F. and Vasil'ev, L.V.: *Nondifferentiable Optimization.* Optimization Software, Inc. Publications Division, New York, 1985.

[30] Erlenkotter, D.: *A dual based procedure for uncapacitated facility location.* Operations Research 26 (No. 1), 1978, 760 - 772.

[31] Ermolev, J.M. and Shor, N.Z.: *On optimization of non-differentiable functions.* Kibernetika 1, 1967, pp. 101 - 102.

[32] Etchebery, J.: *The set covering problem: A new implicit enumeration algorithm.* Operations Research, 205, 1977, pp. 760 - 772.

[33] Fisher, M.L.: *Optimal solution of scheduling problems using Lagrangian multipliers: Part I.* Operation Research, 21, 1973, PP. 1114 - 1127.

[34] Fisher, M.L.: *A dual algorithm for one-machine scheduling problem.* Mathematical Programming, 11, 1976, pp. 229 - 251.

[35] Fisher, M.L. and Shapiro, J.F.: *Constructive duality in integer programming.* SIAM, Journal of Applied Mathematics, 27, 1974, pp. 31 - 52.

[36] Fisher, M.L.: *The Lagrangian relaxation method for solving integer programming problems.* Management Science, Vol.27, No. 1, 1981, pp. 1-18.

[37] Frangioni, A.: *Dual-ascent methods and multicommodity flow problems.* Ph.D. Theseis - TD 5/97, WEB: http://www.di.unipi.it/ frangio, 1997.

[38] Galvin, J., Chen, X., and Smith, R.: *Combining multileaf field to modulate fluence distributions.* Int. J. Radiation Oncology Biol. Phys, 27, 697– 705, 1993.

[39] Geoffrion, A.M.: *Lagrangian relaxation for integer programming.* Mathematical Programming Study, 2(1974), pp. 82 – 114.

[40] Görg A.: *Anwendung von Netzwerkfluessen bei der Realisierung von Krebsbestrahlung durch Multileafkollimatoren .* Diplomarbeit, Universität Kaiserslautern, 2002.

[41] Hamacher, H.W., and Lenzen, F.: *A mixed integer programming approach to the multileaf collimator problem"*, in Wolfgang Shlegel and T. Bortfeld (edts.) *The use of computers in Radiation therapy*, Springer 2000, ISBN 3-540-67176-5, 210-212.

[42] Hamacher, H.W., Klamroth, K.: Linear and Network Optimization, Vieweg Verlag, 2000.

[43] Hamacher, W.H. and Küfer, K.H.: *Inverse radiation therapy planning: a multiple objective optimization approach.* Discrete Applied Mathematics, 118, 145 –161, 2002.

[44] Held, M. and Karp, R.M : *The travelling salesman problem and minimum spanning trees.* Operations Res., 18, 1970, pp. 1138-1162.

[45] Held, M. and Karp, R.M : *The travelling salesman problem and minimum spanning trees: Part II.* Mathematical Programming, 1, 1971, pp. 6-25.

[46] Held, M., Wolfe, P. and Crowder, H.P.: *Validation of subgradient optimization.* Mathematical Programming 6, 1974, pp. 62-88.

[47] Hiriart-Urruty, J.B. and Lemaréchal, C.: *Convex analysis and minimization Algorithms I.* Springer-Verlag, 1993.

[48] Hiriart-Urruty, J.B. and Lemaréchal, C.: *Convex analysis and minimization Algorithms II.* Springer-Verlag, 1993.

[49] Holder, A.: *Designing radiotherapy plans with elastic constraints and interior point methods.* Trinity University, Mathematics Technical Report No. 49, 2000.

[50] Juneberg, C.: *Large scale 0-1 minimization problems solved using dual subgradient schemes, ergodic sequences and core problems.* Master's Thesis, Goeteborg University, 2000.

[51] Kiwiel, K. C.: *Methods of descent for nondifferentiable Optimization.* Springer-Verlag, Berlin, 1985.

[52] Knopp, K.: *Infinite sequence and series.* Dover Publications, New York 1956.

[53] Larsson, T., Patriksson, M. and Strömberg, A.-B.: *Conditional subgradient optimization: theory and applications.* European Journal of Operations Research, 88, 1996, pp. 382-403.

[54] Larsson, T., Patriksson, M. and Strömberg, A.-B.: *Ergodic, primal convergence in dual subgradient schemes for convex programming.* Mathematical Programming, 86, 1999, pp. 283-312.

[55] Lasdon, L.S.: *Optimization Theory of Large Scale Systems.* MacMillan, 1970.

[56] Lee, E., Gallagher, R. and Zaider, M.: *Planning implants of radionuclides for the treatment of prostate cancer.* OPTIMA, Mathematical programming news letter, 1999.

[57] Lemaréchal C.: *Nondifferentiable optimization: Subgradient and ε-subgradient methods.* Lecture notes in optimization and operations research, Vol. 117, Springer-Verlag 1975.

[58] Lemaréchal C.: *Bundle methods in nonsmooth optimizations* in Nonsmooth Optimization , Vol. 3 of IIASA proceedings Series (C. Lemaréchall and R. Mifflin eds.), Pergamon Press, 1977.

[59] Lenzen, F.: *An Integer programming approach to the multileaf collimator problem.* Diploma Thesis, University of Kaiserslautern, 2000.

[60] Maekelea, M.M. and Neittaanmmaeki, P.: *Nonsmooth Optimization.* World Scientific Publishing Co. Pte. Ltd., 1992.

[61] Magnanti, T.L., Shapiro, J.F. and Wagner, M.H.: *Generalized linear programming solves the dual.* Management Science 22 , 1976, pp. 1195 - 1204.

[62] Morill, s., Lane, R., Wong, J. and Rosen, I.: *Dose-volume considerations with linear programming optimization.* Med. Phys. 18(16), 1991, 1201 – 1210.

[63] Murty, K.G.: *Network Programming.* Prentice-Hall, 1992.

[64] Nemhauser, G.L. and Wolsey, L.A.: *Integer and Combinatorial Optimization.* John Wiley & Sons, Inc., 1988

[65] Nocedal, J. and Wright, S.J.: *Numerical Optimization.* Springer- Verlag New York, Inc., 1999.

[66] Parker, R.G. and Rardin, R.L.: *Discrete Optimization.* Academic Press, Inc., 1988.

[67] Polyak, B.T.: *A general method of solving extremum problems.* Soviet Mathematics Doklady, 8, 1967, pp. 593 - 597.

[68] Polyak, B.T.: *Minimization of unsmooth functionals.* U.S.S.R. Computational Mathematics and Mathematical Physics, 9, 1969, 14-29.

[69] Que, W.: *Comparison of algorithms for multileaf collimator field segmentation.* Medical Physics, 26, 2390–2396, 1999.

[70] Rockafellar, R.T.: *Convex Analysis.* Princeton University Press, 1970.

[71] Rosen I., Lane, R., Morill,S. and Belli,J.: *Treatment plan using linear programming.* Med. Phys. 18(2), 1991, 141 – 152.

[72] Salkin, H.M.: *Integer Programming.* Addison-Wesley Publishing Company, inc., 1975.

[73] Sandi, C.: *Subgradient optimization.* Combinatorial Optimization, John Wiley & Sons, 1979, pp. 73- 91.

[74] Shapiro, J F : *Generalized Lagrange multipliers in integer programming.* Operations Research, 19, 1971, pp. 68 - 76.

[75] Shapiro, J.F.: *A survey of Lagrangian techniques for discrete optimization.* Annals of Discrete Mathematics,5, 1979, pp. 113 - 138.

[76] Shepard, D., Ferris, M., Olivera, G., and Mackie, T.: *Optimizing the delivery of radiation therapy to cancer patients.* SIAM Review, 41(1), 1999, 721–744.

[77] Sherali, H.D. and Ulular, O.: *A primal-dual conjugate subgradient algorithm for specially structured linear and convex programming problems.* Applied Mathematics and Optimization, 20, 1989, pp. 193-221.

[78] Sherali, H.D. and Choi, G.: *Recovery of primal solutions when using subgradient optimization methods to solve Lagrangian duals of linear programmes.* Operations Research Letters, 19, 1996, pp. 105-113.

[79] Shor, N.Z.: *An application of the gradient descent method for solving a transportation network problem.* Proceedings of the Scientific Seminar on Theoretical and Application Problems of Cybernetics and Operations Research, 1962, pp. 9 - 17.

[80] Shor, N.Z.: *Minimization of Methods for Non-Differentiable Functions.* Springer-Verlag, Berlin, 1985.

[81] Siochi, R.A: *Minimizing Static intensity modulation delivery time using an intensity solid paradigm.* Int. J. Radiation Oncology Biol. Phys., 43, 1999, 671-680.

[82] Svensson, R., Kaellman, P., and Brahme, A.: *An analytical solution for the dynamic control of multileaf collimators.* Physics in Medicine and Biology, 39, 1994, 37–61.

[83] Wolfe, P.: *A method of conjugate subgradient of minimizing non differentiable function.* Mathematical Programming Study 3, 1975, pp. 145-173.

[84] Wolsey, L.A.: *Integer Programming.* John Wiley & sons, Inc., 1998.

[85] Xia, P. and Verhey, L.J.: *Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple static segments.* Med. Phys. 25(8), 1998, 1424–1434.

Curriculum Vitae

1. Personal Data:

   - Name: Berhanu Guta Wordofa.

   - Date of Birth: 07, December 1966.

   - Place of Birth: Addis Alem, Ethiopia.

2. Education

   - 07/1983:   Ethiopian School Leaving Examination.

   - 09/83 - 07/87:   B.Sc in Mathematics, Addis Ababa University.

   - 09/89 - 07/92:   M.Sc in Mathematics, Addis Ababa University.

   - 10/94 - 09/96:   M.Sc in Industrial Mathematics,University of Kaiserslautern, Germany.

   - 04/2000 - 09/2003:   Ph.D Studies, University of Kasierslautern, Germany.

   - September 17, 2003:   Disputation.

3. Professional Career:

   - 1988 - 1992:   Teaching and Research assistant, Department of Mathematics, Addis Ababa University.

   - 1993 - 1994:   Lecturer, Department of Mathematics, Addis Ababa University.

   - 1997 - 1999:   Lecturer, Department of Mathematics, Addis Ababa University.