Copyright 2005
by
Srijit Kamath
Dedicated to my parents, who have always encouraged me to become a scientist.
ACKNOWLEDGMENTS

First and foremost, my utmost gratitude goes to my advisor, Sartaj Sahni, for the immeasurably valuable guidance and support he has given me during the period of my graduate study. He introduced me to this challenging computational problem in medical physics. He has patiently absorbed all my ideas, proofread every mathematical analysis that I have performed, and gently nudged me in the most promising direction on each occasion. His judgement and experience have proved to be crucial in the success of this work. I could not have asked for a better advisor. I feel that he is one of the best computer science educators in the world. It has been a privilege and an honor to work with him.

I am very thankful to Meera Sitharam for the several insights she has provided on algorithmic problems through her teaching in and out of class and for her general enthusiasm for problem solving that has led to many absorbing discussions. Serving as the coordinator of the algorithms and theory seminar, which was started with her encouragement and initiative, has also been an excellent learning experience for me.

Thanks go to Jonathan Li for collaborating on the problems and for providing clinical data. Thanks go also to Jatinder Palta, Anand Rangarajan and Sanjay Ranka for taking the time to serve on my committee.

Haejae Jung, Anuj Jain, Amita Malik, Mohit Dhawan, Haibin Lu, Jimin Yin and Wencheng Lu made CSE329 a good place to work. Intense preparation for the doctoral qualifying exam was possible thanks to late night study sessions with Vijay Manian and Pompi Diplan. I would like to thank all my friends (simply too many to name!) for their great company and support during the course of my graduate study, with special thanks to fellow students Subi, Manas, Anuj, Andrew, Pranav and Shantanu.

This work would not have been possible without the constant encouragement I have received from my parents and from uncle Narayan, who has also provided numerous insights on graduate study and work in the United States.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>x</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Problem Description</td>
<td>1</td>
</tr>
<tr>
<td>1.2 MLC Models and Constraints</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Prior Work</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Dissertation Outline</td>
<td>7</td>
</tr>
<tr>
<td>2 SEQUENCING OF SEGMENTED MULTILEAF COLLIMATORS</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Methods</td>
<td>8</td>
</tr>
<tr>
<td>2.1.1 Discrete Profile</td>
<td>8</td>
</tr>
<tr>
<td>2.1.2 Movement of Leaves</td>
<td>8</td>
</tr>
<tr>
<td>2.1.3 Optimal Unidirectional Algorithm for one Pair of Leaves</td>
<td>11</td>
</tr>
<tr>
<td>2.1.4 Bi-directional Movement</td>
<td>16</td>
</tr>
<tr>
<td>2.1.5 Algorithm Under Maximum Separation Constraint Condition</td>
<td>20</td>
</tr>
<tr>
<td>2.1.6 Algorithm Under Inter-Pair Minimum Separation Constraint</td>
<td>23</td>
</tr>
<tr>
<td>2.2 Conclusion</td>
<td>34</td>
</tr>
<tr>
<td>3 SEQUENCING OF DYNAMIC MULTILEAF COLLIMATORS</td>
<td>36</td>
</tr>
<tr>
<td>3.1 Methods</td>
<td>36</td>
</tr>
<tr>
<td>3.1.1 Movement of Leaves</td>
<td>36</td>
</tr>
<tr>
<td>3.1.2 Maximum Velocity Constraint</td>
<td>38</td>
</tr>
<tr>
<td>3.1.3 Optimal Unidirectional Algorithm for one Pair of Leaves</td>
<td>38</td>
</tr>
<tr>
<td>3.1.4 Minimum Separation Constraint</td>
<td>42</td>
</tr>
<tr>
<td>3.1.5 Bi-directional Movement</td>
<td>46</td>
</tr>
<tr>
<td>3.1.6 Algorithm Under Maximum Separation Constraint Condition</td>
<td>51</td>
</tr>
<tr>
<td>3.1.7 Algorithm Under Interdigitation Constraint</td>
<td>55</td>
</tr>
<tr>
<td>3.2 Conclusion</td>
<td>65</td>
</tr>
<tr>
<td>4 ELIMINATION OF TONGUE-AND-GROOVE UNDERDOSAGE</td>
<td>66</td>
</tr>
<tr>
<td>4.1 Algorithm with Interdigitation and Tongue-and-Groove Constraints</td>
<td>67</td>
</tr>
<tr>
<td>4.1.1 Tongue-and-Groove Underdosage Effect</td>
<td>67</td>
</tr>
<tr>
<td>4.1.2 Algorithms</td>
<td>68</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Efficient Implementation of the Algorithms</td>
</tr>
<tr>
<td>4.2</td>
<td>Experimental Validation</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison with Algorithm of Que et al. (2004)</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Analysis of the Algorithm of Que et al. (2004)</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Results</td>
</tr>
<tr>
<td>4.4</td>
<td>Conclusion</td>
</tr>
<tr>
<td>5</td>
<td>ALGORITHMS FOR SPLITTING LARGE FIELDS</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.2</td>
<td>Field Splitting Without Feathering</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Optimal Field Splitting for One Leaf Pair</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Optimal Field Splitting for Multiple Leaf Pairs</td>
</tr>
<tr>
<td>5.3</td>
<td>Field Splitting with Feathering</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Splitting a Profile into Two</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Splitting a Profile into Three</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Tongue-and-groove Effect and Interdigitation</td>
</tr>
<tr>
<td>5.4</td>
<td>Results</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
</tr>
<tr>
<td>6</td>
<td>CONCLUSION</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
</tr>
<tr>
<td></td>
<td>BIOGRAPHICAL SKETCH</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4–1</td>
<td>Comparison of the number of segments</td>
<td>83</td>
</tr>
<tr>
<td>4–2</td>
<td>Number of MUs and segments</td>
<td>88</td>
</tr>
<tr>
<td>4–3</td>
<td>Average number of MUs and segments</td>
<td>88</td>
</tr>
<tr>
<td>5–1</td>
<td>Total MUs for five clinical cases</td>
<td>108</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1–1</td>
<td>A linear accelerator</td>
<td>2</td>
</tr>
<tr>
<td>1–2</td>
<td>A multileaf collimator</td>
<td>3</td>
</tr>
<tr>
<td>1–3</td>
<td>Inter-pair minimum separation constraint</td>
<td>4</td>
</tr>
<tr>
<td>1–4</td>
<td>Cross section of leaves</td>
<td>5</td>
</tr>
<tr>
<td>2–1</td>
<td>Geometry and coordinate system</td>
<td>9</td>
</tr>
<tr>
<td>2–2</td>
<td>Discretization of profile</td>
<td>9</td>
</tr>
<tr>
<td>2–3</td>
<td>Leaf trajectory during SMLC delivery</td>
<td>10</td>
</tr>
<tr>
<td>2–4</td>
<td>Obtaining a unidirectional plan</td>
<td>12</td>
</tr>
<tr>
<td>2–5</td>
<td>A profile and its plan</td>
<td>13</td>
</tr>
<tr>
<td>2–6</td>
<td>Minimum separation constraint violation</td>
<td>15</td>
</tr>
<tr>
<td>2–7</td>
<td>Bi-directional movement</td>
<td>17</td>
</tr>
<tr>
<td>2–8</td>
<td>Bi-directional movement under minimum separation constraint</td>
<td>19</td>
</tr>
<tr>
<td>2–9</td>
<td>Bi-directional movement under maximum separation constraint</td>
<td>20</td>
</tr>
<tr>
<td>2–10</td>
<td>Obtaining a plan under maximum separation constraint</td>
<td>21</td>
</tr>
<tr>
<td>2–11</td>
<td>Maximum separation constraint violation</td>
<td>21</td>
</tr>
<tr>
<td>2–12</td>
<td>Obtaining a schedule</td>
<td>24</td>
</tr>
<tr>
<td>2–13</td>
<td>Obtaining a schedule under the constraint</td>
<td>25</td>
</tr>
<tr>
<td>2–14</td>
<td>Eliminating a violation</td>
<td>26</td>
</tr>
<tr>
<td>2–15</td>
<td>Eliminating a violation</td>
<td>30</td>
</tr>
<tr>
<td>2–16</td>
<td>Intensity profiles of adjacent leaf pairs</td>
<td>31</td>
</tr>
<tr>
<td>2–17</td>
<td>Profiles violating inter-pair constraint</td>
<td>32</td>
</tr>
<tr>
<td>3–1</td>
<td>Leaf trajectory during DMLC delivery</td>
<td>37</td>
</tr>
<tr>
<td>3–2</td>
<td>Obtaining a unidirectional plan</td>
<td>40</td>
</tr>
<tr>
<td>3–3</td>
<td>Obtaining a unidirectional plan with minimum separation constraint</td>
<td>44</td>
</tr>
<tr>
<td>3–4</td>
<td>Minimum separation constraint violation</td>
<td>44</td>
</tr>
</tbody>
</table>
In delivering radiation therapy for cancer treatment, it is desirable to deliver high doses of radiation to the target tumor, while permitting a low dosage on the surrounding healthy tissues. In recent years, the development of intensity modulated radiation therapy (IMRT) has made this possible. IMRT may be delivered by several techniques. The delivery of IMRT with a multileaf collimator (MLC) requires the delivery of radiation from several beam orientations. The intensity profile for each beam direction is described as a MLC leaf sequence, which is developed using a leaf sequencing algorithm. Important considerations in developing a leaf sequence for a desired intensity profile include maximizing the monitor unit (MU) efficiency (equivalently minimizing the beam-on time) and minimizing the total treatment time subject to the leaf movement constraints of the MLC model. In this work, we present a systematic study of the optimization of leaf sequencing algorithms and provide rigorous mathematical proofs of optimized leaf sequence settings in terms of MU efficiency under most common leaf movement constraints that include minimum and maximum leaf separation, leaf interdigitation and tongue-and-groove. We also develop algorithms to split large intensity modulated fields into two or three subfields.
CHAPTER 1
INTRODUCTION

1.1 Problem Description

The objective of radiation therapy for cancer treatment is to deliver high doses of radiation to the target tumor, while permitting a low dosage on the surrounding healthy tissues. For example, for head and neck tumors, it is necessary for radiation to be delivered so that the exposure of the spinal cord, optic nerve, salivary glands or other important structures is minimized. In recent years, this has been made possible due to the development of conformal radiation therapy. In conformal therapy, treatment is delivered using a set of radiation beams which are positioned such that the shape of the dose distribution “conforms” with the shape of the tumor. This is typically achieved by positioning beams of varying shapes from different directions so that each beam approximately irradiates the section of the tumor visible from its direction and avoids the organs at risk in the vicinity of the tumor.

Intensity modulated radiation therapy (IMRT) is the state-of-the-art in conformal radiation therapy. IMRT permits the intensity of a radiation beam to be varied across a treatment area, thereby improving the dose conformity. Radiation is delivered using a medical linear accelerator (Figure 1–1). A rotating gantry containing the accelerator structure can rotate around the patient who is positioned on an adjustable treatment couch. Delivery of IMRT is possible by several techniques. In compensator-based IMRT, the beam is modulated with a preshaped piece of material called the compensator (modulator). The degree of modulation of the beam varies depending on the thickness of the material through which the beam is attenuated. The computer determines the shape of each modulator in order to deliver the desired beam. This type of modulation requires the modulator to be fabricated and then manually inserted into the tray mount of a linear accelerator. In tomotherapy-based IMRT, the linear accelerator travels in multiple circles all the way around the gantry ring to deliver the radiation treatment. The beam is collimated to a narrow slit and the intensity of the
beam is modulated during the gantry movement around the patient. Care must be taken to ensure that adjacent circular arcs do not overlap and thereby do not overdose tissues. This type of delivery is referred to as serial tomotherapy. A modification of serial tomotherapy is helical tomotherapy. In helical tomotherapy, the treatment couch moves linearly (continuously) through the rotating accelerator gantry. So each time the accelerator comes around, it directs the beam on a slightly different plane on the patient. In MLC-based IMRT the accelerator structure is equipped with a computer controlled mechanical device called a multileaf collimator (MLC, Figure 1–2) that shapes the radiation beam, so as to deliver the radiation as prescribed by the treatment plan. The MLC may have up to 120 movable leaves that can move along an axis perpendicular to the beam and can be arranged so as to shield or expose parts of the anatomy during treatment. The leaves are arranged in pairs so that each leaf pair forms one row of the arrangement. The set of allowable MLC leaf configurations may be restricted by leaf movement constraints that are manufacturer and/or model dependent.

Figure 1–1: A linear accelerator (the figure is from http://www.lexmed.com/-medical_services/IMRT.htm)

The first stage in the treatment planning process in IMRT is to obtain accurate three dimensional anatomical information about the tumor and its surroundings. This is achieved
using computed tomography (CT) and magnetic resonance (MR) imaging. An ideal dose distribution would ensure perfect conformity to the target volume while completely sparing all other tissues. However, such a distribution is impossible to realize in practice. Therefore, minimum dose targets for tumors and tolerable doses for critical structures are prescribed and an objective function that measures the quality of a plan is developed subject to these dose based constraints. Next, a set of beam parameters (beam angles, profiles, weights) that optimize this objective are determined using a computer program. This method is called “inverse planning” since resultant dose distribution is first described and the best beam parameters that deliver the distribution (approximately) are then solved for. It is to be noted that inverse planning is a general concept and its implementation details vary vastly among various systems. Following the inverse planning in MLC-based IMRT, the delivery of radiation intensity profile for each beam direction is described as a MLC leaf sequence, which is developed using a leaf sequencing algorithm. Important considerations in developing a leaf sequence for a desired intensity profile include maximizing the monitor unit (MU) efficiency (equivalently minimizing the beam-on time) and minimizing the total treatment time subject to the leaf movement constraints of the MLC model. Finally, when the leaf sequences for all beam directions are determined, the treatment is performed from
various beam angles sequentially using computer control. In this work, we develop optimized leaf sequencing algorithms for various MLC models.

1.2 MLC Models and Constraints

The purpose of the leaf sequencing algorithm is to generate a sequence of leaf positions and/or movements that faithfully reproduce the desired intensity map once the beam is delivered, taking into consideration any hardware and dosimetric characteristics of the delivery system. The two most common methods of IMRT delivery with computer-controlled MLCs are the segmental multileaf collimator (SMLC) and dynamic multileaf collimator (DMLC). In SMLC, the beam is switched off while the leaves are in motion. In other words, the delivery is done using multiple static segments or leaf settings. This method is also frequently referred to as the “step and shoot” or “stop and shoot” method. In DMLC the beam is on while the leaves are in motion. The beam is switched on at the start of treatment and is switched off only at the end of treatment. The fundamental difference between the leaf sequences of these two delivery methods is that the leaf sequence defines a finite set of beam shapes for SMLC and trajectories of opposing pairs of leaves for DMLC.

In practical situations, there are some constraints on the movement of the leaves. The minimum separation constraint requires that opposing pairs of leaves be separated by at least some distance ($S_{\text{min}}$) at all times during beam delivery. In MLCs this constraint is applied not only to opposing pairs of leaves, but also to opposing leaves of neighboring pairs. For example, in Figure 1–3, $L_1$ and $R_1$, $L_2$ and $R_2$, $L_3$ and $R_3$, $L_1$ and $R_2$, $L_2$ and $R_1$, $L_2$ and $R_3$, $L_3$ and $R_2$ are pairwise subject to the constraint. The case with $S_{\text{min}} = 0$ is called interdigitation constraint and is applicable to some MLC models. Wherever this constraint applies, opposite adjacent leaves are not permitted to overlap.

![Figure 1–3: Inter-pair minimum separation constraint](image)
In most commercially available MLCs, there is a tongue-and-groove arrangement at the interface between adjacent leaves. A cross section of two adjacent leaves is depicted in Figure 1–4. The width of the tongue-and-groove region is \( l \). The area under this region gets underdosed due to the mechanical arrangement, as it remains shielded if either the tongue or the groove portion of a leaf shields it.

![Cross section of leaves](image)

Figure 1–4: Cross section of leaves

Maximum leaf spread for leaves on the same leaf bank is one more MLC limitation, which necessitates a large field (intensity profile) to be split into two or more adjacent abutting sub-fields. This is true for the Varian MLC (Varian Medical Systems, Palo Alto, CA), which has a field size limitation of about 15 cm. The abutting sub-fields are then delivered as separate treatment fields. This often results in longer delivery times, poor MU efficiency, and field matching problems.

### 1.3 Prior Work

Optimization of the leaf sequencing algorithm has been the subject of numerous investigations (for example, Convery and Rosenbloom 1992, Bortfeld et al. 1994a, Dirkx et al. 1998, Ma et al. 1998, Xia and Verhey 1998, Siochi 1999, Langer et al. 2001, Luan et al. 2003, Chen et al. 2004). Treatment delivery with IMRT is not very efficient in terms of MU efficiency, which is defined as the ratio of dose delivered at a point in the patient with an IMRT field to the MU delivered for that field. Typical MU efficiencies of IMRT treatment
plans are 3 to 10 times lower than those for open/wedge field-based conventional treatment plans. Therefore, total body dose due to increased leakage radiation reaching the patient in an IMRT treatment is a major concern (Followill et al. 1997, Intensity Modulated Radiation Therapy Collaborative Working Group 2001). Low MU efficiency of IMRT delivery negatively impacts the room shielding design because of the increased workload (Intensity Modulated Radiation Therapy Collaborative Working Group 2001, Mutic et al. 2001). The MU efficiency depends on both the degree of intensity modulation and the algorithm used to convert the intensity pattern into a leaf sequence for IMRT delivery. It is therefore important to design a leaf sequencing algorithm that is optimal for MU efficiency to minimize total body dose to the patient. For dynamic beam delivery where dose rate is usually not modulated, an algorithm that optimizes the MU setting at a given dose rate also optimizes the treatment time.

Dynamic leaf sequencing algorithms with the leaves in motion during radiation delivery have been developed (Convery and Rosenbloom 1992, Spirou and Chui 1994), and later modified (van Santvoort and Heijmen 1996, Dirkx et al. 1998) to eliminate the tongue-and-groove underdosage effects. Similar leaf sequencing algorithms have also been developed for the segmental multileaf collimator (SMLC) delivery method (Bortfeld et al. 1994a, Bortfeld et al. 1994b, Ma et al. 1998, Xia and Verhey 1998, Que 1999, Engel 2003, Kalinowski 2003, Li et al. 2003). Many of these studies did not consider any leaf movement constraints. Such leaf sequencing algorithms are applicable for certain types of MLC designs. For other types of MLC designs, notably the Siemens (Siemens Medical Systems, Inc., Iselin, NJ) MLC design (Das et al. 1998) and Elekta (Elekta Oncology Systems Inc., Norcross, GA) MLC design (Jordan and Williams 1994), other mechanical constraints need to be taken into consideration when designing the leaf settings for both dynamic and SMLC delivery. The minimum leaf separation constraint, for example, was recently incorporated into the design of leaf sequence (Convery and Webb 1998). A general description and characteristics of some MLC models can be found in Xia and Verhey (2001).
1.4 Dissertation Outline

In this work, we present a systematic study of the optimization of leaf sequencing algorithms. The dissertation is organized as follows. In chapter 2, we present leaf sequencing algorithms for the SMLC beam delivery and provide rigorous proofs of optimized leaf sequence settings in terms of MU efficiency under various leaf movement constraints. Practical leaf movement constraints that are considered include the minimum leaf separation constraint and minimum inter-leaf separation constraint (leaf interdigitation constraint). The question of whether bi-directional leaf movement will increase the MU efficiency when compared with uni-directional leaf movement only is also addressed. In chapter 3, we develop leaf sequencing algorithms for DMLC beam delivery. Algorithms are presented to sequence leaves with maximum leaf separation constraint and the leaf interdigitation constraint. In chapter 4, we study tongue-and-groove effect for SMLC. We provide bounds on the maximum extent to which tongue-and-groove effect can be eliminated and give necessary and sufficient conditions for a unidirectional leaf sequence to attain the bound. We then present algorithms that generate leaf sequences that eliminate the tongue-and-groove effect and optionally satisfy the interdigitation constraint. We also compare our algorithms to a recently published leaf sequencing algorithm that also eliminates tongue-and-groove underdosage. The problem of splitting large intensity modulated fields into two or three subfields is discussed in chapter 5. All our algorithms generate unidirectional leaf movement schedules and are proved to be optimal in MUs for unidirectional schedules.
CHAPTER 2
SEQUENCING OF SEGMENTED MULTILEAF COLLIMATORS

In this chapter, we present a systematic study of the optimization of leaf sequencing algorithms for the SMLC beam delivery and provide rigorous proofs of optimized leaf sequence settings in terms of MU efficiency under various leaf movement constraints. Practical leaf movement constraints that are considered include the minimum leaf separation constraint and minimum inter-leaf separation constraint (leaf interdigitation constraint). The question of whether bi-directional leaf movement will increase the MU efficiency when compared with uni-directional leaf movement only is also addressed. We first introduce the notation that will be used in the remainder of this work.

2.1 Methods

2.1.1 Discrete Profile

The geometry and coordinate system used in this study are shown in Figure 2–1. We consider delivery of profiles that are piecewise continuous. Let \( I(x) \) be the desired intensity profile. We first discretize the profile so that we obtain the values at sample points \( x_0, x_1, x_2, \ldots, x_m \). \( I(x) \) is assigned the value \( I(x_i) \) for \( x_i \leq x < x_{i+1} \), for each \( i \). Now, \( I(x_i) \) is our desired intensity profile. Figure 2–2 shows a piecewise continuous function and the corresponding discretized profile. The discretized profile can be efficiently delivered with the SMLC method. However, a SMLC sequence can be transformed to a dynamic leaf sequence by allowing both leaves to start at the same point and close together at the same point, so that they sweep across the same spatial interval. We develop our theory for the SMLC delivery.

2.1.2 Movement of Leaves

In our analysis we assume that the leaves are initially at the left most position \( x_0 \) and that the leaves move unidirectionally from left to right. Figure 2–3 illustrates the leaf trajectory during SMLC delivery. Let \( I_l(x_i) \) and \( I_r(x_i) \) respectively denote the amount of monitor units (MUs) delivered when the left and right leaves leave position \( x_i \). Consider
Figure 2–1: Geometry and coordinate system

Figure 2–2: Discretization of profile
the motion of the left leaf. The left leaf begins at $x_0$ and remains here until $I_l(x_0)$ MUs have been delivered. At this time the left leaf is moved to $x_1$, where it remains until $I_l(x_1)$ MUs have been delivered. The left leaf then moves to $x_3$ where it remains until $I_l(x_3)$ MUs have been delivered. At this time, the left leaf is moved to $x_6$, where it remains until $I_l(x_6)$ MUs have been delivered. The final movement of the left leaf is to $x_7$, where it remains until $I_l(x_7) = I_{\text{max}}$ MUs have been delivered. At this time the machine is turned off. The total therapy time, $TT(I_l, I_r)$, is the time needed to deliver $I_{\text{max}}$ MUs. The right leaf starts at $x_2$; moves to $x_4$ when $I_r(x_2)$ MUs have been delivered; moves to $x_5$ when $I_r(x_4)$ MUs have been delivered and so on. Note that the machine is off when a leaf is in motion. We make the following observations:

1. All MUs that are delivered along a radiation beam along $x_i$ before the left leaf passes $x_i$ fall on it. The greater the $x$ value, the later the leaf passes that position. Therefore $I_l(x_i)$ is a non-decreasing function.

2. All MUs that are delivered along a radiation beam along $x_i$ before the right leaf passes $x_i$ are blocked by the leaf. The greater the $x$ value, the later the leaf passes that position. Therefore $I_r(x_i)$ is also a non-decreasing function.

Figure 2–3: Leaf trajectory during SMLC delivery
From these observations we notice that the net amount of MUs delivered at a point is given by \( I_l(x_i) - I_r(x_i) \), which must be the same as the desired profile \( I(x_i) \).

2.1.3 Optimal Unidirectional Algorithm for one Pair of Leaves

Unidirectional movement. When the movement of leaves is restricted to only one direction, both the left and right leaves move along positive \( x \) direction, from left to right (Figure 2–1). Once the desired intensity profile, \( I(x_i) \) is known, our problem becomes that of determining the individual intensity profiles to be delivered by the left and right leaves, \( I_l \) and \( I_r \) such that

\[
I(x_i) = I_l(x_i) - I_r(x_i), \quad 0 \leq i \leq m \tag{2.1}
\]

We refer to \((I_l, I_r)\) as the treatment plan (or simply plan) for \( I \). Once we obtain the plan, we will be able to determine the movement of both left and right leaves during the therapy. For each \( i \), the left leaf can be allowed to pass \( x_i \) when the source has delivered \( I_l(x_i) \) MUs. Also, we can allow the right leaf to pass \( x_i \) when the source has delivered \( I_r(x_i) \) MUs. In this manner we obtain unidirectional leaf movement profiles for a plan.

Algorithm. From Equation 2.1, we see that one way to determine \( I_l \) and \( I_r \) from the given target profile \( I \) is to begin with \( I_l(x_0) = I(x_0) \) and \( I_r(x_0) = 0 \); examine the remaining \( x_i \)'s from left to right; increase \( I_l \) whenever \( I \) increases; and increase \( I_r \) whenever \( I \) decreases. Once \( I_l \) and \( I_r \) are determined the leaf movement profiles are obtained as explained in the previous section. The resulting algorithm is shown in Figure 2–4. Figure 2–5 shows a profile and the corresponding plan obtained using the algorithm.

Ma et al. (1998) shows that Algorithm SINGLEPAIR obtains plans that are optimal in therapy time. Their proof relies on the results of Spirou and Chui (1994), Stein et al. (1994) and Boyer and Strait (1997). We provide a much simpler proof below.

Theorem 1 Algorithm SINGLEPAIR obtains plans that are optimal in therapy time.

Proof: Let \( I(x_i) \) be the desired profile. Let \( inc_1, inc_2, \ldots, inc_k \) be the indices of the points at which \( I(x_i) \) increases. So \( x_{inc_1}, x_{inc_2}, \ldots, x_{inc_k} \) are the points at which \( I(x) \) increases (i.e., \( I(x_{inc_i}) > I(x_{inc_{i-1}}) \)). Let \( \Delta i = I(x_{inc_i}) - I(x_{inc_{i-1}}) \).

Suppose that \((I_L, I_R)\) is a plan for \( I(x_i) \) (not necessarily that generated by Algorithm
Algorithm SINGLEPAIR

\[ I_l(x_0) = I(x_0) \]
\[ I_r(x_0) = 0 \]

For \( j = 1 \) to \( m \) do
- If \( (I(x_j) \geq I(x_{j-1})) \)
  \[ I_l(x_j) = I_l(x_{j-1}) + I(x_j) - I(x_{j-1}) \]
  \[ I_r(x_j) = I_r(x_{j-1}) \]
- Else
  \[ I_r(x_j) = I_r(x_{j-1}) + I(x_j) - I(x_{j-1}) \]
  \[ I_l(x_j) = I_l(x_{j-1}) \]
End for

Figure 2–4: Obtaining a unidirectional plan

SINGLEPAIR). From the unidirectional constraint, it follows that \( I_L(x_i) \) and \( I_R(x_i) \) are non-decreasing functions of \( x \). Since \( I(x_i) = I_L(x_i) - I_R(x_i) \) for all \( i \), we get

\[ \Delta i = (I_L(x_{inci}) - I_R(x_{inci})) - (I_L(x_{inci-1}) - I_R(x_{inci-1})) \]
\[ = (I_L(x_{inci}) - I_L(x_{inci-1})) - (I_R(x_{inci}) - I_R(x_{inci-1})) \]
\[ \leq I_L(x_{inci}) - I_L(x_{inci-1}) \]

Summing up \( \Delta i \), we get

\[ \sum_{i=1}^{k}[I(x_{inci}) - I(x_{inci-1})] \leq \sum_{i=1}^{k}[I_L(x_{inci}) - I_L(x_{inci-1})] = TT(I_L, I_R) \]

Since the therapy time for the plan \((I_l, I_r)\) generated by Algorithm SINGLEPAIR is \( \sum_{i=1}^{k}[I(x_{inci}) - I(x_{inci-1})] \), it follows that \( TT(I_l, I_r) \) is minimum.

Corollary 1 Let \( I(x_i) \), \( 0 \leq i \leq m \) be a desired profile. Let \( I_l(x_i) \) and \( I_r(x_i) \), \( 0 \leq i \leq m \) be the left and right leaf profiles generated by Algorithm SINGLEPAIR. \( I_l(x_i) \) and \( I_r(x_i) \), \( 0 \leq i \leq m \) define optimal therapy time unidirectional left and right leaf profiles for \( I(x_i) \), \( 0 \leq i \leq m \).

Proof: Follows from Theorem 1

In the remainder of this paper, \((I_l, I_r)\) is the optimal treatment plan for the desired profile \( I \).

Properties of the optimal treatment plan. The following observations are made about the optimal treatment plan \((I_l, I_r)\) generated using Algorithm SINGLEPAIR.

Lemma 1 At each \( x_i \) at most one of the profiles \( I_l \) and \( I_r \) changes (increases).
Figure 2–5: A profile and its plan
Lemma 2 Let \((I_L, I_R)\) be any treatment plan for \(I\).

(a) \(\Delta(x_i) = I_L(x_i) - I_l(x_i) = I_R(x_i) - I_r(x_i) \geq 0, 0 \leq i \leq m.\)

(b) \(\Delta(x_i)\) is a non-decreasing function.

Proof: (a) Since \(I(x_i) = I_L(x_i) - I_R(x_i) = I_l(x_i) - I_r(x_i), I_L(x_i) - I_l(x_i) = I_R(x_i) - I_r(x_i).\)

Further, from Corollary 1, it follows that \(I_L(x_i) \geq I_l(x_i), 0 \leq i \leq m.\) Therefore, \(\Delta(x_i) \geq 0, 0 \leq i \leq m.\)

(b) We prove this by contradiction. Suppose that \(\Delta(x_n) > \Delta(x_{n+1})\) for some \(n, 0 \leq n < m.\) Consider the following three all encompassing cases.

Case 1: \(I_l(x_n) = I_l(x_{n+1})\)

Now, \(I_L(x_n) = I_l(x_n) + \Delta(x_n) > I_l(x_{n+1}) + \Delta(x_{n+1}) = I_L(x_{n+1}).\)

This is not possible because \(I_L\) is a non-decreasing function.

Case 2: \(I_r(x_n) = I_r(x_{n+1})\)

Now, \(I_R(x_n) = I_r(x_n) + \Delta(x_n) > I_r(x_{n+1}) + \Delta(x_{n+1}) = I_R(x_{n+1}).\)

This contradicts the fact that \(I_R\) is a non-decreasing function.

Case 3: \(I_l(x_n) \neq I_l(x_{n+1})\) and \(I_r(x_n) \neq I_r(x_{n+1})\)

From Lemma 1 it follows that this case cannot arise.

Therefore, \(\Delta(x_i)\) is a non-decreasing function. ■

Theorem 2  If the optimal plan \((I_l, I_r)\) violates the minimum separation constraint, then there is no plan for \(I\) that does not violate the minimum separation constraint.

Proof: Suppose that \((I_l, I_r)\) violates the minimum separation constraint. Assume that the first violation occurs when \(I_1\) MUs have been delivered. From the unidirectional movement constraint, it follows that the left leaf has just been positioned at \(x_j\) (for some \(j, 0 \leq j \leq m\)) at this time and that the right leaf is at \(x_k\) such that \(x_k - x_j\) is less than the permissible minimum separation. Figure 2–6 illustrates the situation.

We prove the theorem by contradiction. Let \((I_L, I_R)\) be a plan that does not violate the minimum separation constraint. When \(j = 0, (I_l, I_r)\) has a violation at the initial positioning \(x_0\) of the left leaf. Since the leaves move in only one direction, the violation is when \(I_1 = 0.\) When \(I_1 = 0,\) the left leaf in \((I_L, I_R)\) is also at \(x_0\) (because the left leaf must begin at \(x_0\) in all plans; otherwise \(I(x_0) = 0).\) For \((I_L, I_R)\) not to have a violation at \(I_1 = 0,\) the right leaf must begin to the right of \(x_k,\) say at some point \(p > x_k\) (note
that \( p \) may not be one of the \( x_i \)s. The MUs delivered at \( x_k \) by the plan \((I_L, I_R)\) are \( I_L(x_k) - I_R(x_k) = I_L(x_k) \geq I_l(x_k) \) (Corollary 1). Also, \( I_l(x_k) = I(x_k) + I_r(x_k) > I(x_k) \) \((I_r(x_k) > 0)\). So \((I_L, I_R)\) delivers more than \( I(x_k) \) MUs at \( x_k \) and so is not a plan for \( I \). This contradicts the assumption on \((I_L, I_R)\). Hence, \( j \neq 0 \).

Suppose that \( j > 0 \). Now, \( I_l(x_j) > I_l(x_{j-1}) \). Also, \( I_L(x_j) = I_l(x_j) + \Delta(x_j) \) and \( I_L(x_{j-1}) = I_l(x_{j-1}) + \Delta(x_{j-1}) \). Since \( \Delta(x_j) \geq \Delta(x_{j-1}) \) (Lemma 2(b)), \( I_L(x_j) > I_L(x_{j-1}) \). Therefore, the left leaf is positioned at \( x_j \) at some time during the on cycle of the plan \((I_L, I_R)\). Let the amount of MUs delivered when the left leaf arrives at \( x_j \) in \( I_L \) be \( I_2 \). Let the right leaf be at \( x = p \) at this time. Note that \( p \) may not be one of the \( x_i \)s. If \( p > x_k \), then \( I_R(x_k) \leq I_2 \). Also, from Lemma 2 we have \( I_L(x_k) = I_l(x_k) + \Delta(x_k) \geq I_l(x_k) + \Delta(x_{k-1}) = I_l(x_k) + I_2 - I_1 > I_l(x_k) + I_2 - I_r(x_k) = I(x_k) + I_2 \). Therefore, \( I_l(x_k) - I_R(x_k) > I(x_k) \). This contradicts \( I_L(x_k) - I_R(x_k) = I(x_k) \) (since \((I_L, I_R)\) is a plan for \( I \)). Therefore, \( j \) cannot be \( > 0 \) either. So, there is no plan \((I_L, I_R)\) that does not violate the minimum separation constraint.

The separation between the leaves is determined by the difference in \( x \) values of the leaves when the source has delivered a certain amount of MUs. The minimum separation of the leaves is the minimum separation between the two profiles. We call this minimum

---

**Figure 2–6: Minimum separation constraint violation**

[Graph showing separation between the jaws and leaves]
separation $S_{ud-min}$. When the optimal plan obtained using Algorithm SINGLEPAIR is delivered, the minimum separation is $S_{ud-min(\text{opt})}$.

**Corollary 2** Let $S_{ud-min(\text{opt})}$ be the minimum leaf separation in the plan $(I_l, I_r)$. Let $S_{ud-min}$ be the minimum leaf separation in any (not necessarily optimal) given unidirectional plan. $S_{ud-min} \leq S_{ud-min(\text{opt})}$.

2.1.4 Bi-directional Movement

In this section we study beam delivery when bi-directional movement of leaves is permitted. We explore whether relaxing the unidirectional movement constraint helps improve the efficiency of treatment plan.

**Properties of bi-directional movement.** For a given leaf (left or right) movement profile we classify any $x$-coordinate as follows. Draw a vertical line at $x$. If the line cuts the leaf profile exactly once we will call $x$ a *unidirectional point* of that leaf profile. If the line cuts the profile more than once, $x$ is a *bi-directional point* of that profile. A leaf movement profile that has at least one bi-directional point is a *bi-directional profile*. All profiles that are not bi-directional are *unidirectional profiles*. Any profile can be partitioned into segments such that each segment is a unidirectional profile. When the number of such partitions is minimal, each partition is called a *stage* of the original profile. Thus unidirectional profiles consist of exactly one stage while bi-directional profiles always have more than one stage.

In Figure 2–7, the leaf movement profile, $B_l$, shows the position of the left leaf as a function of the amount of MUs delivered by the source. The leaf starts from the left edge and moves in both directions during the therapy. Clearly, $B_l$ is bi-directional. The movement profile of this leaf consists of stages $S_1$, $S_2$ and $S_3$. In stages $S_1$ and $S_3$ the leaf moves from left to right while in stage $S_2$ the leaf moves from right to left. $x_j$ is a bi-directional point of $B_l$. The amount of MUs delivered at $x_j$ is $L_1 + L_2$. In stage $S_1$, when $L_1$ amount of MUs have been delivered, the leaf passes $x_j$. Now, no MU is delivered at $x_j$ till the leaf passes over $x_j$ in $S_2$. Further, $L_2$ MUs are delivered to $x_j$ in stages $S_2$ and $S_3$. Thus we have $I_l(x_j) = L_1 + L_2$. Here, $L_1 = I_1, L_2 = I_3 - I_2$. $x_k$ is a unidirectional point of $B_l$. The MUs delivered at $x_k$ are $L_3 = I_4$. Note that the intensity profile $I_l$ is different from the leaf movement profile $B_l$, unlike in the unidirectional case.
Lemma 3 Let \((I_l, I_r)\) be a plan delivered by the bi-directional leaf movement profile pair \((B_l, B_r)\) (i.e., \(B_l\) and \(B_r\) are, respectively, the left and right leaf movement profiles)

\(a\) \(I_l\) is non-decreasing.

\(b\) \(I_r\) is non-decreasing.

Proof: (a) Whenever a point \(x_i, 0 \leq i \leq m\), is blocked by the left leaf, the points \(x_0, x_1, \ldots, x_{i-1}\) are also blocked. It follows that \(I_l(x_i) \geq I_l(x_j), 0 \leq j \leq i \leq m\).

(b) The proof is similar to (a).

From Lemma 3 we note that a bi-directional leaf movement profile \(B\) delivers a non-decreasing intensity profile. This non-decreasing intensity profile can also be delivered using a unidirectional leaf movement profile (Section 2.1.3). We will call this profile the unidirectional leaf movement profile that corresponds to the bi-directional profile \(B\) and we will denote it by \(U\) to emphasize that it is unidirectional. Thus every bi-directional leaf movement profile has a corresponding unidirectional leaf profile that delivers the same amount of MUs at each \(x_i\) as does the bi-directional profile.
Theorem 3 The unidirectional treatment plan constructed by Algorithm SINGLEPAIR is optimal in therapy time even when bi-directional leaf movement is permitted.

Proof: Let $B_L$ and $B_R$ be bidirectional leaf movement profiles that deliver a desired intensity profile $I$. Let $I_L$ and $I_R$, respectively, be the intensity profiles for $B_L$ and $B_R$. From Lemma 3, we know that $I_L$ and $I_R$ are non-decreasing. Also, $I_L(x_i) - I_R(x_i) = I(x_i), 1 \leq i \leq m$. From the proof of Theorem 1, it follows that the therapy time for the unidirectional plan $(I_L, I_R)$ generated by Algorithm SINGLEPAIR is no more than that of $(I_L, I_R)$.

Incorporating minimum separation constraint. Let $U_L$ and $U_R$ be unidirectional leaf movement profiles that deliver the desired profile $I(x_i)$. Let $B_L$ and $B_R$ be a set of bi-directional left and right leaf profiles such that $U_L$ and $U_R$ correspond to $B_L$ and $B_R$ respectively, i.e., $(B_L, B_R)$ delivers the same plan as $(U_L, U_R)$. We call the minimum separation of leaves in this bi-directional plan $(B_L, B_R)$ $S_{bd-min}$.

Theorem 4 $S_{bd-min} \leq S_{ud-min}$ for a bi-directional leaf movement profile pair and its corresponding unidirectional profile.

Proof: Suppose that the minimum separation $S_{ud-min}$ occurs when $I_{ms}$ MUs are delivered. At this time, the left leaf arrives at $x_j$ and the right leaf is positioned at $x_k$. Let $B'_L$ and $U'_L$ respectively, be the profiles obtained when $B_L$ and $U_L$ are shifted right by $S_{ud-min}$. Since $U'_L$ is $U_L$ shifted right by $S_{ud-min}$ and since the distance between $U_L$ and $U_R$ is $S_{ud-min}$ when $I_{ms}$ MUs have been delivered, $U'_L$ and $U_R$ touch when $I_{ms}$ units have been delivered. Therefore, the total MUs delivered by $(U'_L, U_R)$ at $x_k$ is zero. So the total MUs delivered by $(B'_L, B_R)$ at $x_k$ is also zero (recall that $U'_L$ and $U_R$, respectively, are corresponding profiles for $B'_L$ and $B_R$). This isn’t possible if $B_R$ is always to the right of $B'_L$ (for example, in the situation of Figure 2–8, the MUs delivered by $(B'_L, B_R)$ at $x_k$ are $(L_1 + L_2) - (L'_1 + L'_2 + L'_3) > 0$). Therefore $B'_L$ and $B_R$ must touch (or cross) at least once. So $S_{bd-min} \leq S_{ud-min}$.

Theorem 5 If the optimal unidirectional plan $(I_L, I_R)$ violates the minimum separation constraint, then there is no bi-directional movement plan that does not violate the minimum separation constraint.
**Proof:** Let $B_l$ and $B_r$ be bi-directional leaf movements that deliver the required profile. Let the minimum separation between the leaves be $S_{bd-min}$. Let the corresponding unidirectional leaf movements be $U_l$ and $U_r$ and let $S_{ud-min}$ be the minimum separation between $U_l$ and $U_r$. Also, let $S_{min}$ be the minimum allowable separation between the leaves. From Corollary 2 and Theorem 4, we get $S_{bd-min} \leq S_{ud-min} \leq S_{ud-min(opt)} < S_{min}$. 

**Incorporating maximum separation constraint.** Let $U_l$ and $U_r$ be unidirectional leaf movement profiles that deliver the desired profile $I$. Let $S_{ud-max}$ be the maximum leaf separation using the profiles $U_l$ and $U_r$ and let $S_{ud-max(opt)}$ be the maximum leaf separation for the plan $(I_l, I_r)$. Let $B_l$ and $B_r$ be a set of bi-directional left and right leaf profiles such that $U_l$ and $U_r$ correspond to $B_l$ and $B_r$, respectively. Let $S_{bd-max}$ be the maximum separation between the leaves when these bi-directional movement profiles are used.

**Theorem 6** $S_{bd-max} \geq S_{ud-max}$ for every bi-directional leaf movement profile and its corresponding unidirectional movement profile.

**Proof:** Suppose that the maximum separation $S_{ud-max}$ occurs when $I_{ms}$ MUs are delivered. At this time, the left leaf is positioned at $x_j$ and the right leaf arrives at $x_k$. Let $B_l'$ and $U_l'$ respectively, be the profiles obtained when $B_l$ and $U_l$ are shifted right by $S_{ud-max}$. 

Figure 2-8: Bi-directional movement under minimum separation constraint
Since $U'_l$ is $U_l$ shifted right by $S_{ud-max}$ and since the distance between $U_l$ and $U_r$ is $S_{ud-max}$ when $I_{ms}$ MUs have been delivered, $U'_l$ and $U_r$ touch when $I_{ms}$ units have been delivered. Therefore, the total MUs delivered by $(U_r, U'_l)$ at $x_k$ is zero. So the total MUs delivered by $(B_r, B'_l)$ at $x_k$ is also zero (recall that $U'_l$ and $U_r$, respectively, are corresponding profiles for $B'_l$ and $B_r$). This isn’t possible if $B_r$ is always to the left of $B'_l$ (for example, in the situation of Figure 2–9, the MUs delivered by $(B_r, B'_l)$ at $x_k$ are $(L'_1 + L'_2 + L'_3) - (L_1 + L_2) > 0$). Therefore $B'_l$ and $B_r$ must touch (or cross) at least once. So $S_{bd-max} \geq S_{ud-max}$.

Figure 2–9: Bi-directional movement under maximum separation constraint

2.1.5 Algorithm Under Maximum Separation Constraint Condition

In this section we present an algorithm that generates an optimal treatment plan under the maximum separation constraint. Recall that Algorithm SINGLEPAIR generates the optimal plan without considering this constraint. We modify Algorithm SINGLEPAIR so that all instances of violation of maximum separation (that may possibly exist) are eliminated. We know that bi-directional leaf profiles do not help eliminate the constraint. So we consider only unidirectional profiles.

Algorithm. The algorithm is described in Figure 2–10.

Theorem 7 Algorithm MAXSEPARATION obtains plans that are optimal in therapy time, under the maximum separation constraint.
Algorithm MAXSEPARATION

1. Apply Algorithm SINGLEPAIR to obtain the optimal plan \((I_l, I_r)\).
2. Find the least value of intensity, \(I_1\), such that the leaf separation in \((I_l, I_r)\) when \(I_1\) MUs are delivered is \(> S_{\text{max}}\), where \(S_{\text{max}}\) is the maximum allowed separation between the leaves. If there is no such \(I_1\), \((I_l, I_r)\) is the optimal plan; end.
3. Let \(x_j\) and \(x_k\), respectively, be the position of the left and right leaves at this time (see Figure 2–11). Relocate the right leaf at \(x'_k\) such that \(x'_k - x_j = S_{\text{max}}\), when \(I_1\) MUs are delivered. Let \(\Delta I = I_l(x_j) - I_1 = I_2 - I_1\). Move the profile of \(I_r\), which follows \(x'_k\), up by \(\Delta I\) along \(I\) direction. To maintain \(I(x) = I_l(x) - I_r(x)\) for every \(x\), move the profile of \(I_l\), which follows \(x'_k\), up by \(\Delta I\) along \(I\) direction. Goto Step 2.

Figure 2–10: Obtaining a plan under maximum separation constraint

![Diagram](image)

Figure 2–11: Maximum separation constraint violation
Proof: We use induction to prove the theorem.

The statement we prove, $S(n)$, is the following:

After Step 3 of the algorithm is applied $n$ times, the resulting plan, $(I_{n}, I_{rn})$, satisfies

(a) It has no maximum separation violation when $I < I_2(n)$ MUs are delivered, where $I_2(n)$ is the value of $I_2$ during the $n$th iteration of Algorithm MAXSEPARATION.

(b) For plans that satisfy (a), $(I_{n}, I_{rn})$ is optimal in therapy time.

1. Consider the base case, $n = 1$.

Let $(I, I_r)$ be the plan generated by Algorithm SINGLEPAIR. After Step 3 is applied once, the resulting plan $(I_{11}, I_{r1})$ meets the requirement that there is no maximum separation violation when $I < I_2(1)$ MUs are delivered by the radiation source. The therapy time increases by $\Delta I$, i.e., $TT(I_{11}, I_{r1}) = TT(I, I_r) + \Delta I$.

Assume that there is another plan, $(I_{11}', I_{r1}')$, which satisfies condition (a) of $S(1)$ and $TT(I_{11}', I_{r1}') < TT(I_{11}, I_{r1})$. We show this assumption leads to a contradiction and so there is no such plan $(I_{11}', I_{r1}')$.

Let $x_j, x_k$ and $x_k'$ be as in Algorithm MAXSEPARATION. We consider three cases for the relationship between $I_{11}'(x_j)$ and $I_{11}(x_j)$.

(a) $I_{11}'(x_j) = I_{11}(x_j) = I_2(1)$

Since there is no maximum separation violation when $I < I_2(1)$ MUs are delivered, $I_{r1}'(x_k') \geq I_{11}'(x_j) = I_{11}(x_k')$. Since $I(x_k') = I_{11}'(x_k') - I_{r1}'(x_k') = I_{11}(x_k') - I_{r1}(x_k')$, we have $I_{11}'(x_k') \geq I_{11}(x_k')$. We now construct a plan $(I_{11}'', I_{r1}'')$ as follows:

$$
I_{11}''(x) = \begin{cases} 
  I_l(x) & 0 \leq x < x_k' \\
  I_{11}'(x) - \Delta I & x \geq x_k' 
\end{cases}
$$

$$
I_{r1}''(x) = \begin{cases} 
  I_r(x) & 0 \leq x < x_k' \\
  I_{r1}'(x) - \Delta I & x \geq x_k' 
\end{cases}
$$

Clearly $I_{11}''(x) - I_{r1}''(x) = I_l(x), 0 \leq x \leq x_m$. Also, $I_{11}''$ is non-decreasing ($I_{11}''(x_k') = I_{11}(x_k') - \Delta I = I_l(x_k') - \Delta I = I_l(x_{k-1}) = I_{11}''(x_{k-1})$). Similarly $I_{r1}''$ is non-decreasing. So $(I_{11}'', I_{r1}'')$ is a plan for $I(x_i)$. \[\]
Also, $TT(I''_{l1}, I''_{r1}) = TT(I'_{l1}, I'_{r1}) - \Delta I < TT(I_{l1}, I_{r1}) - \Delta I = TT(I_{l}, I_{r})$.

This contradicts our knowledge that $(I_l, I_r)$ is the optimal unconstrained plan.

(b) $I'_{l1}(x_j) > I_{l1}(x_j)$

This leads to a contradiction as in the previous case.

(c) $I'_{l1}(x_j) < I_{l1}(x_j)$

In this case, $I'_{l1}(x_j) < I_{l1}(x_j) = I_{l}(x_j)$. This violates Corollary 1. So this case cannot arise.

Therefore $S(1)$ is true.

2. Induction step

Assume $S(n)$ is true. If there are no more maximum separation violations in the resulting plan, $(I_{ln}, I_{rn})$, then it is the optimal plan. If there are more violations, we find the next violation. Apply Step 3 of the algorithm to get a new plan. Assume that there is another plan, which costs less time than the plan generated by Algorithm MAXSEPARATION. We consider three cases as in the base case and show by contradiction that there is no such plan. Therefore $S(n+1)$ is true whenever $S(n)$ is true.

Since the number of iterations of Steps 2 and 3 of the algorithm is finite (at most one iteration can occur when the left leaf is at $x_i, 0 \leq i \leq m$), all maximum separation violations will eventually be eliminated.

Note that the minimum leaf separation of the plan constructed by Algorithm MAXSEPARATION is $\min\{S_{ud-min(opt)}, S_{max}\}$. From Theorem 7, it follows that Algorithm MAXSEPARATION constructs an optimal plan that satisfies both the minimum and maximum separation constraints provided that $S_{ud-min(opt)} \geq S_{min}$. Note that when $S_{ud-min(opt)} < S_{min}$, there is no plan that satisfies the minimum separation constraint.

2.1.6 Algorithm Under Inter-Pair Minimum Separation Constraint

Introduction. We use a single pair of leaves to deliver intensity profiles defined along the axis of the pair of leaves. However, in a real application, we need to deliver intensity profiles defined over a 2-D region. Each pair of leaves is controlled independently. If there are no constraints on the leaf movements, we divide the desired profile into a set of parallel
profiles defined along the axes of the leaf pairs. Each leaf pair $i$ then delivers the plan for the corresponding intensity profile $I_i(x)$. The set of plans of all leaf pairs forms the solution set. We refer to this set as the treatment schedule (or simply schedule). In this section, we present leaf sequencing algorithms for SMLC with and without constraints. The constraints considered are (i) minimum separation constraint and (ii) tongue-and-groove constraint and (optionally) interdigitation constraint.

We use the term intra-pair minimum separation constraint to refer to the constraint imposed on an opposing pair of leaves and inter-pair minimum separation constraint to refer to the constraint imposed on opposing leaves of neighboring pairs. Recall that, in Section 2.1.3, we proved that for a single pair of leaves, if the optimal plan does not satisfy the minimum separation constraint, then no plan satisfies the constraint. In this section we present an algorithm to generate the optimal schedule for the desired profile defined over a 2-D region. We then modify the algorithm to generate schedules that satisfy the inter-pair minimum separation constraint.

**Optimal schedule without the minimum separation constraint.** Assume we have $n$ pairs of leaves. For each pair, we have $m$ sample points. The input is represented as a matrix with $n$ rows and $m$ columns, where the $i$th row represents the desired intensity profile to be delivered by the $i$th pair of leaves. We apply Algorithm SINGLEPAIR to determine the optimal plan for each of the $n$ leaf pairs. This method of generating schedules is described in Algorithm MULTIPAIR (Figure 2–12).

**Algorithm MULTIPAIR**

```
For (i = 1; i <= n; i++)
    Apply Algorithm SINGLEPAIR to the $i$th pair of leaves to obtain plan $(I_{il}, I_{ir})$ that delivers the intensity profile $I_i(x)$.
End For
```

**Figure 2–12: Obtaining a schedule**

**Lemma 4** Algorithm MULTIPAIR generates schedules that are optimal in therapy time.

**Proof:** Treatment is completed when all leaf pairs (which are independent) deliver their respective plans. The therapy time of the schedule generated by Algorithm MULTIPAIR is $\max\{TT(I_{1l}, I_{1r}), TT(I_{2l}, I_{2r}), \ldots, TT(I_{nl}, I_{nr})\}$. From Theorem 1, it follows that this therapy time is optimal.
Optimal algorithm with inter-pair minimum separation constraint. The schedule generated by Algorithm MULTIPAIR may violate both the intra- and inter-pair minimum separation constraints. If the schedule has no violations of these constraints, it is the desired optimal schedule. If there is a violation of the intra-pair constraint, then it follows from Theorem 2 that there is no schedule that is free of constraint violation. So, assume that only the inter-pair constraint is violated. We eliminate all violations of the inter-pair constraint starting from the left end, i.e., from \( x_0 \). To eliminate the violations, we modify those plans of the schedule that cause the violations. We scan the schedule from \( x_0 \) along the positive \( x \) direction looking for the least \( x_v \) at which is positioned a right leaf (say \( Ru \)) that violates the inter-pair separation constraint. After rectifying the violation at \( x_v \) with respect to \( Ru \) we look for other violations. Since the process of eliminating a violation at \( x_v \), may at times, lead to new violations at \( x_j, x_j < x_v \), we need to retract a certain distance (we will show that this distance is \( S_{\text{min}} \)) to the left, every time a modification is made to the schedule. We now restart the scanning and modification process from the new position. The process continues until no inter-pair violations exist. Algorithm MINSEPARATION (Figure 2–13) outlines the procedure.

Algorithm MINSEPARATION

1. \( x = x_0 \)
2. While (there is an inter-pair violation) do
3. Find the least \( x_v, x_v \geq x \), such that a right leaf is positioned at \( x_v \) and this right leaf has an inter-pair separation violation with one or both of its neighboring left leaves. Let \( u \) be the least integer such that the right leaf \( Ru \) is positioned at \( x_v \) and \( Ru \) has an inter-pair separation violation. Let \( Lt \) denote the left leaf (or one of the left leaves) with which \( Ru \) has an inter-pair violation. Note that \( t \in \{u - 1, u + 1\} \).
4. Modify the schedule to eliminate the violation between \( Ru \) and \( Lt \).
5. If there is now an intra-pair separation violation between \( Rt \) and \( Lt \), no feasible schedule exists, terminate.
6. \( x = x_v - S_{\text{min}} \)
7. End While

Figure 2–13: Obtaining a schedule under the constraint

Let \( M = ((I_{1l}, I_{1r}), (I_{2l}, I_{2r}), \ldots, (I_{nt}, I_{nr})) \) be the schedule generated by Algorithm MULTIPAIR for the desired intensity profile.

Let \( N(p) = ((I_{1lp}, I_{1rp}), (I_{2lp}, I_{2rp}), \ldots, (I_{nlp}, I_{nrp})) \) be the schedule obtained after Step 4
of Algorithm MINSEPARATION is applied $p$ times to the input schedule $M$. Note that $M = N(0)$.

To illustrate the modification process we use an example (see Figure 2–14). To make things easier, we only show two neighboring pairs of leaves. Suppose that the $(p + 1)$th violation occurs when the right leaf of pair $u$ is positioned at $x_v$ and the left leaf of pair $t$, $t \in \{u-1, u+1\}$, arrives at $x_u, x_v - x_u < S_{\text{min}}$. Let $x'_u = x_v - S_{\text{min}}$. To remove this inter-pair separation violation, we modify $(I_{tlp}, I_{trp})$. The other profiles of $N(p)$ are not modified. The new $I_{tlp}$ (i.e., $I_{tl(p+1)}$) is as defined below.

$$I_{tl(p+1)}(x) = \begin{cases} 
I_{tlp}(x) & x_0 \leq x < x'_u \\
\max\{I_{tlp}(x), I_{tl}(x) + \Delta I\} & x'_u \leq x \leq x_m
\end{cases}$$

where $\Delta I = I_{arp}(x_v) - I_{tl}(x'_u) = I_2 - I_1$. $I_{tr(p+1)}(x) = I_{tl(p+1)}(x) - I_t(x)$, where $I_t(x)$ is the target profile to be delivered by the leaf pair $t$. Since $I_{tr(p+1)}$ differs from $I_{trp}$ for $x \geq x'_u = x_v - S_{\text{min}}$ there is a possibility that $N(p+1)$ has inter-pair separation violations for right leaf positions $x \geq x'_u = x_v - S_{\text{min}}$. Since none of the other right leaf profiles are changed from those of $N(p)$ and since the change in $I_{tl}$ only delays the rightward movement of the left leaf of pair $t$, no inter-pair violations are possible in $N(p+1)$ for $x < x'_u = x_v - S_{\text{min}}$.
One may also verify that since $I_{tl0}$ and $I_{tr0}$ are non-decreasing functions of $x$, so also are $I_{tlp}$ and $I_{trp}$, $p > 0$.

**Lemma 5** Let $F = ((I'_{tl1}, I'_{lr1}), (I'_{tl2}, I'_{lr2}), \ldots, (I'_{tnl}, I'_{tnr}))$ be any feasible schedule for the desired profile, i.e., a schedule that does not violate the intra- or inter-pair minimum separation constraints. Let $S(p)$, be the following assertions.

(a) $I'_{tl}(x) \geq I_{tlp}(x)$, $0 \leq i \leq n$, $x_0 \leq x \leq x_m$

(b) $I'_{lr}(x) \geq I_{lrp}(x)$, $0 \leq i \leq n$, $x_0 \leq x \leq x_m$

$S(p)$ is true for $p \geq 0$.

**Proof:** The proof is by induction on $p$.

1. Consider the base case, $p = 0$. From Corollary 1 and the fact that the plans $(I_{tl0}, I_{lr0}), 0 \leq i \leq n$, are generated using Algorithm SINGLEPAIR, it follows that $S(0)$ is true.

2. Assume $S(p)$ is true. Suppose Algorithm MINSEPARATION finds a next violation and modifies the schedule $N(p)$ to $N(p + 1)$. Suppose that the next violation occurs when the right leaf of pair $u$ is positioned at $x_v$ and the left leaf of pair $t$ arrives at $x_u$, $x_v - x_u < S_{\text{min}}$ (see Figure 2-14). Let $x'_u = x_v - S_{\text{min}}$. We modify pair $t$’s plan for $x'_u \leq x \leq x_m$, to eliminate the violation. All other plans in the schedule remain unaltered. Therefore, to establish $S(p + 1)$ it suffices to prove that

$$I'_{tl}(x) \geq I_{tl(p+1)}(x), x'_u \leq x \leq x_m$$  \hspace{1cm} (2.2)

$$I'_{tr}(x) \geq I_{tr(p+1)}(x), x'_u \leq x \leq x_m$$  \hspace{1cm} (2.3)

We need prove only one of these two relationships since $I'_{tl}(x) - I_{tl}(x) = I_{tl(p+1)}(x) - I_{tr(p+1)}(x), x_0 \leq x \leq x_m$. We now consider pair $t$’s plan for $x'_u \leq x \leq x_m$. We analyze three cases, that are exhaustive, and show that Equation 2.2 is true for each. This, in turn, implies that $S(p + 1)$ is true whenever $S(p)$ is true and hence completes the proof.

(a) No modification (relative to $M = N(0)$) has been made to pair $t$’s plan for $x \geq x'_u$ prior to this. In this case, $I_{tlp}(x) = I_{tl0}(x) = I_{tl}(x), x \geq x'_u$. 


The situation is illustrated in Figure 2–14.
Since there is no minimum separation violation in \( F \), the left leaf of pair \( t \) passes \( x'_u \) only after the right leaf of pair \( u \) passes \( x_v \), i.e.,

\[
I'_t(x'_u) \geq I'_{ur}(x_v) \quad (2.4)
\]

Since \( S(p) \) is true,

\[
I'_{ur}(x_v) \geq I_{urp}(x_v) = I_{tl}(p+1)(x'_u) \quad (2.5)
\]

From Equations 2.4 and 2.5,

\[
I'_t(x'_u) \geq I_{tl}(p+1)(x'_u) \quad (2.6)
\]

Adding and subtracting \( I'_t(x'_u) \) to \( I'_t(x) \),

\[
I'_t(x) = I'_t(x'_u) + I'_t(x) - I'_t(x'_u), \quad 0 \leq x \leq x_m \quad (2.7)
\]

Similarly,

\[
I_{tl}(p+1)(x) = I_{tl}(p+1)(x'_u) + I_{tl}(p+1)(x) - I_{tl}(p+1)(x'_u), \quad 0 \leq x \leq x_m \quad (2.8)
\]

Since \( I_{tl}(x) = I_{tl}(x), x \geq x'_u \),

\[
I_{tl}(p+1)(x) = I_{tl}(x) + \Delta I, \quad x'_u \leq x \leq x_m \quad (2.9)
\]

From Equations 2.8 and 2.9, we get

\[
I_{tl}(p+1)(x) = I_{tl}(p+1)(x'_u) + (I_t(x) + \Delta I) - (I_t(x'_u) + \Delta I), \quad x'_u \leq x \leq x_m
\]

\[
= I_{tl}(p+1)(x'_u) + I_{tl}(x) - I_{tl}(x'_u), \quad x'_u \leq x \leq x_m \quad (2.10)
\]

Subtracting Equation 2.10 from Equation 2.7,

\[
I'_t(x) - I_{tl}(p+1)(x) = (I'_t(x'_u) - I_{tl}(p+1)(x'_u)) + (I_t(x) - I_t(x)) - (I'_t(x'_u) - I_{tl}(x'_u)), \quad x'_u \leq x \leq x_m \quad (2.11)
\]
From Equations 2.6 and 2.11,

\[
I'_{tl}(x) - I_{tl}(p+1)(x) \geq (I'_{tl}(x) - I_{tl}(x)) - (I'_{tl}(x') - I_{tl}(x'))
\]

\[
x'_{u} \leq x \leq x_{m}
\]

(2.12)

From Lemma 2b,

\[
I'_{tl}(x) - I_{tl}(x) \geq I'_{tl}(x'_{u}) - I_{tl}(x'_{u}), x'_{u} \leq x \leq x_{m}
\]

(2.13)

From Equations 2.12 and 2.13, we get

\[
I'_{tl}(x) \geq I_{tl}(p+1)(x), x'_{u} \leq x \leq x_{m}
\]

(2.14)

(b) Some prior modification has been made to pair t’s plan for \(x \geq x'_{u}\). There exists a modification at \(x_{w}\) such that \(I_{tlp}(x) > I_{tl}(x) + \Delta I, x_{w} \leq x \leq x_{m}\), and there is no \(x < x_{w}\) that satisfies this condition. Note that \(I_{tlp}(x'_{u}) \leq \) amount of MUs delivered when profile \(I_{tlp}(x)\) arrives at \(x_{u}\) (since \(I_{tlp}(x)\) is a non-decreasing function of \(x\) < \(I_{urp}(x)\) (since there is a minimum separation violation when profile \(I_{urp}(x)\) is at \(x)\). Therefore, \(I_{tlp}(x'_{u}) < I_{tl}(x'_{u}) + I_{urp}(x_{v}) - I_{tl}(x'_{u}) = I_{tl}(x'_{u}) + \Delta I\). So, \(x_{w} > x'_{u}\).

In this case (see Figure 2–15),

\[
I_{tl}(p+1)(x) = \begin{cases} 
I_{tl}(x) + \Delta I & x'_{u} \leq x < x_{w} \\
I_{tlp}(x) & x_{w} \leq x \leq x_{m}
\end{cases}
\]

Note that, in the example of Figure 2–15, a prior modification was made to pair t’s plan for \(x \geq x_{q}\). However, \(I_{tlp}(x) < I_{tl}(x) + \Delta I, x_{q} \leq x < x_{w}\).

We get \(I'_{tl}(x) \geq I_{tl}(p+1)(x), x'_{u} \leq x < x_{w}\), for reasons similar to those in the previous case. Also, \(I'_{tl}(x) \geq I_{tl}(p+1)(x) = I_{tlp}(x), x_{w} \leq x \leq x_{m}\), since \(S(p)\) is true. It follows that \(I'_{tl}(x) \geq I_{tl}(p+1)(x), x'_{u} \leq x \leq x_{m}\).

(c) Some prior modification has been made to pair t’s plan for \(x \geq x'_{u}\). However, \(I_{tlp}(x) \leq I_{tl}(x) + \Delta I, x'_{u} \leq x \leq x_{m}\).

In this case, \(I_{tl}(p+1)(x) = I_{tl}(x) + \Delta I, x'_{u} \leq x \leq x_{m}\). This is similar to the first case.
Lemma 6 If an intra-pair minimum separation violation is detected in Step 5 of MINSEPARATION, then there is no feasible schedule for the desired profile.

Proof: Suppose that there is a feasible schedule $F$ and that leaf pair $t$ has an intra-pair minimum separation violation in $N(p), p > 0$. From Lemma 5 it follows that

(a) $I'_t(x) \geq I_{tp}(x), x_0 \leq x \leq x_m$

(b) $I'_r(x) \geq I_{rp}(x), x_0 \leq x \leq x_m$

where $I'$ and $I$ are as in Lemma 5. However, from the proof of Theorem 2 it follows that if $I_{tp}$ and $I_{rp}$ have a minimum separation violation, then no treatment plan $(I'_t, I'_r)$ that satisfies (a) and (b) can be feasible. Therefore, no feasible schedule $F$ exists.

Example 1 We illustrate an instance where an inter-pair minimum separation violation is detected in Step 5 of MINSEPARATION. Figure 2–16 shows two intensity profiles, to be delivered by adjacent leaf pairs (say $t$ and $t + 1$). The plans for $I_t(x)$ and $I_{t+1}(x)$ are obtained using algorithm MULTIPAIR. They are shown in Figure 2–17. Each of these plans ($I_t(x), I_r(x)$) and ($I_{(t+1)t}(x), I_{(t+1)r}(x)$) is feasible, i.e., there is no intra-pair minimum separation ($S_{min} = 7$). However, when MINSEPARATION is applied (for simplicity consider leaf pairs $t$ and $t + 1$ in isolation), it detects an inter-pair minimum separation violation between $I_{(t+1)t}$ and $I_{tr}$, when $I_{(t+1)t}$ arrives at $x = 6$ and $I_{tr}$ is positioned at $x = 11$. 
To eliminate this violation, $I_{(t+1)l}$ is positioned at $x = 4$ (since $11 - 4 = 7 = S_{\text{min}}$) and its profile is raised from $x = 4$. Consequently $I_{(t+1)r}$ is also raised from $x = 4$ resulting in the plan $(I_{(t+1)l}(x), I_{(t+1)r}(x))$. This modification results in an intra-pair violation for pair $t + 1$, when $I_{(t+1)l}$ is at $x = 1$ and $I_{(t+1)r}$ is at $x = 4$. From Lemma 6, there is no feasible schedule.

Figure 2–16: Intensity profiles of adjacent leaf pairs

For $N(p), p \geq 0$ and every leaf pair $j, 1 \leq j \leq n$, define $I_{jlp}(x_{i-1}) = I_{jrp}(x_{i-1}) = 0, \Delta_{jlp}(x_i) = I_{lp}(x_i) - I_{lp}(x_{i-1}), 0 \leq i \leq m$ and $\Delta_{jrp}(x_i) = I_{rp}(x_i) - I_{rp}(x_{i-1}), 0 \leq i \leq m$. Notice that $\Delta_{jlp}(x_i)$ gives the time (in monitor units) for which the left leaf of pair $j$ stops at position $x_i$. Let $\Delta_{jlp}(x_i)$ and $\Delta_{jrp}(x_i)$ be zero for all $x_i$ when $j = 0$ as well as when $j = n + 1$.

**Lemma 7** For every $j, 1 \leq j \leq n$ and every $i, 1 \leq i \leq m$,

$$\Delta_{jlp}(x_i) \leq \max \{\Delta_{j0lp}(x_i), \Delta_{(j-1)lp}(x_i + S_{\text{min}}), \Delta_{(j+1)lp}(x_i + S_{\text{min}})\}$$ (2.15)
Figure 2-17: Profiles violating inter-pair constraint

Proof: The proof is by induction on $p$. For the induction base, $p = 0$. Putting $p = 0$ into the right side of Equation 2.15, we get

$$\max\{\Delta_{j0}(x_i), \Delta_{(j-1)r0}(x_i + S_{min}), \Delta_{(j+1)r0}(x_i + S_{min})\} \geq \Delta_{j0}(x_i) \quad (2.16)$$

For the induction hypothesis, let $q \geq 0$ be any integer and assume that Equation 2.15 holds when $p = q$. In the induction step, we prove that the equation holds when $p = q + 1$. Let $t, u$, and $x_v$ be as in iteration $p - 1$ of the while loop of algorithm MINSEPARATION. Following this iteration, only $\Delta_{tlp}$ and $\Delta_{trp}$ are different from $\Delta_{tl(p-1)}$ and $\Delta_{tr(p-1)}$, respectively. Furthermore, only $\Delta_{tlp}(x_w)$ and $\Delta_{trp}(x_w)$, where $x_w = x_v - S_{min}$ may be larger than the corresponding values following iteration $p - 1$. At all but at most one other $x$ value (where $\Delta$ may have decreased), $\Delta_{tlp}$ and $\Delta_{trp}$ are the same as the corresponding values following iteration $p - 1$.

Since $x_v$ is the right leaf position for the leftmost violation, the left leaf of pair $t$ arrives at $x_w = x_v - S_{min}$ after the right leaf of pair $u$ arrives at $x_v = x_w + S_{min}$. Following the modification made to $I_{tl(p-1)}$, the left leaf of pair $t$ leaves $x_w$ at the same time as the right leaf of pair $u$ leaves $x_w + S_{min}$. Therefore, $\Delta_{tp}(x_w) \leq \Delta_{ur(p-1)}(x_w + S_{min}) = \Delta_{urp}(x_w + S_{min})$.

The induction step now follows from the induction hypothesis and the observation that $u \in \{t - 1, t + 1\}$.\[\square\]
Lemma 8 For every \( j, 1 \leq j \leq n \) and every \( i, 1 \leq i \leq m \),

\[
\Delta_{jrp}(x_i) = \Delta_{jlp}(x_i) - (I_j(x_i) - I_j(x_{i-1})
\] (2.17)

where \( I_j(x_{-1}) = 0 \).

**Proof:** We examine \( N(p) \). The monitor units delivered by leaf pair \( j \) at \( x_i \) are \( I_{jlp}(x_i) - I_{jrp}(x_i) \) and the units delivered at \( x_{i-1} \) are \( I_{jlp}(x_{i-1}) - I_{jrp}(x_{i-1}) \). Therefore,

\[
I_j(x_i) = I_{jlp}(x_i) - I_{jrp}(x_i)
\] (2.18)

\[
I_j(x_{i-1}) = I_{jlp}(x_{i-1}) - I_{jrp}(x_{i-1})
\] (2.19)

Subtracting Equation 2.19 from Equation 2.18, we get

\[
I_j(x_i) - I_j(x_{i-1}) = (I_{jlp}(x_i) - I_{jlp}(x_{i-1})) - (I_{jrp}(x_i) - I_{jrp}(x_{i-1}))
\]

\[
= \Delta_{jlp}(x_i) - \Delta_{jrp}(x_i)
\] (2.20)

The lemma follows from this equality.

Notice that once a right leaf \( u \) moves past \( x_m \), no separation violation with respect to this leaf is possible. Therefore, \( x_v \) (see algorithm MINSEPARATION) \( \leq x_m \). Hence, \( \Delta_{jlp}(x_i) \leq \Delta_{jlp}(x_{i-1}) \), and \( \Delta_{jrp}(x_i) \leq \Delta_{jrp}(x_{i-1}) \), \( x_m - S_{min} \leq x_i \leq x_m \), \( 1 \leq j \leq n \). Starting with these upper bounds, which are independent of \( p \), on \( \Delta_{jrp}(x_i) \), \( x_m - S_{min} \leq x_i \leq x_m \) and using Equations 2.15 and 2.17, we can compute an upper bound on the remaining \( \Delta_{jlp}(x_i) \)s and \( \Delta_{jrp}(x_i) \)s (from right to left). The remaining upper bounds are also independent of \( p \).

Let the computed upper bound on \( \Delta_{jlp}(x_i) \) be \( U_{jl}(x_i) \). It follows that the therapy time for \( (I_{jlp}, I_{jrp}) \) is at most \( T_{max}(j) = \sum_{0 \leq i \leq m} U_{jl}(x_i) \). Therefore, the therapy time for \( N(p) \) is at most \( T_{max} = \max_{1 \leq j \leq n} \{T_{max}(j)\} \).

**Theorem 8** The following are true of Algorithm MINSEPARATION:

(a) The algorithm terminates.

(b) When the algorithm terminates in Step 5, there is no feasible schedule.

(c) Otherwise, the schedule generated is feasible and is optimal in therapy time for unidirectional schedules.
Proof: (a) As noted above, Lemmas 7 and 8 provide an upper bound, $T_{max}$ on the therapy time of any schedule produced by algorithm MINSEPARATION. It is easy to verify that

$$I_{il(p+1)}(x) \geq I_{ilp}(x), 0 \leq i \leq n, x_0 \leq x \leq x_m$$
$$I_{ir(p+1)}(x) \geq I_{irp}(x), 0 \leq i \leq n, x_0 \leq x \leq x_m$$

and that

$$I_{tl(p+1)}(x_u') > I_{tlp}(x_u')$$
$$I_{tr(p+1)}(x_u') > I_{trp}(x_u')$$

Notice that even though a $\Delta$ value (proof of Lemma 7) may decrease at an $x_i$, the $I_{ilp}$ and $I_{irp}$ values never decrease at any $x_i$ as we go from one iteration of the while loop of MINSEPARATION to the next. Since $I_{tl}$ increases by atleast one unit at atleast one $x_i$ on each iteration, it follows that the while loop can be iterated at most $mnT_{max}$ times.

(b) Follows from Lemma 6.

(c) If termination does not occur in Step 5, then no minimum separation violations remain and the final schedule is feasible. From Lemma 5, it follows that the final schedule is optimal in therapy time for unidirectional schedules.

Corollary 3 When $S_{min} = 0$, Algorithm Minseparation always generates an optimal feasible schedule.

Proof: When $S_{min} = 0$, Algorithm Minseparation cannot terminate in Step 5 because the Step 4 modification never causes the left leaf of a leaf pair to cross the right leaf of that pair. The Corollary follows now from Theorem 8.

2.2 Conclusion

In conclusion, we presented mathematical formalisms and rigorous proofs of leaf sequencing algorithms for segmental multileaf collimation which maximize MU efficiency. These leaf sequencing algorithms explicitly account for minimum leaf separation constraint and leaf interdigitation constraint. We have shown that our algorithms obtain all feasible
solutions that are optimal in treatment MUs. Furthermore, our analysis shows that unidirectional leaf movement is at least as efficient as bi-directional movement. Thus these algorithms are well suited for common use in SMLC beam delivery.
CHAPTER 3
SEQUENCING OF DYNAMIC MULTILEAF COLLIMATORS

Delivery using DMLC is different from that using SMLC. The leaf positions change with respect to time. In terms of the MLC controller it is the change in position with respect to monitor units delivered that is important. The inputs required are the leaf positions at various control points, the fractional number of monitor units to be delivered at each control point, and the total number of monitor units to be delivered for that beam. In this chapter, we present a systematic study of the optimization of leaf sequencing algorithms for the dynamic beam delivery and provide rigorous proofs of optimized leaf sequence settings in terms of MU efficiency under various leaf movement constraints. Practical leaf movement constraints that are considered include the leaf interdigitation constraint. The question of whether bi-directional leaf movement will increase the MU efficiency when compared with unidirectional leaf movement only is also addressed.

3.1 Methods

3.1.1 Movement of Leaves

In our analysis we will assume that \( I(x_0) > 0 \) and \( I(x_m) > 0 \) and that when the beam delivery begins the leaves can be positioned anywhere. We also assume that the leaves can move with any velocity \( v, -v_{\text{max}} \leq v \leq v_{\text{max}} \), where \( v_{\text{max}} \) is the maximum allowable velocity of the leaves and that the leaf acceleration can be infinite. The consequences of assuming infinite leaf acceleration are negligible. Figure 3–1 illustrates the leaf trajectory during DMLC delivery. In the example, the leaves move from left to right. Let \( I_l(x_i) \) and \( I_r(x_i) \), respectively, denote the amount of Monitor Units (MUs) delivered when the left and right leaves leave position \( x_i \). Consider the motion of the left leaf. The left leaf begins at \( x_0 \) and remains here until \( I_l(x_0) \) MUs have been delivered. At this time the left leaf leaves \( x_0 \) and is moved to \( x_1 \), where it remains until \( I_l(x_1) \) MUs have been delivered. The left leaf then moves to \( x_3 \) where it remains until \( I_l(x_3) \) MUs have been delivered. At this time, the left leaf is moved to \( x_5 \), where it remains until \( I_l(x_5) \) MUs have been delivered. Then it
moves to $x_6$, where it remains until $I_l(x_6)$ MUs have been delivered. The final movement of the left leaf is to $x_{10}$. The left leaf arrives at $x_{10}$ when $I_{\text{max}}$ MUs have been delivered. At this time the machine is turned off. The total therapy time, $TT(I_l, I_r)$, is the time needed to deliver $I_{\text{max}}$ MUs. The right leaf starts at $x_0$ and begins to move rightaway till it reaches $x_2$; leaves $x_2$ when $I_r(x_2)$ MUs have been delivered; leaves $x_4$ when $I_r(x_4)$ MUs have been delivered, and so on. Note that the machine is on throughout the treatment. All MUs that are delivered along a radiation beam along $x_i$ before the left leaf passes $x_i$ fall on it. Similarly, all MUs that are delivered along a radiation beam along $x_i$ before the right leaf passes $x_i$, are blocked by the leaf. So the net amount of MUs delivered at a point is given by $I_l(x_i) - I_r(x_i)$, which must be the same as the desired profile $I(x_i)$.

![Figure 3–1: Leaf trajectory during DMLC delivery](image)

**Theorem 9** The following are true of every leaf pair trajectory that delivers a discrete profile:

(a) The left leaf must reach $x_0$ at some time.

(b) The right leaf must reach $x_m$ at some time.

(c) The left leaf must reach $x_m$ at some time.

(d) The right leaf must reach $x_0$ at some time.

**Proof:** (a) Suppose that the left leaf always stays to the right of $x_0$, then $x_0$ does not receive any MUs, contradicting our assumption that $I(x_0) > 0$. 
(b) Similar to that of (a).

(c) If the left leaf doesn’t reach $x_m$ (i.e., it doesn’t go to the right of $x_{m-1}$), from (b), it follows that the region between $x_{m-1}$ and $x_m$ receives a non-uniform distribution of MUs. Hence the discrete profile can’t be accurately delivered.

(d) Similar to that of (c).

3.1.2 Maximum Velocity Constraint

As noted earlier, the velocity of leaves cannot exceed some maximum limit (say $v_{max}$) in practice. This implies that the leaf profile cannot be horizontal at any point. From Figure 2–3, observe that the time needed for a leaf to move from $x_i$ to $x_{i+1}$ is $\geq (x_{i+1} - x_i)/v_{max}$. If $\Phi$ is the flux density of MUs from the source, the number of MUs delivered in this time along a beam is $\geq \Phi*(x_{i+1}-x_i)/v_{max}$. So, $I_l(x_{i+1}) - I_l(x_i) \geq \Phi*(x_{i+1}-x_i)/v_{max} = \Phi*\Delta x/v_{max}$. The same is true for the right leaf profile $I_r$.

3.1.3 Optimal Unidirectional Algorithm for one Pair of Leaves

Unidirectional movement. When the movement of leaves is restricted to only one direction, both the left and right leaves move along the positive $x$ direction, from left to right (Figure 2–1). Once the desired intensity profile, $I(x_i)$ is known, our problem becomes that of determining the individual intensity profiles to be delivered by the left and right leaves, $I_l$ and $I_r$ such that

$$I(x_i) = I_l(x_i) - I_r(x_i), \quad 0 \leq i \leq m$$

(3.1)

Of course, $I_l$ and $I_r$ are subject to the maximum velocity constraint. We refer to $(I_l, I_r)$ as the treatment plan (or simply plan) for $I$. Once we obtain the plan, we will be able to determine the movement of both left and right leaves during the therapy. For each $i$, the left leaf can be allowed to pass $x_i$ when the source has delivered $I_l(x_i)$ MUs. Also, we can allow the right leaf to pass $x_i$ when the source has delivered $I_r(x_i)$ MUs. In this manner we obtain unidirectional leaf movement profiles for a plan. Some simple observations about the leaf profiles are made below.

Theorem 10 In every unidirectional plan the leaves begin at $x_0$ and end at $x_m$.

Proof: Follows from Theorem 9 and the unidirectional constraint.
Lemma 9 In the region between each pair of successive sample points, say \( x_i \) and \( x_{i+1} \), both leaf profiles maintain the same shape, i.e., one is merely a vertical translation of the other.

Proof: As explained previously, the input profile is discretized to a square wave \( I \). Since the profile of \( I \) is horizontal between successive sample points and since it is equal to \( I_l - I_r \), \( I_l \) and \( I_r \) must have the same shape. For example, if the left leaf moves at a constant velocity \( v \) between points \( x_i \) and \( x_{i+1} \), so should the right leaf.

Lemma 10 In an optimal plan, both leaves must move at \( v_{\text{max}} \) between every successive pair of sample points they move across.

Proof: Suppose that in an optimal solution the leaves move between points \( x_i \) and \( x_{i+1} \) at a possibly varying velocity \( v(x) \leq v_{\text{max}} \). From Lemma 9, we know that both leaf profiles are the same between \( x_i \) and \( x_{i+1} \). Setting \( v(x) = v_{\text{max}} \) results in new leaf profiles whose difference remains the same as before (which is the desired profile \( I \)) and total therapy time is lowered. This leads to a contradiction.

Corollary 4 In an optimal plan, no leaf stops at an \( x \) that is not one of the \( x_i \)'s.

Algorithm. From Equation 3.1, we see that one way to determine \( I_l \) and \( I_r \) from the given target profile \( I \) is to begin from \( x_0 \); set \( I_l(x_0) = I(x_0) \) and \( I_r(x_0) = 0 \); examine the remaining \( x_i \)'s to the right; increase \( I_l \) at \( x_i \) whenever \( I \) increases and by the same amount (in addition to the minimum increase imposed by the maximum velocity constraint); and similarly increase \( I_r \) whenever \( I \) decreases. This can be done till we reach \( x_m \). So the treatment begins with the leaves positioned at the leftmost sample point and ends with the leaves positioned at the rightmost sample point. Once \( I_l \) and \( I_r \) are determined the leaf movement profiles are obtained as explained earlier. Note that we move the leaves at the maximum velocity \( v_{\text{max}} \) whenever they are to be moved. The resulting algorithm is shown in Figure 3–2. Figure 2–3 shows a profile \( I \) and the corresponding plan \( (I_l, I_r) \) obtained using Algorithm DMLC-SINGLEPAIR.

Ma et al. (1998) shows that Algorithm DMLC-SINGLEPAIR obtains plans that are optimal in therapy time. Their proof relies on the results of Spirou and Chui (1994), Stein et al. (1994) and Boyer and Strait (1997). We provide a simpler and direct proof below.
Algorithm DMLC-SINGLEPAIR

\[ I_l(x_0) = I(x_0) \]
\[ I_r(x_0) = 0 \]

For \( j = 1 \) to \( m \) do

If \( I(x_j) \geq I(x_{j-1}) \)

\[ I_l(x_j) = I_l(x_{j-1}) + I(x_j) - I(x_{j-1}) + \Phi \ast \Delta x/v_{\max} \]
\[ I_r(x_j) = I_r(x_{j-1}) + \Phi \ast \Delta x/v_{\max} \]

Else

\[ I_r(x_j) = I_r(x_{j-1}) + I(x_{j-1}) - I(x_j) + \Phi \ast \Delta x/v_{\max} \]
\[ I_l(x_j) = I_l(x_{j-1}) + \Phi \ast \Delta x/v_{\max} \]

End for

Figure 3-2: Obtaining a unidirectional plan

**Theorem 11** Algorithm DMLC-SINGLEPAIR obtains plans that are optimal in therapy time.

**Proof:** Let \( I(x_i) \) be the desired profile. Let \( 0 = \text{inc}0 < \text{inc}1 < \ldots < \text{inc}k \) be the indices of the points at which \( I(x_i) \) increases. So \( x_{\text{inc}0}, x_{\text{inc}1}, \ldots, x_{\text{inc}k} \) are the points at which \( I(x) \) increases (i.e., \( I(x_{\text{inc}i}) > I(x_{\text{inc}i-1}) \), assume that \( I(x_{-1}) = 0 \)). Let \( \Delta i = I(x_{\text{inc}i}) - I(x_{\text{inc}i-1}) \), \( i \geq 0 \).

Suppose that \( (I_L, I_R) \) is a plan for \( I(x_i) \) (not necessarily the plan generated by Algorithm DMLC-SINGLEPAIR). Since \( I(x_i) = I_L(x_i) - I_R(x_i) \) for all \( i \), we get

\[ \Delta i = (I_L(x_{\text{inc}i}) - I_R(x_{\text{inc}i})) - (I_L(x_{\text{inc}i-1}) - I_R(x_{\text{inc}i-1})) \]
\[ = (I_L(x_{\text{inc}i}) - I_L(x_{\text{inc}i-1})) - (I_R(x_{\text{inc}i}) - I_R(x_{\text{inc}i-1})) \]
\[ = (I_L(x_{\text{inc}i}) - I_L(x_{\text{inc}i-1}) - \Phi \ast \Delta x/v_{\max}) - (I_R(x_{\text{inc}i}) - I_R(x_{\text{inc}i-1}) - \Phi \ast \Delta x/v_{\max}) \]

Note that from the maximum velocity constraint \( I_R(x_{\text{inc}i}) - I_R(x_{\text{inc}i-1}) \geq \Phi \ast \Delta x/v_{\max}, i \geq 1 \). So \( I_R(x_{\text{inc}i}) - I_R(x_{\text{inc}i-1}) - \Phi \ast \Delta x/v_{\max} \geq 0, i \geq 1 \), and \( \Delta i \leq I_L(x_{\text{inc}i}) - I_L(x_{\text{inc}i-1}) - \Phi \ast \Delta x/v_{\max} \). Also, \( \Delta 0 = I(x_0) - I(x_{-1}) = I(x_0) \leq I_L(x_0) - I_L(x_{-1}) \), where \( I_L(x_{-1}) = 0 \).

Summing up \( \Delta i \), we get

\[ \sum_{i=0}^{k}[I(x_{\text{inc}i}) - I(x_{\text{inc}i-1})] \leq \sum_{i=0}^{k}[I_L(x_{\text{inc}i}) - I_L(x_{\text{inc}i-1})] - k \ast \Phi \ast \Delta x/v_{\max} \].

Let \( S_1 = \sum_{i=0}^{k}[I_L(x_{\text{inc}i}) - I_L(x_{\text{inc}i-1})] \). Then, \( S_1 \geq \sum_{i=0}^{k}[I(x_{\text{inc}i}) - I(x_{\text{inc}i-1})] + k \ast \Phi \ast \Delta x/v_{\max} \). Let \( S_2 = \sum[I_L(x_j) - I_L(x_{j-1})] \), where the summation is carried out over indices \( j \) (0 \( \leq j \leq m \)) such that \( I(x_j) \leq I(x_{j-1}) \). There are a total of \( m + 1 \) indices of which \( k + 1 \) do not satisfy this condition. So there are \( m - k \) indices \( j \) at which \( I(x_j) \leq I(x_{j-1}) \). At each of these
Now, consider the following three all encompassing cases.

\[ n < m \].

Further, from Corollary 5, it follows that \( I_L(x_i) \geq I_L(x_i), 0 \leq i \leq m \). Therefore, \( \Delta(x_i) \geq 0, 0 \leq i \leq m \). Hence, the treatment plan \((I_L, I_R)\) generated by DMLC-SINGLEPAIR is optimal in therapy time.

**Corollary 5** Let \( I(x_i) \), \( 0 \leq i \leq m \) be a desired profile. Let \( I_L(x_i) \) and \( I_R(x_i) \), \( 0 \leq i \leq m \) be the left and right leaf profiles generated by Algorithm DMLC-SINGLEPAIR. \( I_L(x_i) \) and \( I_R(x_i) \), \( 0 \leq i \leq m \) define optimal therapy time unidirectional left and right leaf profiles for \( I(x_i) \), \( 0 \leq i \leq m \).

**Proof:** Follows from Theorem 11

In the remainder of Section 3.1, \((I_L, I_R)\) is the optimal treatment plan generated by Algorithm DMLC-SINGLEPAIR for the desired profile \( I \).

**Properties of the optimal treatment plan.** The following observations are made about the optimal treatment plan \((I_L, I_R)\) generated using Algorithm DMLC-SINGLEPAIR.

**Lemma 11** At most one of the leaves stops at each \( x_i \).

**Lemma 12** Let \((I_L, I_R)\) be any treatment plan for \( I \).

(a) \( \Delta(x_i) = I_L(x_i) - I_l(x_i) = I_R(x_i) - I_r(x_i) \geq 0, 0 \leq i \leq m. \)

(b) \( \Delta(x_i) \) is a non-decreasing function.

**Proof:**

(a) Since \( I(x_i) = I_L(x_i) - I_R(x_i) = I_l(x_i) - I_r(x_i) \), \( I_L(x_i) - I_l(x_i) = I_R(x_i) - I_r(x_i) \).

(b) We prove this by contradiction. Suppose that \( \Delta(x_n) > \Delta(x_{n+1}) \) for some \( n, 0 \leq n < m \). Consider the following three all encompassing cases.

Case 1: \( I_l(x_{n+1}) = I_l(x_n) + \Phi \cdot \Delta x/v_{\max} \) (left leaf does not stop at \( x_{n+1} \))

Now, \( I_L(x_n) = I_l(x_n) + \Delta(x_n) > I_l(x_{n+1}) - \Phi \cdot \Delta x/v_{\max} + \Delta(x_{n+1}) = I_L(x_{n+1}) - \Phi \cdot \Delta x/v_{\max} \).

This is not possible because \( I_L(x_{n+1}) \geq I_L(x_n) + \Phi \cdot \Delta x/v_{\max} \) from the maximum velocity constraint.

Case 2: \( I_r(x_{n+1}) = I_r(x_n) + \Phi \cdot \Delta x/v_{\max} \) (right leaf does not stop at \( x_{n+1} \))

Now, \( I_R(x_n) = I_r(x_n) + \Delta(x_n) > I_r(x_{n+1}) - \Phi \cdot \Delta x/v_{\max} + \Delta(x_{n+1}) = I_R(x_{n+1}) - \Phi \cdot \Delta x/v_{\max} \) for all \( n, 0 \leq n < m \).
\( \Delta x/v_{\text{max}} \).

This is not possible because \( I_R(x_{n+1}) \geq I_R(x_n) + \Phi \ast \Delta x/v_{\text{max}} \) from the maximum velocity constraint.

Case 3: \( I_l(x_{n+1}) \neq I_l(x_n) + \Phi \ast \Delta x/v_{\text{max}} \) and \( I_r(x_{n+1}) \neq I_r(x_n) + \Phi \ast \Delta x/v_{\text{max}} \) (both leaves stop at \( x_{n+1} \))

From Lemma 11 it follows that this case cannot arise.

Therefore, \( \Delta(x_i) \) is a non-decreasing function.

\[ \text{Corollary 6} \] Let \( \Lambda_l(x_i) \) (\( \Lambda_r(x_i) \)) and \( \Lambda_L(x_i) \) (\( \Lambda_R(x_i) \)), respectively, denote the amount of time for which the left (right) leaf stops at \( x_i \) in plans \( (I_l, I_r) \) and \( (I_L, I_R) \). Then

(a) \( \Lambda_L(x_i) \geq \Lambda_l(x_i) \).

(b) \( \Lambda_R(x_i) \geq \Lambda_r(x_i) \).

\[ \text{Proof:} \] (a) Suppose that \( \Lambda_L(x_i) < \Lambda_l(x_i) \). We have the following two cases:

Case 1: Both leaves move at the maximum velocity between \( x_{i-1} \) and \( x_i \) in \( (I_L, I_R) \). We get \( \Delta(x_i) < \Delta(x_{i-1}) \) contradicting Lemma 12(b).

Case 2: In \( (I_L, I_R) \), the leaves do not travel uniformly at the maximum velocity between \( x_{i-1} \) and \( x_i \). In this case, transform plan \( (I_L, I_R) \) to a plan \( (I'_L, I'_R) \) as follows. Between \( x_{i-1} \) and \( x_i \) move the leaves at the maximum velocity. The leaves now arrive at \( x_i \) earlier than they did in \( (I_L, I_R) \) by an amount \( \delta_i \). Propagate this difference to the right from \( x_i \) so that \( I'_L(x_j) = I_L(x_j) - \delta_i \) and \( I'_R(x_j) = I_R(x_j) - \delta_i \), \( j \geq i \). Note that this transformation preserves the \( \Lambda \)'s, i.e., \( \Lambda'_L(x_j) = \Lambda_L(x_j) \). Also, the resulting leaf profiles, \( I'_L \) and \( I'_R \), still form a plan for \( I \). Let \( \Delta'(x_j) = I'_L(x_j) - I_l(x_j) = I'_R(x_j) - I_r(x_j) \). Since \( \Lambda'_L(x_i) = \Lambda_L(x_i) < \Lambda_l(x_i) \) and since the leaves move at maximum velocity from \( x_{i-1} \) to \( x_i \) in \( (I_l, I_r) \) and \( (I'_l, I'_r) \), we have \( \Delta'(x_i) < \Delta'(x_{i-1}) \) contradicting Lemma 12(b).

(b) Similar to proof of (a).

3.1.4 Minimum Separation Constraint

Some MLCs have a minimum separation constraint that requires the left and right leaves to maintain a minimum separation \( S_{\text{min}} \) at all times during the treatment. Notice that in the plan generated by Algorithm DMLC-SINGLEPAIR, the two leaves start and end at the same point. So they are in contact at \( x_0 \) and \( x_m \). When \( s_{\text{min}} > 0 \), the minimum separation constraint is violated at \( x_0 \) and \( x_m \). In order to overcome this problem we modify
Algorithm DMLC-SINGLEPAIR to guarantee minimum separation between the leaves in the vicinity of the end points \((x_0 \text{ and } x_m)\). In particular, we allow the left leaf to be initially positioned at a point \(x_0' = x_0 - s_{\text{min}}\) and the right leaf to be finally positioned at \(x_m' = x_m + S_{\text{min}}\). The only changes made relative to Algorithm DMLC-SINGLEPAIR are for the movement of the left leaf from \(x_0'\) to \(x_0\) and for the right leaf from \(x_m\) to \(x_m'\).

We define the movement of the left leaf from \(x_0'\) to \(x_0\) (and a symmetric definition for the right leaf from \(x_m\) to \(x_m'\)) to be such that it maintains a distance of exactly \(S_{\text{min}}\) from the right leaf at all times. Once the left leaf reaches \(x_0\) it follows the trajectory as before. While this modification results in the exact profile being delivered between \(x_0\) and \(x_m\) it also results in some undesirable exposure to the intervals \((x_0', x_0)\) and \((x_m, x_m')\). In the remainder of this section we will consider an exposure of this kind permissible, provided the exact profile is delivered between \(x_0\) and \(x_m\). Note that for most accelerators (Varian, Elekta) undesirable exposure to the intervals \((x_0', x_0)\) and \((x_m, x_m')\) can be avoided by positioning the back-up jaws at \(x_0\) and \(x_m\) respectively. However, a difficulty arises when the number of monitor units delivered at the time the left leaf reaches \(x_0\) in this new plan (call it \((I_1', I_r')\)) is greater than \(I_l(x_0)\). This would prevent us from using the old plan from \(x_0\) to \(x_m\), since the leaf cannot pass \(x_0\) before it arrives there. Observe however, that if the left leaf were to arrive at \(x_0\) any earlier, it would be too close to the right leaf. In the discussion that follows we show that in this and other cases where the original plan violates the constraint, there are no feasible solutions that deliver exactly the desired profile between \(x_0\) and \(x_m\), while permitting exposure outside this region. The modified algorithm, which we call DMLC-MINSINGLEPAIR, is described in Figure 3–3. Note that the therapy time of the plan produced by DMLC-MINSINGLEPAIR is the same as that of the plan produced by DMLC-SINGLEPAIR. Therefore, the modified plan is optimal in therapy time.

**Theorem 12** (a) \(S_{\text{min}} \Phi / v_{\text{max}} > I_l(x_0)\) or (b) If the plan \((I_1', I_r')\) generated by DMLC-MINSINGLEPAIR violates the minimum separation constraint, then there is no plan for \(I\) that does not violate the minimum separation constraint.

**Proof:** Suppose that \((I_1', I_r')\) violates the minimum separation constraint. Assume that the first violation occurs when \(I_1\) MUs have been delivered. Since there is no violation when less than \(I_1\) MUs are delivered and since the leaves are either stationary or move at the maximum
Algorithm DMLC-MINSINGLEPAIR

1. Apply Algorithm DMLC-SINGLEPAIR
2. Modify the profile of the left leaf from $x_0'$ to $x_0$ and the right leaf from $x_m$ to $x_m'$ to maintain a minimum interleaf distance of $S_{\text{min}}$. Call this profile $(I'_l, I'_r)$.
3. If the number of MUs delivered when the left leaf arrives at $x_0$ is greater than $I_l(x_0)$ there is no feasible solution. End.
4. Else output $(I'_l, I'_r)$. There is a feasible solution only if $(I'_l, I'_r)$ is feasible.

Figure 3–3: Obtaining a unidirectional plan with minimum separation constraint

velocity, at the time of the violation, it must be the case that the right leaf is stationary at a sample point (say $x_k$) and the left leaf is moving. The violation occurs when the left leaf passes $x' = x_k - S_{\text{min}}$. Since the left leaf is moving, $I_1 = I_l(x') < I_r(x_k)$. Figure 3–4 illustrates the situation. Suppose that there is another plan $(I''_l, I''_r)$ that does not violate the minimum separation constraint. Let $I''_l(x') = I_l(x') + \Delta(x')$ and let $I''_r(x_k) = I_r(x_k) + \Delta(x_k)$. From Lemma 12(a), $\Delta(x'), \Delta(x_k) \geq 0$ and from Lemma 12(b), $\Delta(x') \leq \Delta(x_k)$. Here, we have made use of the fact that in the statement of Lemma 12, we can replace the plan $(I_l, I_r)$ with the plan $(I'_l, I'_r)$. Now, $I''_l(x_k) - I''_l(x') = I_l(x_k) + \Delta(x_k) - (I_l(x') + \Delta(x')) = (I_r(x_k) - I_l(x')) + (\Delta(x_k) - \Delta(x'))$. Since $I_r(x_k) > I_l(x')$ and $\Delta(x_k) \geq \Delta(x')$, we get $I''_r(x_k) > I''_l(x')$. Therefore there is a minimum separation violation in $(I''_l, I''_r)$ when the the left leaf passes $x'$.

Figure 3–4: Minimum separation constraint violation
The separation between the leaves is determined by the difference in x values of the leaves when the source has delivered a certain amount of MUs. The minimum separation of the leaves is the minimum separation between the two profiles. We call this minimum separation $S_{ud-min}$. When the optimal plan obtained using Algorithm DMLC-SINGLEPAIR is delivered, the minimum separation is $S_{ud-min}^{(opt)}$.

**Corollary 7** Let $S_{ud-min}^{(opt)}$ be the minimum leaf separation in the plan $(I'_l, I'_r)$. Let $S_{ud-min}$ be the minimum leaf separation in any (not necessarily optimal) given unidirectional plan. $S_{ud-min} \leq S_{ud-min}^{(opt)}$.

**Theorem 13** If Algorithm DMLC-MINSINGLEPAIR terminates in Step 3, then there is no plan for I that does not violate the minimum separation constraint.

**Proof:** Let $(I''_l, I''_r)$ be a feasible plan (i.e., a plan that delivers I and satisfies the minimum separation constraint). Let $(I'_l, I'_r)$ be the plan of Step 2 of DMLC-MINSINGLEPAIR. From Corollary 5, it follows that $\hat{I}''_r(x_0 + S_{min}) \geq \hat{I}'_r(x_0 + S_{min}) + I''_r(x_0)$, where $\hat{I}_r(x)$ is the number of MUs delivered when the right leaf reaches x (note that $I_r(x)$ is the number of MUs delivered when the right leaf leaves x). Since $(I'_l, I'_r)$ has no minimum separation violation, $I''_l(x_0) \geq \hat{I}''_l(x_0) \geq \hat{I}'_l(x_0 + S_{min})$. Also, because DMLC-MINSINGLEPAIR terminates in Step 3, $\hat{I}'_l(x_0) > I_l(x_0)$. Hence, $I''_l(x_0) - I''_r(x_0) \geq \hat{I}'_r(x_0 + S_{min}) = \hat{I}'_l(x_0) > I_l(x_0) = I(x_0)$. So, $(I''_l, I''_r)$ does not deliver the proper dose at $x_0$ and so cannot be feasible.

**Comparison with SMLC.** Kamath et al. (2003) discuss the optimal therapy time algorithm for SMLC. Their algorithm generates an optimal plan that satisfies the minimum separation constraint, whenever one exists. We prove the existence of profiles for which there are feasible plans using SMLC, but no feasible plans using DMLC.

**Lemma 13** Let the minimum separation between the leaves in the optimal SMLC plan be $S_s-min$. Let the minimum leaf separation in the plan generated by algorithm DMLC-MINSINGLEPAIR be $S_d-min$. Then $S_d-min \leq S_s-min$.

**Proof:** Consider the delivery of a profile I by the optimal SMLC plan of Kamath et al. (2003). Call this plan $(I^*_l, I^*_r)$. Let $\hat{I}^*_l(x)$ be the number of MUs delivered when the left leaf arrives at x using the plan $I^*_l$. $I^*_r(x)$, and $\hat{I}_r(x)$ are defined similarly. Let $\Gamma_l(x_k) = I^*_l(x_k) - \hat{I}^*_l(x_k) = I'_l(x_k) - \hat{I}'_l(x_k)$. $\Gamma_r(x_k)$ is defined similarly. Note that $\Gamma_l(x_k) > 0$.
iff the left leaf stops at $x_k$ and $\Gamma_l(x_k)$ gives the amount of MUs delivered while the left leaf is stopped at $x_k$. Let $x_i$ and $x_j$, $j > i$, be such that $S_{s_{-\min}} = x_j - x_i$ and $\hat{I}_l^s(x_i) < I_r^s(x_j)$. Such an $x_i$ and $x_j$ must exist by definition of $S_{s_{-\min}}$.

It is easy to see that $I_r^s(x_j) - I_r^s(x_i) = \sum_{k=i+1}^{j} \Gamma_r(x_k)$. From this and $\hat{I}_l^s(x_i) < I_r^s(x_j)$, we get

$$\sum_{k=i+1}^{j} \Gamma_r(x_k) > \hat{I}_l^s(x_i) - I_r^s(x_i) \tag{3.2}$$

Since $I(x_i) = I_l^s(x_i) - I_r^s(x_i) = \hat{I}_l^s(x_i) + \Gamma_l(x_i) - I_r^s(x_i) = \hat{I}_l^s(x_i) + \Gamma_l(x_i) - I_r^s(x_i)$,

$$I_r^s(x_i) = \hat{I}_l^s(x_i) - (\hat{I}_l^s(x_i) - I_r^s(x_i)) \tag{3.3}$$

Also, we see that $I_l^s(x_i) = I_l^s(x_i) + \sum_{k=i+1}^{j} \Gamma_r(x_k) + (j - i) \Phi \Delta x/v_{\max} = \hat{I}_l^s(x_i) - (\hat{I}_l^s(x_i) - I_r^s(x_i)) + \sum_{k=i+1}^{j} \Gamma_r(x_k) + (j - i) \Phi \Delta x/v_{\max}$ (from (3.3)) > $\hat{I}_l^s(x_i) + (j - i) \Phi \Delta x/v_{\max}$ (from (3.2)). So, $S_{d_{-\min}} \leq x_j - x_i = S_{s_{-\min}}$.

The following result immediately follows and can be easily verified.

**Corollary 8** All profiles that have feasible plans using DMLC have feasible plans using SMLC. There exist profiles for which there are feasible plans using SMLC, but no feasible plans using DMLC.

Figure 3–5 shows two plans for an intensity profile. The SMLC plan for the profile is feasible. The corresponding DMLC plan obtained using Algorithm DMLC-MINSINGLEPAIR is infeasible.

3.1.5 Bi-directional Movement

In this section we study beam delivery when bi-directional movement of leaves is permitted. We explore whether relaxing the unidirectional movement constraint helps improve the efficiency of treatment plan.

**Properties of bi-directional movement.** For a given leaf (left or right) movement profile we classify any $x$-coordinate as follows. Draw a vertical line at $x$. If the line cuts the leaf profile exactly once we will call $x$ a unidirectional point of that leaf profile. If the line cuts the profile more than once, $x$ is a bi-directional point of that profile. A leaf movement profile that has at least one bi-directional point is a bi-directional profile. All profiles that are
Figure 3–5: SMLC plan: feasible; DMLC plan: infeasible
not bi-directional are unidirectional profiles. Any profile can be partitioned into segments such that each segment is a unidirectional profile. When the number of such partitions is minimal, each partition is called a stage of the original profile. Thus unidirectional profiles consist of exactly one stage while bi-directional profiles always have more than one stage.

In Figure 3–6, the bi-directional leaf movement profile, $B_L$, shows the position of the left leaf as a function of the amount of MUs delivered by the source. The movement profile of this leaf consists of stages $S_1, S_2$ and $S_3$. In stages $S_1$ and $S_3$ the leaf moves from left to right while in stage $S_2$ the leaf moves from right to left. $x_j$ is a bi-directional point of $B_L$. Let $I_L$ be the intensity profile corresponding to the leaf movement profile $B_L$. $I_L(x)$ gives the number of MUs delivered at $x$ using movement profile $B_L$. The amount of MUs delivered at $x_j$ is $L_1 + L_2$. In stage $S_1$, when $I_1$ amount of MUs have been delivered, the leaf passes $x_j$. Now, no MU is delivered at $x_j$ till the leaf passes over $x_j$ in $S_2$. Further, $I_3 - I_2$ MUs are delivered to $x_j$ in stages $S_2$ and $S_3$. Thus we have $I_1(x_j) = L_1 + L_2$, where $L_1 = I_1$ and $L_2 = I_3 - I_2$. $x_k$ is a unidirectional point of $B_L$. The MUs delivered at $x_k$ are $L_3 = I_4$. Note that the intensity profile $I_L$ is different from the leaf movement profile $B_L$, unlike in the unidirectional case.

![Figure 3–6: Bi-directional movement](image)

**Lemma 14** Let $I_L$ and $I_R$ be the intensity profiles corresponding to the bi-directional leaf movement profile pair $(B_L, B_R)$ (i.e., $B_L$ and $B_R$ are, respectively, the left and right leaf
Let \( I(x_i) = I_L(x_i) - I_R(x_i) \), \( 0 \leq i \leq m \), be the intensity profile delivered by \((B_L, B_R)\). Then

(a) \( I_L(x_{i+1}) \geq I_L(x_i) + \Phi \Delta x/v_{\text{max}} \).

(b) \( I_R(x_{i+1}) \geq I_R(x_i) + \Phi \Delta x/v_{\text{max}} \).

**Proof:** (a) Between the time, \( t_1 \), the left leaf moves rightward from \( x_i \) for the last time (since the left leaf ends at \( x_m \), such a last right move must occur) and the least time \( t_2 \), \( t_2 > t_1 \), that the left leaf reaches \( x_{i+1} \) (again, since the left leaf ends at \( x_m \), such a \( t_2 \) exists), \( x_{i+1} \) receives at least \( \Phi \Delta x/v_{\text{max}} \) MUs that are not delivered to \( x_i \). At all other times during the therapy, whenever the left leaf doesn’t cover \( x_i \), it also doesn’t cover \( x_{i+1} \). Hence, outside the time interval \([t_1, t_2] \), the number of MUs delivered to \( x_{i+1} \) is at least as many as delivered to \( x_i \). Therefore, for the entire therapy, \( I_L(x_{i+1}) \geq I_L(x_i) + \Phi \Delta x/v_{\text{max}} \).

(b) The proof is similar to that of (a). 

From Lemma 14 we note that every bi-directional leaf movement profile \((B_L, B_R)\) delivers an intensity profile \((I_L, I_R)\) that satisfies the maximum velocity constraint. Hence, \((I_L, I_R)\) is deliverable using a unidirectional leaf movement profile (Section 3.1.3). We will call this profile the **unidirectional leaf movement profile that corresponds to the bi-directional profile**. Thus every bi-directional leaf movement profile has a corresponding unidirectional leaf profile that delivers the same amount of MUs at each \( x_i \) as does the bi-directional profile.

**Theorem 14** The unidirectional treatment plan constructed by Algorithm DMLC - SINGLEPAIR is optimal in therapy time even when bi-directional leaf movement is permitted.

**Proof:** Let \( B_L \) and \( B_R \) be bidirectional leaf movement profiles that deliver a desired intensity profile \( I \). Let \( I_L \) and \( I_R \), respectively, be the corresponding intensity profiles for \( B_L \) and \( B_R \). From Lemma 14, we know that \( I_L \) and \( I_R \) are deliverable by unidirectional leaf movement profiles. Also, \( I_L(x_i) - I_R(x_i) = I(x_i), 1 \leq i \leq m \). From the proof of Theorem 11, it follows that the therapy time for the unidirectional plan \((I_L, I_R)\) generated by Algorithm DMLC-SINGLEPAIR is no more than that of \((I_L, I_R)\).

**Incorporating minimum separation constraint.** Let \( U_L \) and \( U_R \) be unidirectional leaf movement profiles that deliver the desired profile \( I(x_i) \). Let \( B_L \) and \( B_R \) be a set of
bi-directional left and right leaf profiles such that $U_l$ and $U_r$ correspond to $B_l$ and $B_r$ respectively, i.e., $(B_l, B_r)$ delivers the same plan as $(U_l, U_r)$. We call the minimum separation of leaves in this bi-directional plan $(B_l, B_r)$ $S_{bd-min}$. $S_{ud-min}$ is the minimum separation of leaves in $(U_l, U_r)$.

**Theorem 15** $S_{bd-min} \leq S_{ud-min}$ for every bi-directional leaf movement profile pair $(B_l, B_r)$ and its corresponding unidirectional profile $(U_l, U_r)$.

**Proof:** Suppose that the minimum separation $S_{ud-min}$ occurs when $I_{ms}$ MUs are delivered. At this time, the left leaf arrives at $x_j$ and the right leaf is positioned at $x_k$. Let $B'_l$ and $U'_l$ respectively, be the profiles obtained when $B_l$ and $U_l$ are shifted right by $S_{ud-min}$. Since $U'_l$ is $U_l$ shifted right by $S_{ud-min}$ and since the distance between $U_l$ and $U_r$ is $S_{ud-min}$ when $I_{ms}$ MUs have been delivered, $U'_l$ and $U_r$ touch when $I_{ms}$ units have been delivered. Therefore, the total MUs delivered by $(U'_l, U_r)$ at $x_k$ is zero. So the total MUs delivered by $(B'_l, B_r)$ at $x_k$ is also zero (recall that $U'_1$ and $U_r$, respectively, are corresponding profiles for $B'_l$ and $B_r$). This isn’t possible if $B_r$ is always to the right of $B'_l$ (for example, in the situation of Figure 3–7, the MUs delivered by $(B'_l, B_r)$ at $x_k$ are $(L_1 + L_2) - (L'_1 + L'_2) > 0$). Therefore $B'_l$ and $B_r$ must touch (or cross) at least once. So $S_{bd-min} \leq S_{ud-min}$. 

![Figure 3–7: Bi-directional movement under minimum separation constraint](image-url)
Theorem 16 If the optimal unidirectional plan \((I_l', I_r')\) violates the minimum separation constraint, then there is no bi-directional movement plan that does not violate the minimum separation constraint.

Proof: Let \(B_l\) and \(B_r\) be bi-directional leaf movements that deliver the required profile. Let the minimum separation between the leaves be \(S_{bd-min}\). Let the corresponding unidirectional leaf movements be \(U_l\) and \(U_r\) and let \(S_{ud-min}\) be the minimum separation between \(U_l\) and \(U_r\). Also, let \(S_{min}\) be the minimum allowable separation between the leaves. From Corollary 7 and Theorem 15, we get \(S_{bd-min} \leq S_{ud-min} \leq S_{ud-min(opt)} < S_{min}\).

Incorporating maximum separation constraint. Let \(U_l\) and \(U_r\) be unidirectional leaf movement profiles that deliver the desired profile \(I\). Let \(S_{ud-max}\) be the maximum leaf separation using the profiles \(U_l\) and \(U_r\) and let \(S_{ud-max(opt)}\) be the maximum leaf separation for the plan \((I_l, I_r)\) generated by Algorithm DMLC-SINGLEPAIR. Let \(B_l\) and \(B_r\) be a set of bi-directional left and right leaf profiles such that \(U_l\) and \(U_r\) correspond to \(B_l\) and \(B_r\), respectively. Let \(S_{bd-max}\) be the maximum separation between the leaves when these bi-directional movement profiles are used.

Theorem 17 \(S_{bd-max} \geq S_{ud-max}\) for every bi-directional leaf movement profile and its corresponding unidirectional movement profile.

Proof: Suppose that the maximum separation \(S_{ud-max}\) occurs when \(I_{ms}\) MUs are delivered. At this time, the left leaf is positioned at \(x_j\) and the right leaf arrives at \(x_k\). Let \(B_l'\) and \(U_l'\) respectively, be the profiles obtained when \(B_l\) and \(U_l\) are shifted right by \(S_{ud-max}\). Since \(U_l'\) is \(U_l\) shifted right by \(S_{ud-max}\) and since the distance between \(U_l\) and \(U_r\) is \(S_{ud-max}\) when \(I_{ms}\) MUs have been delivered, \(U_l'\) and \(U_r\) touch when \(I_{ms}\) units have been delivered. Therefore, the total MUs delivered by \((U_r, U_l')\) at \(x_k\) is zero. So the total MUs delivered by \((B_r, B_l')\) at \(x_k\) is also zero (recall that \(U_l'\) and \(U_r\), respectively, are corresponding profiles for \(B_l'\) and \(B_r\)). This isn’t possible if \(B_r\) is always to the left of \(B_l'\) (for example, in the situation of Figure 3–8, the MUs delivered by \((B_r, B_l')\) at \(x_k\) are \((L_1' + L_2') - (L_1 + L_2) > 0\)). Therefore \(B_l'\) and \(B_r\) must touch (or cross) at least once. So \(S_{bd-max} \geq S_{ud-max}\).

3.1.6 Algorithm Under Maximum Separation Constraint Condition

In this section we present an algorithm that generates an optimal treatment plan under the maximum separation constraint. Recall that Algorithm DMLC-SINGLEPAIR
generates the optimal plan without considering this constraint. We modify Algorithm DMLC-SINGLEPAIR so that all instances of violation of maximum separation (that may possibly exist) are eliminated. We know (Theorem 17) that bi-directional leaf profiles do not help eliminate the constraint. So we consider only unidirectional profiles.

**Algorithm.** The algorithm is described in Figure 3–9.

Algorithm DMLC-MAXSEPARATION
1. Apply Algorithm DMLC-SINGLEPAIR to obtain the optimal plan \((I_l, I_r)\).
2. Find the least value of intensity, \(I_1\), such that the leaf separation in \((I_l, I_r)\) when \(I_1\) MUs are delivered is \(S_{max}\), where \(S_{max}\) is the maximum allowed separation between the leaves and the leaf separation when \(I_1 + \epsilon\) MUs are delivered is \(> S_{max}\), for some positive constant \(\epsilon\). If there is no such \(I_1\), \((I_l, I_r)\) is the optimal plan; end.
3. From Lemma 10 it follows that when \(I_1\) MUs are delivered, the left leaf is stopped at some \(x_j\). Let \(x'\) be the position of the right leaf at this time (see Figure 3–10). Note that \(x'\) may not be one of the sample points \(x_i\), \(j \leq i \leq m\). Let \(\Delta I = I_l(x_j) - I_1 = I_2 - I_1\). Move the profile of \(I_r\), which follows \(x'\), up by \(\Delta I\) along \(I\) direction. To maintain \(I(x) = I_l(x) - I_r(x)\) for every \(x\), move the profile of \(I_l\), which follows \(x'\), up by \(\Delta I\) along \(I\) direction.

Goto Step 2.

Figure 3–9: Obtaining a plan under maximum separation constraint

**Theorem 18** Algorithm DMLC-MAXSEPARATION obtains plans that are optimal in therapy time, under the maximum separation constraint.
Proof: We use induction to prove the theorem.

The statement we prove, $S(n)$, is the following:

After Step (iii) of the algorithm is applied $n$ times, the resulting plan, $(I_{ln}, I_{rn})$, satisfies

(a) It has no maximum separation violation when $I < I_2(n)$ MUs are delivered, where $I_2(n)$ is the value of $I_2$ during the $n$th iteration of Algorithm DMLC-MAXSEPARATION.

(b) For plans that satisfy (a), $(I_{ln}, I_{rn})$ is optimal in therapy time.

1. Consider the base case, $n = 1$.

Let $(I_l, I_r)$ be the plan generated by Algorithm DMLC-SINGLEPAIR. After Step (iii) is applied once, the resulting plan $(I_{l1}, I_{r1})$ meets the requirement that there is no maximum separation violation when $I < I_2(1)$ MUs are delivered by the radiation source. The therapy time increases by $\Delta I$, i.e., $TT(I_{l1}, I_{r1}) = TT(I_l, I_r) + \Delta I$.

Assume that there is another plan, $(I'_{l1}, I'_{r1})$, which satisfies condition (a) of $S(1)$ and $TT(I'_{l1}, I'_{r1}) < TT(I_{l1}, I_{r1})$. We show this assumption leads to a contradiction and so there is no such plan $(I'_{l1}, I'_{r1})$.

Let $x_j$ and $x'$ be as in Algorithm DMLC-MAXSEPARATION. We consider three cases for the relationship between $I_{l1}(x_j)$ and $I_{l1}(x_j)$. 

---

Figure 3–10: Maximum separation constraint violation
(a) \( I'_1(x_j) = I_{l1}(x_j) = I_2(1) \)

Since there is no maximum separation violation when \( I < I_2(1) \) MUs are delivered, \( I'_{r1}(x') \geq I'_1(x_j) = I_{l1}(x_j) = I_{r1}(x') \). Since \( I(x') = I'_1(x') - I'_{r1}(x') = I_1(x') - I_{r1}(x') \), we have \( I'_1(x') \geq I_{l1}(x') \). We now construct a plan \((I''_{l1}, I''_{r1})\) as follows:

\[
I''_{l1}(x) = \begin{cases} 
I_l(x) & 0 \leq x < x' \\
I'_1(x) - \Delta I & x \geq x'
\end{cases}
\]

\[
I''_{r1}(x) = \begin{cases} 
I_r(x) & 0 \leq x < x' \\
I'_{r1}(x) - \Delta I & x \geq x'
\end{cases}
\]

Clearly \( I''_{l1}(x) - I''_{r1}(x) = I(x), 0 \leq x \leq x_m \). Also, \( I''_{l1} \) is non-decreasing and satisfies the maximum velocity constraint \( I''_{l1}(x') = I'_1(x') - \Delta I \geq I_{l1}(x') - \Delta I = I_l(x') \geq I_l(x' - \Delta x) + \Phi \Delta x/v_{\text{max}} = I'_1(x' - \Delta x) + \Phi \Delta x/v_{\text{max}} \). Similarly \( I''_{r1} \) is non-decreasing and satisfies the maximum velocity constraint. So \((I''_{l1}, I''_{r1})\) is a plan for \( I(x_i) \).

Also, \( TT(I''_{l1}, I''_{r1}) = TT(I'_1, I'_{r1}) - \Delta I < TT(I_{l1}, I_{r1}) - \Delta I = TT(I_l, I_r) \).

This contradicts our knowledge that \( (I_l, I_r) \) is the optimal unconstrained plan.

(b) \( I'_1(x_j) > I_{l1}(x_j) \)

This leads to a contradiction as in the previous case.

(c) \( I'_1(x_j) < I_{l1}(x_j) \)

In this case, \( I'_1(x_j) < I_{l1}(x_j) = I_l(x_j) \). This violates Corollary 5. So this case cannot arise.

Therefore \( S(1) \) is true.

2. Induction step

Assume \( S(n) \) is true. If there are no more maximum separation violations in the resulting plan, \((I_m, I_{rn})\), then it is the optimal plan. If there are more violations, we find the next violation. Apply Step (iii) of the algorithm to get a new plan. Assume that there is another plan, which costs less time than the plan generated by Algorithm DMLC-MAXSEPATION. We consider three cases as in the base case and show by
contradiction that there is no such plan. Therefore \( S(n + 1) \) is true whenever \( S(n) \) is true.

Since the number of iterations of Steps (ii) and (iii) of the algorithm is finite (for each iteration, the left leaf must be stationary at \( x_j \) and there can be at most one iteration for each such \( x_j \)), all maximum separation violations will eventually be eliminated.

When the minimum separation constraint is also applicable, we can use Algorithm DMLC-MINSINGLEPAIR in place of Algorithm DMLC-SINGLEPAIR in Step (i) of Algorithm DMLC-MAXSEPARATION. Note that in this case the minimum leaf separation of the plan constructed by Algorithm DMLC-MAXSEPARATION is \( \min\{S_{ud-min(opt)}, S_{max}\} \).

From Theorem 18, it follows that Algorithm DMLC-MAXSEPARATION constructs an optimal plan that satisfies both the minimum and maximum separation constraints provided that \( S_{ud-min(opt)} \geq S_{min} \). Note that when \( S_{ud-min(opt)} < S_{min} \), there is no plan that satisfies the minimum separation constraint.

3.1.7 Algorithm Under Interdigitation Constraint

Introduction. The inter-pair minimum separation constraint with \( S_{min} = 0 \) is of special interest and is referred to as the interdigitation constraint. Recall that, in Section 3.1.3, we proved that for a single pair of leaves, if the optimal plan does not satisfy the minimum separation constraint, then no plan satisfies the constraint. In this section we present an algorithm to generate the optimal schedule for the desired profile defined over a 2-D region. We then modify the algorithm to generate schedules that satisfy the interdigitation constraint. Note that in our discussion on single pair of leaves (Section 3.1.1), we assumed that \( I(x_0) > 0 \) and that \( I(x_m) > 0 \). However, with multiple leaf pairs, the first and last sample points with non-zero intensity levels could be different for different pairs. Hence we will no longer make this assumption.

Optimal schedule without the interdigitation constraint. For sequencing of multiple leaf pairs, we apply Algorithm DMLC-SINGLEPAIR to determine the optimal plan for each of the \( n \) leaf pairs. This method of generating schedules is described in Algorithm DMLC-MULTIPAIR (Figure 3–11). Note that since \( x_0, x_m \) are not necessarily non-zero for any row, we replace \( x_0 \) by \( x_l \) and \( x_m \) by \( x_g \) in Algorithm DMLC-SINGLEPAIR for each row,
where \( x_l \) and \( x_g \), respectively, denote the first and last non-zero sample points of that row. Also, for rows that contain only zeroes, the plan simply places the corresponding leaves at the rightmost point in the field (call it \( x_{m+1} \)).

Algorithm DMLC-MULTIPAIR
For \( i = 1; i \leq n; i + + \)
- Apply Algorithm DMLC-SINGLEPAIR to the \( i \)th pair of leaves to obtain plan \((I_{il}, I_{ir})\) that delivers the intensity profile \( I_i(x) \).
End For

Figure 3–11: Obtaining a schedule

**Lemma 15** Algorithm DMLC-MULTIPAIR generates schedules that are optimal in therapy time.

**Proof:** Treatment is completed when all leaf pairs (which are independent) deliver their respective plans. The therapy time of the schedule generated by Algorithm DMLC-MULTIPAIR is \( \max\{TT(I_{1l}, I_{1r}), TT(I_{2l}, I_{2r}), \ldots, TT(I_{nl}, I_{nr})\} \). From Theorem 11, it follows that this therapy time is optimal.

**Optimal algorithm with interdigitation constraint.** The schedule generated by Algorithm DMLC-MULTIPAIR may violate the interdigitation constraint. Note that no intra-pair constraint violations can occur for \( S_{min} = 0 \). So the interdigitation constraint is essentially an inter-pair constraint. If the schedule has no interdigitation constraint violations, it is the desired optimal schedule. If there are violations in the schedule, we eliminate all violations of the interdigitation constraint starting from the left end, i.e., from \( x_0 \). To eliminate the violations, we modify those plans of the schedule that cause the violations.

We scan the schedule from \( x_0 \) along the positive \( x \) direction looking for the least \( x_v \) at which is positioned a right leaf (say \( R_u \)) that violates the inter-pair separation constraint. After rectifying the violation at \( x_v \) with respect to \( R_u \) we look for other violations. Since the process of eliminating a violation at \( x_v \), may at times, lead to new violations involving right leaves positioned at \( x_v \), we need to search afresh from \( x_v \) every time a modification is made to the schedule. We now continue the scanning and modification process until no interdigitation violations exist. Algorithm DMLC-INTERDIGITATION (Figure 3–12) outlines the procedure.
Algorithm DMLC-INTERDIGITATION
1. \( x = x_0 \)
2. While (there is an interdigitation violation) do
3. Find the least \( x_v, x_v \geq x \), such that a right leaf is positioned at \( x_v \) and this right leaf has an interdigitation violation with one or both of its neighboring left leaves. Let \( u \) be the least integer such that the right leaf \( R_u \) is positioned at \( x_v \) and \( R_u \) has an interdigitation violation. Let \( L_t \) denote the left leaf with which \( R_u \) has an interdigitation violation. Note that \( t \in \{ u - 1, u + 1 \} \). In case \( R_u \) has violations with two adjacent left leaves, we let \( t = u - 1 \).
4. Modify the schedule to eliminate the violation between \( R_u \) and \( L_t \).
5. \( x = x_v \)
6. End While

Figure 3–12: Obtaining a schedule under the constraint

Let \( M = ((I_{1l}, I_{1r}), (I_{2l}, I_{2r}), \ldots, (I_{nl}, I_{nr})) \) be the schedule generated by Algorithm DMLC-MULTIPAIR for the desired intensity profile.

Let \( N(p) = ((I_{1lp}, I_{1rp}), (I_{2lp}, I_{2rp}), \ldots, (I_{nlp}, I_{nrp})) \) be the schedule obtained after Step 4 of Algorithm DMLC-INTERDIGITATION is applied \( p \) times to the input schedule \( M \). Note that \( M = N(0) \).

To illustrate the modification process we use examples. There are two types of violations that may occur. Call them Type1 and Type2 violations and call the corresponding modifications Type1 and Type2 modifications. To make things easier, we only show two neighboring pairs of leaves. Suppose that the \((p+1)\)th violation occurs between the right leaf of pair \( u \), which is positioned at \( x_v \), and the left leaf of pair \( t, t \in \{ u - 1, u + 1 \} \).

In a Type1 violation, the left leaf of pair \( t \) starts its sweep at a point \( x_{\text{Start}}(t, p) > x_v \) (see Figure 3–13). To remove this interdigitation violation, modify \((I_{tlp}, I_{trp})\) to \((I_{tl(p+1)}, I_{tr(p+1)})\) as follows. We let the leaves of pair \( t \) start at \( x_v \) and move them at the maximum velocity \( v_{\text{max}} \) towards the right, till they reach \( x_{\text{Start}}(t, p) \). Let the number of MUs delivered when they reach \( x_{\text{Start}}(t, p) \) be \( I_1 \). Raise the profiles \( I_{tlp}(x) \) and \( I_{trp}(x) \), \( x \geq x_{\text{Start}}(t, p) \), by an amount \( I_1 = \Phi \ast (x_{\text{Start}}(t, p) - x_v)/v_{\text{max}} \). We get,

\[
I_{tl(p+1)}(x) = \begin{cases} \Phi \ast (x - x_v)/v_{\text{max}} & x_v \leq x < x_{\text{Start}}(t, p) \\ I_{tlp}(x) + I_1 & x \geq x_{\text{Start}}(t, p) \end{cases}
\]

\[
I_{tr(p+1)}(x) = I_{tl(p+1)}(x) - I_t(x), \text{ where } I_t(x) \text{ is the target profile to be delivered by the leaf pair } t.
\]
A Type 2 violation occurs when the left leaf of pair $t$, which starts its sweep from $x \leq x_v$, passes $x_v$ before the right leaf of pair $u$ passes $x_v$ (Figure 3–14). In this case, $I_{tl(p+1)}$ is as defined below.

$$I_{tl(p+1)}(x) = \begin{cases} I_{tlp}(x) & x < x_v \\ I_{tlp}(x) + \Delta x_v & x \geq x_v \end{cases}$$

where $\Delta x_v = I_{urp}(x_v) - I_{tlp}(x_v) = I_3 - I_2$. Once again, $I_{tr(p+1)}(x) = I_{tl(p+1)}(x) - I_t(x)$, where $I_t(x)$ is the target profile to be delivered by the leaf pair $t$.

In both Type 1 and Type 2 modifications, the other profiles of $N(p)$ are not modified. Since $I_{tr(p+1)}$ differs from $I_{trp}$ for $x \geq x_v$ there is a possibility that $N(p+1)$ has interpair separation violations for right leaf positions $x \geq x_v$. Since none of the other right leaf profiles are changed from those of $N(p)$ and since the change in $I_{tl}$ only delays the rightward movement of the left leaf of pair $t$, no interdigitation violations are possible in $N(p+1)$ for $x < x_v$. One may also verify that since $I_{tl0}$ and $I_{trl0}$ are feasible plans that satisfy the maximum velocity constrains, so also are $I_{tlp}$ and $I_{trlp}$, $p > 0$.

**Lemma 16** $I_{jrp}(x_{Start(j,p)}) = 0$, $1 \leq j \leq m$, $p \geq 0$.

**Proof:** The proof is by induction on $p$. Let $T(p)$ be the following statement: $I_{jrp}(x_{Start(j,p)}) = 0$. 

- **Base Case:** For $p = 0$, $I_{jrp}(x_{Start(j,p)}) = 0$ holds for all $j$, since $I_{jrp}(x_{Start(j,0)}) = 0$ due to the initial conditions.
- **Inductive Step:** Assume $T(p)$ holds for some $p \geq 0$. For $p + 1$, consider $I_{jrp}(x_{Start(j,p+1)})$. By the definition of $I_{jrp}(x_{Start(j,p+1)})$, we have $I_{jrp}(x_{Start(j,p+1)}) = I_{jrp}(x_{Start(j,p)}) + \Delta x_v$. Since $I_{jrp}(x_{Start(j,p)}) = 0$ by the inductive hypothesis, we have $I_{jrp}(x_{Start(j,p+1)}) = \Delta x_v$. But $\Delta x_v = I_{urp}(x_v) - I_{tlp}(x_v) = I_3 - I_2$, which is non-negative due to the feasibility of $I_{jrp}(x_{Start(j,p)})$. Therefore, $I_{jrp}(x_{Start(j,p+1)}) = 0$.

By the principle of mathematical induction, $T(p)$ holds for all $p$. 

**Figure 3–13: Eliminating a Type1 violation**

![Figure 3–13: Eliminating a Type1 violation](image-url)
Figure 3–14: Eliminating a Type2 violation (close parallel dotted and solid line segments overlap, they have been drawn with a small separation to enhance readability)

- For the base case, \( p = 0 \). \((I_{j0}, I_{jr0})\) is the plan generated by Algorithm DMLC-SINGLEPAIR and it satisfies the stated property.

- Assume that \( T(p) \) is true. For the \((p+1)\)th violation, we have the following two cases:
  - The \((p+1)\)th violation is a Type1 violation.
    A Type1 modification is applied. Such a modification always results in changing the start position of the leaves of pair \( t \) (as defined in Algorithm DMLC-INTERDIGITATION) to \( x_v \) (which becomes \( x_{\text{Start}(t, p+1)} \)) and \( I_{tr(p+1)}(x_v) = 0 \). For \( j \neq t \), \( I_{jr(p+1)}(x_{\text{Start}(j, p+1)}) = I_{jrp}(x_{\text{Start}(j, p)}) = 0 \) by induction hypothesis.
  - The \((p+1)\)th violation is a Type2 violation.
    A Type2 modification is applied. Let \( t \) be as in Algorithm DMLC - INTERDIGITATION. Suppose that \( x_{\text{Start}(t, p)} < x_v \). Since a Type2 modification does not alter the plan for \( x < x_v \), \( I_{tr(p+1)}(x_{\text{Start}(t, p+1)}) = I_{trp}(x_{\text{Start}(t, p)}) = 0 \). If \( x_{\text{Start}(t, p)} = x_v \), it must be the case that \( x_{\text{Start}(u, p)} = x_{\text{Start}(t, p)} \) (as otherwise, there is a Type1 violation at \( x_{\text{Start}(u, p)} < x_v \)). So the right leaf of pair \( u \) is not stopped at \( x_v \). Hence, there is no interdigitation violation at \( x_v \).
So, the case \(x_{\text{Start}}(t, p) = x_v\) cannot arise. For \(j \neq t\), the plan is unchanged.

So, \(I_{j(p+1)}(x_{\text{Start}}(j, p+1)) = I_{jrp}(x_{\text{Start}}(j, p)) = 0\) by induction hypothesis.

\[\square\]

**Corollary 9** A Type2 violation in which \(I_{tlp}(x_v) = 0\) cannot occur.

**Proof:** From the proof of Lemma 16, it follows that whenever there is a Type2 violation, \(x_{\text{Start}}(t, p) < x_v\). Hence, \(I_{tlp}(x_v) > 0\).

\[\square\]

**Lemma 17** In case of a Type1 violation, \((I_{tlp}, I_{trp})\) is the same as \((I_{tl0}, I_{tr0})\).

**Proof:** Let \(p\) be such that there is a Type1 violation. Let \(t, u\) and \(v\) be as in Algorithm DMLC-INTERDIGITATION. If \((I_{tlp}, I_{trp})\) is different from \((I_{tl0}, I_{tr0})\), leaf pair \(t\) was modified in an earlier iteration (say iteration \(q < p\)) of the while loop of Algorithm DMLC-INTERDIGITATION. Let \(v(q)\) be the \(v\) value in iteration \(q\). If iteration \(q\) was a Type1 violation, then \(x_{\text{Start}}(t, p) \leq x_{\text{Start}}(t, q+1) = x_{v(q)} \leq x_v\). So, iteration \(p\) cannot be a Type1 violation. If iteration \(q\) was a Type2 violation, \(x_{\text{Start}}(t, p) \leq x_{\text{Start}}(t, q) \leq x_{v(q)} \leq x_v\). Again, iteration \(p\) cannot be a Type1 violation. Hence, there is no prior iteration \(q, q < p\), when the profiles \((I_{tl}, I_{tr})\) were modified.

\[\square\]

**Lemma 18** (a) A Type1 modification eliminates a Type1 violation.

(b) A Type2 modification eliminates a Type2 violation.

**Proof:** (a) From Lemma 16, \(I_{urp}(x_v) = 0\). By changing the start position of leaf pair \(t\) to \(x_v\), we eliminate this violation.

(b) Follows from the construction of \((I_{tl(p+1)}, I_{tr(p+1)})\).

Note that \(I_{tlp}(x)\) and \(I_{trp}(x)\) are defined only for \(x \geq x_{\text{Start}}(t, p)\). In the sequel, we adopt the convention that \(z \geq I_{tlp}(x) (z \geq I_{irp}(x))\) is true whenever \(I_{tlp}(x) (I_{irp}(x))\) is undefined, irrespective of whether \(z\) is defined or not.

**Lemma 19** Let \(F = ((I'_{1l}, I'_{1r}), (I'_{2l}, I'_{2r}), \ldots, (I'_{nl}, I'_{nr}))\) be any feasible schedule for the desired profile, i.e., a schedule that does not violate the interdigitation constraint. Let \(S(p)\), be the following assertions.

(a) \(I'_{il}(x) \geq I_{tlp}(x), 0 \leq i \leq n, x_0 \leq x \leq x_m\)

(b) \(I'_{ir}(x) \geq I_{trp}(x), 0 \leq i \leq n, x_0 \leq x \leq x_m\)

\(S(p)\) is true for \(p \geq 0\).

**Proof:** The proof is by induction on \(p\).
1. Consider the base case, \( p = 0 \). From Corollary 5 and the fact that the plans \((I_{t0}, I_{r0}), 0 \leq i \leq n\), are generated using Algorithm DMLC-SINGLEPAIR, it follows that \( S(0) \) is true.

2. Assume \( S(p) \) is true. Suppose Algorithm DMLC-INTERDIGITATION finds a next violation and modifies the schedule \( N(p) \) to \( N(p+1) \). Suppose that the next violation occurs between the right leaf of pair \( u \), positioned at \( x_v \), and the left leaf of pair \( t \). We modify pair \( t \)'s plan for \( x \geq x_v \), to eliminate the violation. All other plans in the schedule remain unaltered. Therefore, to establish \( S(p+1) \) it suffices to prove that

\[
I'_{tl}(x) \geq I_{tl(p+1)}(x), x \geq x_v
\]  

(3.4)

\[
I'_{tr}(x) \geq I_{tr(p+1)}(x), x \geq x_v
\]  

(3.5)

We need prove only one of these two relationships since \( I'_{tl}(x) - I'_{tr}(x) = I_{tl(p+1)}(x) - I_{tr(p+1)}(x), x_0 \leq x \leq x_m \). We now consider pair \( t \)'s plan for \( x \geq x_v \). Note that the \((p+1)\)th violation may either be a Type1 or Type2 violation. We show that Equation 3.4 is true in both cases. This, in turn, implies that \( S(p+1) \) is true whenever \( S(p) \) is true and hence completes the proof. Note that in \((I_{tl(p+1)}, I_{tr(p+1)})\), the leaves move at maximum speed between adjacent sample points. So, it is sufficient to show Equation 3.4 for sample points \( \geq x_v \).

(a) The \((p+1)\)th violation is a Type1 violation.

From \( S(p) \) it follows that \( I'_{ur}(x_v) \geq I_{urp}(x_v) \). So, the right leaf of pair \( u \) leaves \( x_v \) no earlier in \( I'_{ur} \) than it does in \( I_{urp} \). From this and the fact that \( F \) satisfies the interdigitation constraint, we conclude that leaf pair \( t \) cannot begin its sweep at the right of \( x_v \). This observation together with the fact that in \((I_{tl(p+1)}, I_{tr(p+1)})\) the leaves move at the maximum velocity from \( x_v \) to \( x' = x_{\text{Start}}(t, p) \) implies that \( \hat{I}_{tl}(x') \geq \hat{I}_{tl(p+1)}(x') \) and \( \hat{I}_{tr}(x') \geq \hat{I}_{tr(p+1)}(x') \), where \( \hat{I} \) denotes an arrival time. Now, from Lemma 16, we get \( I'_{tr}(x') \geq \hat{I}'_{tr}(x') \geq \hat{I}_{tr(p+1)}(x') = I_{tr(p+1)}(x') \).

So \( I'_{tl}(x') = I'_{tr}(x') + I_I(x') \geq I_{tr(p+1)}(x') + I_I(x') = I_{tl(p+1)}(x') \). From this and the fact that the left leaf of pair \( t \) moves at the maximum velocity between
it follows that Equation 3.4 holds for all \( x \) between \( x_v \) and \( x' \). To prove that Equation 3.4 holds for all sample points to the right of \( x' \) (and so holds for all \( x \) between \( x_0 \) and \( x_m \)), consider a sample point \( x_w \) that is to the right of \( x' \). Let \( \Delta I' = I'_t(x') - I_{tl(p+1)}(x') \geq 0 \) and let \( I_1 \) be as in Algorithm DMLC-INTERDIGITATION. Define a new plan \((I''_t, I''_r)\) for leaf pair \( t \) as below

\[
I''_t(x) = \begin{cases} 
  \text{undefined} & x < x' \\
  I'_t(x) - \Delta I' - I_1 & x \geq x'
\end{cases}
\]

\[
I''_r(x) = \begin{cases} 
  \text{undefined} & x < x' \\
  I'_r(x) - \Delta I' - I_1 & x \geq x'
\end{cases}
\]

Note that \( I''_t(x') = I'_t(x') - \Delta I' - I_1 = I_t(x') - I_1 = I_{tlp}(x') \geq 0 \). Similarly, \( I''_r(x') \geq 0 \). Hence \((I''_t, I''_r)\) is a plan for \( I_t \). Also, \( I''_t(x_w) = I'_t(x_w) - \Delta I' - I_1 \leq I_t(x_w) - I_1 \). If \( I'_t(x_w) < I_{tl(p+1)}(x_w) \), \( I''_t(x_w) < I_{tl(p+1)}(x_w) - I_1 = I_{tlp}(x_w) = I_{tl0}(x_w) \) (Lemma 17). This contradicts Corollary 5. Hence, \( I'_t(x_w) \geq I_{tl(p+1)}(x_w) \).

(b) The \((p + 1)\)th violation is a Type2 violation.

The situation is illustrated in Figure 3–14. Since \( F \) satisfies the interdigitation constraint, the left leaf of pair \( t \) does not pass \( x_v \) before the right leaf of pair \( u \) passes \( x_v \). So,

\[
I'_t(x_v) \geq I'_{ur}(x_v)
\]  \( (3.6) \)

From \( S(p) \) and the definition of a Type2 modification, we get,

\[
I'_{ur}(x_v) \geq I_{arp}(x_v) = I_{tl(p+1)}(x_v)
\]  \( (3.7) \)

Equations 3.6 and 3.7 yield

\[
I'_t(x_v) - I_{tl(p+1)}(x_v) \geq 0
\]  \( (3.8) \)

Lemma 12b implies,

\[
I'_t(x) - I_t(x) \geq I'_t(x_v) - I_t(x_v), x \geq x_v
\]  \( (3.9) \)
(Lemma 12b yields Equation 3.9 only for \( x \geq x_v \) and \( x \) is a sample point. From this and the fact that the left leaf moves at maximum velocity in \( I_t \) between adjacent sample points, we get Equation 3.9 for all \( x, x \geq x_v \).)

From Equation 3.9, we get

\[
I_t'(x) - I_{t(p+1)}(x) \geq I_t'(x_v) - I_{t(p+1)}(x_v) - I_t(x_v) - I_t(x_v) - I_{t(p+1)}(x_v), x \geq x_v
\] (3.10)

From the definitions of Type1 and Type2 modifications and the working of Algorithm DMLC-INTERDIGITATION, it follows that

\[
I_{t(p+1)}(x) - I_t(x) = I_{t(p+1)}(x_v) - I_t(x_v), x \geq x_v
\] (3.11)

From Equations 3.10, 3.11 and 3.8, we get

\[
I_t'(x) - I_{t(p+1)}(x) \geq I_t'(x_v) - I_{t(p+1)}(x_v) \geq 0, x \geq x_v
\] (3.12)

Therefore,

\[
I_t'(x) \geq I_{t(p+1)}(x), x \geq x_v
\] (3.13)

\[\square\]

**Lemma 20** For the execution of Algorithm DMLC-INTERDIGITATION

(a) \( O(n) \) Type1 violations can occur.

(b) \( O(n^2 m) \) Type2 violations can occur.

(c) Let \( T_{\text{max}} \) be the optimal therapy time for the input matrix. The time complexity is

\( O(m n + n \times \min\{m n, T_{\text{max}}\}) \).

**Proof:** (a) It follows from Lemma 17 that each leaf pair can be involved in at most one Type1 violation as pair \( t \), i.e., the pair whose profile is modified. Hence, the number of Type1 violations is \( \leq n \).

(b) We first obtain a bound on the number of Type2 violations at a fixed \( x_v \). Let \( u, t \) be as in Algorithm DMLC-INTERDIGITATION. Note that \( u \) is chosen to be the least possible index. Let \( u_i \) be the value of \( u \) in the \( i \)th iteration of Algorithm DMLC-INTERDIGITATION at \( x_v \). \( t_i \) is defined similarly. Let \( u_{i_{\text{max}}}^i = \max_{j \leq i}\{u_j\} \). If \( t_i = u_i - 1 \), it is possible that \( u_{i+1} = t_i = u_i - 1 \) and \( t_{i+1} = u_i - 2 \). Note that in this case, \( t_{i+1} \neq u_i = u_{i+1} + 1 \). Next,
it is possible that $u_{i+2} = u_i - 2$ and $t_{i+2} = u_{i-3}$ (again $t_{i+2} \neq u_i - 1 = u_{i+2} + 1$). In general, one may verify that $t_i = u_i + 1$ is possible only if $u_i^{\max} = u_i$. If $t_i = u_i + 1$, then $u_{i+1} \geq t_i = u_i + 1$, since the violation between $u_i$ and $t_i$ has been eliminated and no profiles with an index less than $t_i$ have been changed during iteration $i$ at $x_v$. It is also easy to verify that $t_i = 1, u_i = 2 = u_{i+1} \geq u_i^{\max}, u_{i+2}^{\max} > u_i^{\max}$. From this and $t_i \in \{u_i + 1, u_i - 1\}$ it follows that $u_i^{\max} > u_i^{\max}$. We know that $u_i^{\max} \geq 1$. It follows that $u_2^{\max} \geq 2, u_4^{\max} \geq 3, u_7^{\max} \geq 4$ and in general, $u_{i(i+1)/2+1}^{\max} \geq i + 1$. Clearly, for the last violation (say $j$th) at $x_v$, $u_j^{\max} \leq n$ and for this to be true, $j = O(n^2)$. So the number of Type2 violations at $x_v$ is $O(n^2)$. Since $x_v$ has to be a sample point, there are $m$ possible choices for it. Hence, the total number of Type2 violations is $O(n^2m)$.

(c) Since the input matrix contains only integer intensity values, each violation modification raises the profile for one pair of leaves by at least one unit. Hence, if $T_{\max}$ is the optimal therapy time, no profile can be raised more than $T_{\max}$ times. Therefore, the total number of violations that Algorithm DMLC-INTERDIGITATION needs to repair is at most $nT_{\max}$. Combining this bound with those of (a) and (b), we get $O(\min\{n^2m, nT_{\max}\})$ as a bound on the total number of violations repaired by Algorithm DMLC-INTERDIGITATION. By proper choice of data structures and programming methods it is possible to implement Algorithm DMLC-INTERDIGITATION so as to run in $O(mn + n \cdot \min\{nm, T_{\max}\})$ time.

Note that Lemma 20 provides two upper bounds of on the complexity of Algorithm DMLC-INTERDIGITATION: $O(n^2m)$ and $O(n \cdot \max\{m, T_{\max}\})$. In most practical situations, $T_{\max} < nm$ and so $O(n \cdot \max\{m, T_{\max}\})$ can be considered a tighter bound.

**Theorem 19** The following are true of Algorithm DMLC-INTERDIGITATION:

(a) The algorithm terminates.

(b) The schedule generated is feasible and is optimal in therapy time for unidirectional schedules.

**Proof:**

(a) Lemma 20 provides a polynomial upper bound ($O(n^2m)$) on the complexity of Algorithm DMLC-INTERDIGITATION. The result follows from this.
(b) When the algorithm terminates, no interdigitation violations remain and the final schedule is feasible. From Lemma 19, it follows that the final schedule is optimal in therapy time for unidirectional schedules.

3.2 Conclusion

We have presented mathematical formalisms and rigorous proofs of leaf sequencing algorithms for dynamic multileaf collimation that maximize MU efficiency. These leaf sequencing algorithms explicitly account for leaf interdigitation constraint. We have shown that our algorithms obtain feasible solutions that are optimal in treatment MUs. Furthermore, our analysis shows that unidirectional leaf movement is at least as efficient as bi-directional movement. Thus these algorithms are well suited for common use in DMLC beam delivery.
Delivered of IMRT with MLC in the step-and-shoot mode uses multiple static MLC segments to achieve intensity modulation. The sides of each leaf of a MLC have a protruding tongue or a step on one side that fits into a similar groove of the adjacent leaf. This results in different radiological path lengths across different parts of the leaves. Galvin et al. (1993a) first described that the different radiological path lengths manifest themselves as varying doses in a plane perpendicular to the leaf motion. The low dose region between two adjacent leaves was classified as the tongue-and-groove effect. In an IMRT treatment using an MLC, the tongue-and-groove effect occurs when the tongue, or the groove or both for the most time during treatment delivery cover the overlapping region between two adjacent pairs of leaves. As pointed out by many investigators, the tongue-and-groove arrangement always results in underdosages of as much as 10-25% in the treatment fields in both static and dynamic multileaf collimation (Galvin et al. 1993a, Galvin et al. 1993b, Chui et al. 1994, Mohan 1995, Wang et al. 1996, Sykes and Williams 1998).

Several recent publications (van Santvoort and Heijmen 1996, Webb et al. 1997, Convery and Webb 1998, Dirkx et al. 1998, Xia and Verhey 1998) have shown that the tongue-and-groove effect can be significantly reduced by synchronization of the leaves. However, the cost of leaf synchronization is usually an increase in the total number of sub fields and monitor units. van Santvoort and Heijmen (1996) propose an algorithm to eliminate tongue-and-groove effects for DMLC treatment plans. Although they note that their algorithm increases the number of monitor units, they do not examine the optimality or suboptimality of the plans they obtain. We recently published a paper (Kamath et al. 2003) that gave mathematical formalisms and rigorous proofs of leaf sequencing algorithms for segmental multileaf collimation, which maximize MU efficiency. We proved that our leaf sequencing algorithms that explicitly account for minimum leaf separation obtain feasible unidirectional solutions that are optimal. We now extend that work to develop algorithms.
that explicitly account for leaf interdigitation and the tongue-and-groove effect and are optimal in MU efficiency for unidirectional schedules. We show also that the algorithm of van Santvoort and Heijmen (1996) obtains optimal dynamic multileaf collimation treatment schedules.

4.1 Algorithm with Interdigitation and Tongue-and-Groove Constraints

4.1.1 Tongue-and-Groove Underdosage Effect

Figure 4–1 shows a beams-eye view of the region to be treated by two adjacent leaf pairs, \( t \) and \( t+1 \). Consider the shaded rectangular areas \( A_t(x_i) \) and \( A_{t+1}(x_i) \) that require exactly \( I_t(x_i) \) and \( I_{t+1}(x_i) \) MUs to be delivered, respectively. The tongue-and-groove overlap area between the two leaf pairs over the sample point \( x_i \), \( A_{t,t+1}(x_i) \), is colored black. Let the amount of MUs delivered in \( A_{t,t+1}(x_i) \) be \( I_{t,t+1}(x_i) \). Ignoring leaf transmission, the following lemma is a consequence of the fact that \( A_{t,t+1}(x_i) \) is exposed only when both \( A_t(x_i) \) and \( A_{t+1}(x_i) \) are exposed.

\[
\text{Lemma 21} \quad I_{t,t+1}(x_i) \leq \min\{I_t(x_i), I_{t+1}(x_i)\}, \quad 0 \leq i \leq m, \quad 1 \leq t < n.
\]

Schedules in which \( I_{t,t+1}(x_i) = \min\{I_t(x_i), I_{t+1}(x_i)\} \) are said to be free of tongue-and-groove underdosage effects.

Unless treatment schedules are carefully designed, it is possible that \( I_{t,t+1}(x_i) \ll \min\{I_t(x_i), I_{t+1}(x_i)\} \) for some \( i \) and \( t \). For example, in a schedule in which \( I_{tr}(x_i) = 30 \), \( I_{tl}(x_i) = 50 \), \( I_{(t+1)r}(x_i) = 50 \) and \( I_{(t+1)l}(x_i) = 60 \), we have \( I_{t,t+1}(x_i) = I_{tl}(x_i) - I_{(t+1)r}(x_i) = 50 - 50 = 0 \). Note that in this case, \( \min\{I_t(x_i), I_{t+1}(x_i)\} = I_{(t+1)l}(x_i) - I_{tl}(x_i) = 60 - 50 = 10 \). It is clear from this example that \( I_{t,t+1}(x_i) \) could be 0 even when \( \min\{I_t(x_i), I_{t+1}(x_i)\} \) is arbitrarily large.
4.1.2 Algorithms

Kamath et al. (2003) present an algorithm that generates a schedule that satisfies inter-pair minimum separation constraint. The schedule is optimal in therapy time. However, it does not account for the tongue-and-groove effect. In this section, we present two algorithms. Algorithm TONGUEANDGROOVE generates minimum therapy time unidirectional schedules that are free of tongue-and-groove underdosage and maybe used for MLCs that do not have a interdigitation constraint. Algorithm TONGUEANDGROOVE-ID generates minimum therapy time unidirectional schedules that are free of tongue-and-groove underdosage while simultaneously satisfying the interdigitation constraint and is for MLCs that have an interdigitation constraint.

The following lemma provides a necessary and sufficient condition for a unidirectional schedule to be free of tongue-and-groove underdosage effects.

**Lemma 22** A unidirectional schedule is free of tongue-and-groove underdosage effects if and only if,

(a) $I_t(x_i) = 0$ or $I_{t+1}(x_i) = 0$, or

(b) $I_{tr}(x_i) \leq I_{(t+1)r}(x_i) \leq I_{(t+1)l}(x_i) \leq I_{tl}(x_i)$, or

(c) $I_{(t+1)r}(x_i) \leq I_{tr}(x_i) \leq I_{tl}(x_i) \leq I_{(t+1)l}(x_i)$,

$0 \leq i \leq m, 1 \leq t < n$.

**Proof:** It is easy to see that any schedule that satisfies the above conditions is free of tongue-and-groove underdosage effects. So what remains is for us to show that every schedule that is free of tongue-and-groove underdosage effects satisfies the above conditions. Consider any such schedule. If condition (a) is satisfied at every $i$ and $t$, the proof is complete. So assume $i$ and $t$ such that $I_t(x_i) \neq 0$ and $I_{t+1}(x_i) \neq 0$ exist. We need to show that either (b) or (c) is true for this value of $i$ and $t$. Since the schedule is free of tongue-and-groove effects,

$$I_{t,t+1}(x_i) = \min\{I_t(x_i), I_{t+1}(x_i)\} > 0$$ (4.1)

From the unidirectional constraint, it follows that $A_{t,t+1}(x_i)$ first gets exposed when both right leaves pass $x_i$, and it remains exposed till the first of the left leaves passes $x_i$. Further, if a left leaf passes $x_i$ before a neighboring right leaf passes $x_i$, $A_{t,t+1}(x_i)$ is not exposed at
all. So,

\[ I_{t,t+1}(x_i) = \max\{0, I_{(t,t+1)l}(x_i) - I_{(t,t+1)r}(x_i)\} \quad (4.2) \]

where \( I_{(t,t+1)r}(x_i) = \max\{I_{tr}(x_i), I_{(t+1)r}(x_i)\} \) and \( I_{(t,t+1)l}(x_i) = \min\{I_{tl}(x_i), I_{(t+1)l}(x_i)\} \).

From 4.1 and 4.2, it follows that

\[ I_{t,t+1}(x_i) = I_{(t,t+1)l}(x_i) - I_{(t,t+1)r}(x_i) \quad (4.3) \]

Consider the case \( I_t(x_i) \geq I_{t+1}(x_i) \). Suppose that \( I_{tr}(x_i) > I_{(t+1)r}(x_i) \). It follows that \( I_{(t,t+1)r}(x_i) = I_{tr}(x_i) \) and \( I_{(t,t+1)l}(x_i) = I_{(t+1)l}(x_i) \). Now from 4.3, we get

\[ I_{t,t+1}(x_i) = I_{(t+1)l}(x_i) - I_{tr}(x_i) \]
\[ < I_{(t+1)l}(x_i) - I_{(t+1)r}(x_i) \]
\[ = I_{t+1}(x_i) \]
\[ \leq I_t(x_i) \]

So \( I_{t,t+1}(x_i) < \min\{I_t(x_i), I_{t+1}(x_i)\} \), which contradicts 4.1. So

\[ I_{tr}(x_i) \leq I_{(t+1)r}(x_i) \quad (4.4) \]

Now, suppose that \( I_{tl}(x_i) < I_{(t+1)l}(x_i) \). From \( I_t(x_i) \geq I_{t+1}(x_i) \), it follows that \( I_{(t,t+1)l}(x_i) = I_{tl}(x_i) \) and \( I_{(t,t+1)r}(x_i) = I_{(t+1)r}(x_i) \). Hence, from 4.3, we get

\[ I_{t,t+1}(x_i) = I_{tl}(x_i) - I_{(t+1)r}(x_i) \]
\[ < I_{(t+1)l}(x_i) - I_{(t+1)r}(x_i) \]
\[ = I_{t+1}(x_i) \]
\[ \leq I_t(x_i) \]

So \( I_{t,t+1}(x_i) < \min\{I_t(x_i), I_{t+1}(x_i)\} \), which contradicts 4.1. So

\[ I_{tl}(x_i) \geq I_{(t+1)l}(x_i) \quad (4.5) \]

From 4.4 and 4.5, we can conclude that when \( I_t(x_i) \geq I_{t+1}(x_i) \), (b) is true. Similarly one can show that when \( I_{t+1}(x_i) \geq I_t(x_i) \), (c) is true.

Lemma 22 is equivalent to saying that the time period for which a pair of leaves (say pair \( t \)) exposes the region \( A_{t,t+1}(x_i) \) is completely contained by the time period for which
pair $t + 1$ exposes region $A_{t,t+1}(x_i)$, or vice versa, whenever $I_t(x_i) \neq 0$ and $I_{t+1}(x_i) \neq 0$. Note that if either $I_t(x_i)$ or $I_{t+1}(x_i)$ is zero the containment is not necessary. We will refer to the necessary and sufficient condition of Lemma 22 as the **tongue-and-groove constraint condition**. Schedules that satisfy this condition will be said to satisfy the tongue-and-groove constraint. van Santvoort and Heijmen (1996) present an algorithm that generates schedules that satisfy the tongue-and-groove constraint for DMLC.

Xia and Verhey (1998) claim that every schedule that violates the interdigitation constraint also violates the tongue-and-groove constraint. We demonstrate with a counterexample that this is not necessarily the case. The intensity matrix shown in Figure 4–2(a) can be exposed in a single segment as shown in Figure 4–2(b). The segment is free of tongue-and-groove constraint violations, while it clearly violates the interdigitation constraint.

**Figure 4–2:** Counterexample. The intensity matrix shown in (a) can be treated using a single segment with 50 MUs as shown in (b). Areas shaded dark are covered by left leaves and those shaded light are covered by right leaves. Areas not shaded are exposed. Interdigitation constraint violation occurs though there is no tongue-and-groove violation.

**Elimination of tongue-and-groove effect.** Note that the schedule generated by Algorithm MULTIPAIR may violate the tongue-and-groove constraint. If the schedule has no tongue-and-groove constraint violations, it is the desired optimal schedule. If there are violations in the schedule, we eliminate all violations of the tongue-and-groove constraint starting from the left end, i.e., from $x_0$. To eliminate the violations, we modify those plans of the schedule that cause the violations. We scan the schedule from $x_0$ along the positive $x$ direction looking for the least $x_w$ at which there exist leaf pairs $u, t, t \in \{u - 1, u + 1\}$, that violate the constraint at $x_w$. After rectifying the violation at $x_w$, we look for other violations. Since the process of eliminating a violation at $x_w$, may at times, lead to new
violations at \( x_w \), we need to search afresh from \( x_w \) every time a modification is made to the schedule. However, we will prove a bound of \( O(n) \) on the number of violations that can occur at \( x_w \). After eliminating all violations at a particular sample point, \( x_w \), we move to the next point, i.e., we increment \( w \) and look for possible violations at the new point. We continue the scanning and modification process until no tongue-and-groove constraint violations exist. Algorithm TONGUEANDGROOVE (Figure 4–3) outlines the procedure.

Algorithm TONGUEANDGROOVE

1. \( x = x_0 \)
2. While (there is a tongue-and-groove violation) do
3. Find the least \( x_w, x_w \geq x \), such that there exist leaf pairs \( u, u + 1 \), that violate the tongue-and-groove constraint at \( x_w \).
4. Modify the schedule to eliminate the violation between leaf pairs \( u \) and \( u + 1 \).
5. \( x = x_w \)
6. End While

Figure 4–3: Obtaining a schedule under the tongue-and-groove constraint

Let \( M = ((I_{1l}, I_{1r}), (I_{2l}, I_{2r}), \ldots, (I_{nl}, I_{nr})) \) be the schedule generated by Algorithm MULTIPAIR for the desired intensity profile.

Let \( N(p) = ((I_{tlp}, I_{trp}), (I_{2lp}, I_{2rp}), \ldots, (I_{nlp}, I_{nrp})) \) be the schedule obtained after Step 4 of Algorithm TONGUEANDGROOVE is applied \( p \) times to the input schedule \( M \). Note that \( M = N(0) \).

To illustrate the modification process we use examples. To make things easier, we only show two neighboring pairs of leaves. Suppose that the \((p + 1)\)th violation occurs between the leaves of pair \( u \) and pair \( t = u + 1 \) at \( x_w \). Note that \( I_{tlp}(x_w) \neq I_{ulp}(x_w) \), as otherwise, either (b) or (c) of Lemma 22 is true. In case \( I_{tlp}(x_w) > I_{ulp}(x_w) \), swap \( u \) and \( t \). Now, we have \( I_{tlp}(x_w) < I_{ulp}(x_w) \). In the sequel, we refer to these \( u \) and \( t \) values as the \( u \) and \( t \) of Algorithm TONGUEANDGROOVE. From Lemma 22 and the fact that a violation has occurred, it follows that \( I_{trp}(x_w) < I_{urp}(x_w) \). To remove this tongue-and-groove constraint violation, we modify \((I_{tlp}, I_{trp})\). The other profiles of \( N(p) \) are not modified.
The new plan for pair $t$, $(I_{tl(p+1)}, I_{tr(p+1)})$ is as defined below. If $I_{ulp}(x_w) - I_{tlp}(x_w) \leq I_{urp}(x_w) - I_{trp}(x_w)$, then

$$I_{tl(p+1)}(x) = \begin{cases} I_{tlp}(x) & x_0 \leq x < x_w \\ I_{tlp}(x) + \Delta I & x_w \leq x \leq x_m \end{cases}$$

(4.6)

where $\Delta I = I_{ulp}(x_w) - I_{tlp}(x_w)$. $I_{tr(p+1)}(x) = I_{tl(p+1)}(x) - I_t(x)$, where $I_t(x)$ is the target profile to be delivered by the leaf pair $t$.

Otherwise,

$$I_{tr(p+1)}(x) = \begin{cases} I_{trp}(x) & x_0 \leq x < x_w \\ I_{trp}(x) + \Delta I' & x_w \leq x \leq x_m \end{cases}$$

(4.7)

where $\Delta I' = I_{urp}(x_w) - I_{trp}(x_w)$. $I_{tl(p+1)}(x) = I_{tr(p+1)}(x) + I_t(x)$, where $I_t(x)$ is the target profile to be delivered by the leaf pair $t$.

The former case is illustrated in Figure 4–4 and the latter is illustrated in Figure 4–5. Note that our strategy for plan modification is similar to that used by van Santvoort and Heijmen (1996) to eliminate a tongue-and-groove violation for dynamic multileaf collimator plans.

![Figure 4–4: Tongue-and-groove constraint violation: case1](image_url)
Since \((I_{tl(p+1)}, I_{tr(p+1)})\) differs from \((I_{ulp}, I_{urp})\) for \(x \geq x_w\) there is a possibility that \(N(p+1)\) is involved in tongue-and-groove violations for \(x \geq x_w\). Since none of the other leaf profiles are changed from those of \(N(p)\) no tongue-and-groove constraint violations are possible in \(N(p+1)\) for \(x < x_w\). One may also verify that since \(I_{d0}\) and \(I_{tr0}\) are non-decreasing functions of \(x\), so also are \(I_{ulp}\) and \(I_{urp}\), \(p > 0\).

**Lemma 23** Let \(F = ((I'_{l1}, I'_{r1}), (I'_{l2}, I'_{r2}), \ldots, (I'_{ln_l}, I'_{rn_r}))\) be any unidirectional schedule for the desired profile that satisfies the tongue-and-groove constraint. Let \(S(p)\), be the following assertions.

(a) \(I'_{il}(x) \geq I_{ulp}(x), 0 \leq i \leq n, x_0 \leq x \leq x_m\)

(b) \(I'_{ir}(x) \geq I_{urp}(x), 0 \leq i \leq n, x_0 \leq x \leq x_m\)

\(S(p)\) is true for \(p \geq 0\).

**Proof:** The proof is by induction on \(p\).

1. Consider the base case, \(p = 0\). From Corollary 1 and the fact that the plans \((I_{d0}, I_{r0}), 0 \leq i \leq n\), are generated using Algorithm SINGLEPAIR, it follows that \(S(0)\) is true.
2. Assume \( S(p) \) is true. Suppose Algorithm TONGUEANDGROOVE finds a next violation and modifies the schedule \( N(p) \) to \( N(p + 1) \). Suppose that the next violation occurs between leaf pairs \( u \) and \( t \), \( t \in \{ u - 1, u + 1 \} \). Hence, \( I_{tlp}(x_w) < I_{ulp}(x_w) \). We modify pair \( t' \)'s plan for \( x \geq x_w \), to eliminate the violation. All other plans in the schedule remain unaltered. Therefore, to establish \( S(p + 1) \) it suffices to prove that

\[
I'_{tl}(x) \geq I_{tl(p+1)}(x), x \geq x_w
\]  

(4.8)

\[
I'_{tr}(x) \geq I_{tr(p+1)}(x), x \geq x_w
\]

(4.9)

We need prove only one of these two relationships since \( I'_{tl}(x) - I'_{tr}(x) = I_{tl(p+1)}(x) - I_{tr(p+1)}(x), x_0 \leq x \leq x_m \) (i.e., \( I'_{tl}(x) - I_{tl(p+1)}(x) = I'_{tr}(x) - I_{tr(p+1)}(x) \)). We now consider pair \( t' \)'s plan for \( x \geq x_w \) and show that Equation 4.8 is always true. This, in turn, implies that \( S(p + 1) \) is true whenever \( S(p) \) is true and hence completes the proof.

Suppose that \( I_{ulp}(x_w) - I_{tlp}(x_w) \leq I_{urp}(x_w) - I_{trp}(x_w) \). Then, \( I_{ulp}(x_w) - I_{urp}(x_w) \leq I_{ulp}(x_w) - I_{trp}(x_w) \), i.e., \( I_u(x_w) \leq I_t(x_w) \). Clearly, in a schedule \( F \), which is free of tongue-and-groove violation between pairs \( u \) and \( t \) at \( x_w \), only the ordering \( I'_{tr}(x_w) \leq I'_{ur}(x_w) \leq I'_{ul}(x_w) \) is possible (refer Lemma 22) in this scenario (the exception being when \( I_u(x_w) = I_t(x_w) \), in which case all the quantities in the ordering are equal). From this ordering, \( I'_{ul}(x_w) \geq I'_{ul}(x_w) \). From the induction hypothesis, \( I'_{ul}(x_w) \geq I_{ulp}(x_w) = I_{ulp(p+1)}(x_w) \). From Equation 4.6, \( I_{ulp(p+1)}(x_w) = I_{ulp}(x_w) = I_{ulp(p+1)}(x_w) \). Hence, \( I'_{ul}(x_w) \geq I_{ulp(p+1)}(x_w) \) when \( I_{ulp}(x_w) - I_{tlp}(x_w) \leq I_{urp}(x_w) - I_{trp}(x_w) \). A symmetric argument can be presented to show that \( I'_{ul}(x_w) \geq I_{tl(p+1)}(x_w) \) when \( I_{ulp}(x_w) - I_{tlp}(x_w) > I_{urp}(x_w) - I_{trp}(x_w) \). So \( I'_{ul}(x_w) \geq I_{tl(p+1)}(x_w) \).

It remains to be proved that \( I'_{ul}(x_i) \geq I_{tl(p+1)}(x_i) \), \( w < i \leq m \). Suppose for a contradiction that \( \exists v > w, I'_{ul}(x_v) < I_{tl(p+1)}(x_v) \). Let \( \Delta I'' = I'_{ul}(x_w) - I'_{tl}(x_w) \). Note that \( I'_{ul}(x_w) \geq I_{ulp(p+1)}(x_w) \) and so \( \Delta I'' \geq I_{ulp(p+1)}(x_w) - I_{tl0}(x_w) = I_{ulp(p+1)}(x_v) - I_{tl0}(x_v) \) (from the working of Algorithm TONGUEANDGROOVE). Define a new plan \( (I''_{ul}, I''_{tr}) \) as follows:
\[
I''_{tl}(x_i) = \begin{cases} 
I_{t0}(x_i) & i < w \\
I'_{tl}(x_i) - \Delta I'' & w \leq i \leq m 
\end{cases}
\]

\[
I''_{tr}(x_i) = \begin{cases} 
I_{tr0}(x_i) & i < w \\
I'_{tr}(x_i) - \Delta I'' & w \leq i \leq m 
\end{cases}
\]

Note that \(I''_{tl}(x_w) = I'_{tl}(x_w) - \Delta I'' = I_{t0}(x_w) \geq I_{tl0}(x_{w-1}) = I''_{tl}(x_{w-1})\). Similarly, \(I''_{tr}(x_w) \geq I''_{tr}(x_{w-1})\). So \((I''_{tl}, I''_{tr})\) is a plan for \(t\). Also, \(I''_{tl}(x_v) = I'_{tl}(x_v) - \Delta I'' \leq I''_{tl}(x_v) - I_{tl}(p+1)(x_v) + I_{tl0}(x_v)\) (since \(\Delta I'' \geq I_{tl}(p+1)(x_v) - I_{tl0}(x_v)\) as explained above).

From this and our assumption that \(I'_{tl}(x_v) < I_{tl}(p+1)(x_v)\), it follows that \(I''_{tl}(x_v) < I_{tl0}(x_v)\). Since plan \((I_{tl0}, I_{tr0})\) was generated using Algorithm SINGLEPAIR, \(I''_{tl}(x_v) < I_{tl0}(x_v)\) violates Corollary 1. So our assumption was wrong and hence Equation 4.8 is always true.

**Elimination of tongue-and-groove effect and interdigitation.** As we have pointed out, the elimination of tongue-and-groove constraint violations does not guarantee elimination of interdigitation constraint violations. Therefore the schedule generated by Algorithm TONGUEANDGROOVE may not be free of interdigitation violations. The algorithm we propose for obtaining schedules that simultaneously satisfy both constraints, Algorithm TONGUEANDGROOVE-ID, is similar to Algorithm TONGUEANDGROOVE. The only difference between the two algorithms lies in the definition of the constraint condition. To be precise we make the following definition.

**Definition 1** A unidirectional schedule is said to satisfy the tongue-and-groove-id constraint if

(a) \(I_{tr}(x_i) \leq I_{(t+1)r}(x_i) \leq I_{(t+1)}(x_i) \leq I_{tl}(x_i)\), or

(b) \(I_{(t+1)r}(x_i) \leq I_{tr}(x_i) \leq I_{tl}(x_i) \leq I_{(t+1)}(x_i)\),

for \(0 \leq i \leq m, 1 \leq t < n\).

The only difference between this constraint and the tongue-and-groove constraint is that this constraint enforces condition (a) or (b) above to be true at all sample points \(x_i\) including those at which \(I_t(x_i) = 0\) and/or \(I_{t+1}(x_i) = 0\).
Lemma 24 A schedule satisfies the tongue-and-groove-id constraint iff it satisfies the tongue-and-groove constraint and the interdigitation constraint.

Proof: It is obvious that the tongue-and-groove-id constraint subsumes the tongue-and-groove constraint. If a schedule has a violation of the interdigitation constraint, \( \exists i, t, I_{(t+1)}l(x_i) < I_{tr}(x_i) \) or \( I_{tl}(x_i) < I_{(t+1)}r(x_i) \). From Definition 1, it follows that schedules that satisfy the tongue-and-groove-id constraint do not violate the interdigitation constraint. Therefore a schedule that satisfies the tongue-and-groove-id constraint satisfies the tongue-and-groove constraint and the interdigitation constraint.

For the other direction of the proof, consider a schedule \( O \) that satisfies the tongue-and-groove constraint and the interdigitation constraint. From the fact that \( O \) satisfies the tongue-and-groove constraint and from Lemma 22 and Definition 1, it only remains to be proved that for schedule \( O \),

\[
\text{a) } I_{tr}(x_i) \leq I_{(t+1)r}(x_i) \leq I_{(t+1)}l(x_i) \leq I_{tl}(x_i), \quad \text{or} \\
\text{b) } I_{(t+1)r}(x_i) \leq I_{tr}(x_i) \leq I_{tl}(x_i) \leq I_{(t+1)}l(x_i),
\]

whenever \( I_t(x_i) = 0 \) or \( I_{t+1}(x_i) = 0 \), \( 0 \leq i \leq m \), \( 1 \leq t < n \).

When \( I_t(x_i) = 0 \),

\[
I_{tl}(x_i) = I_{tr}(x_i) \tag{4.10}
\]

Since \( O \) satisfies the interdigitation constraint,

\[
I_{tr}(x_i) \leq I_{(t+1)}l(x_i) \tag{4.11}
\]

and

\[
I_{(t+1)r}(x_i) \leq I_{tl}(x_i) \tag{4.12}
\]

From Equations 4.10, 4.11 and 4.12, we get \( I_{(t+1)r}(x_i) \leq I_{tr}(x_i) = I_{tl}(x_i) \leq I_{(t+1)}l(x_i) \). So (b) is true whenever \( I_t(x_i) = 0 \). Similarly, (a) is true whenever \( I_{t+1}(x_i) = 0 \). Therefore, \( O \) satisfies the tongue-and-groove-id constraint.

Algorithm TONGUEANDGROOVE-ID finds violations of the tongue-and-groove-id constraint from left to right in exactly the same manner in which Algorithm TONGUEANDGROOVE detects tongue-and-groove violations. Also, the violations are eliminated as before, i.e., as prescribed by Equations 4.6 and 4.7 and illustrated in Figures 4-4 and 4-5.
respectively. Algorithm TONGUEANDGROOVE-ID is shown in Figure 4–6. All notation
used in the algorithm and the related discussion in the remainder of Section 4.1.2 is also
the same as that used in Section 4.1.2 and corresponds directly to the usage in Algorithm
TONGUEANDGROOVE.

Algorithm TONGUEANDGROOVE-ID
1. \( x = x_0 \)
2. While (there is a tongue-and-groove-id violation) do
3. Find the least \( x_w, x_w \geq x \), such that there exist leaf pairs \( u, u + 1 \), that violate the
tongue-and-groove-id constraint at \( x_w \).
4. Modify the schedule to eliminate the violation between leaf pairs \( u \) and \( u + 1 \).
5. \( x = x_w \)
6. End While

Figure 4–6: Obtaining a schedule under both the constraints

Lemma 25 Let \( F = ((I_{l1}', I_{r1}'), (I_{l2}', I_{r2}'), \ldots, (I_{nl}', I_{nr}')) \) be any unidirectional schedule for
the desired profile that satisfies the tongue-and-groove-id constraint. Let \( S(p) \), be the fol-
lowing assertions.

\( (a) \quad I_{il}'(x) \geq I_{ilp}(x), \ 0 \leq i \leq n, x_0 \leq x \leq x_m \)

\( (b) \quad I_{ir}'(x) \geq I_{irp}(x), \ 0 \leq i \leq n, x_0 \leq x \leq x_m \)

\( S(p) \) is true for \( p \geq 0 \).

Proof: The proof is by induction on \( p \).

1. Consider the base case, \( p = 0 \). From Corollary 1 and the fact that the plans
\((I_{il0}, I_{ir0}), 0 \leq i \leq n, \) are generated using Algorithm SINGLEPAIR, it follows that
\( S(0) \) is true.

2. Assume \( S(p) \) is true. Suppose Algorithm TONGUEANDGROOVE-ID finds a next
violation and modifies the schedule \( N(p) \) to \( N(p+1) \). Suppose that the next violation
occurs between leaf pairs \( u \) and \( t, t \in \{u - 1, u + 1\} \). As in the proof of Lemma 23,
we only need prove either Equation 4.8 or Equation 4.9 to complete this proof. We
complete the proof for the following three cases that are exhaustive.

case 1: \( I_t(x_w) \neq 0 \) and \( I_u(x_w) \neq 0 \).

The remainder of the proof for this case is the same as that of Lemma 23.

case 2: \( I_t(x_w) = 0 \).

In this case, \( I_{ilp}(x_w) = I_{irp}(x_w) \). Since \( I_{ilp}(x_w) \geq I_{irp}(x_w) \), we have \( I_{ilp}(x_w) - \)
\( I_{tp}(x_w) \geq I_{urp}(x_w) - I_{trp}(x_w) \). The modification prescribed by Equation 4.7 is applicable. Note that if \( I_{urp}(x_w) - I_{trp}(x_w) = I_{ulp}(x_w) - I_{tlp}(x_w) \), Equation 4.6 is the same as Equation 4.7. In particular,

\[
I_{tr(p+1)}(x_w) = I_{trp}(x_w) + I_{urp}(x_w) - I_{trp}(x_w) = I_{urp}(x_w)
\] (4.13)

Since \( I_t(x_w) = 0 \),

\[
I_{tr(p+1)}(x_w) = I_{tl(p+1)}(x_w)
\] (4.14)

From Equations 4.13 and 4.14,

\[
I_{urp}(x_w) = I_{tl(p+1)}(x_w)
\] (4.15)

Since \( F \) satisfies the interdigitation constraint, the left leaf of pair \( t \) does not pass \( x_w \) before the right leaf of pair \( u \) passes \( x_w \). So,

\[
I'_{tl}(x_w) \geq I'_{ur}(x_w)
\] (4.16)

From \( S(p) \) and Equation 4.15, we get,

\[
I'_{ur}(x_w) \geq I_{urp}(x_w) = I_{tl(p+1)}(x_w)
\] (4.17)

Equations 4.16 and 4.17 yield

\[
I'_t(x_w) - I_{tl(p+1)}(x_w) \geq 0
\] (4.18)

Lemma 2b implies,

\[
I'_t(x) - I_t(x) \geq I'_t(x_w) - I_t(x_w), x \geq x_w
\] (4.19)

Subtracting \( I_{tl(p+1)}(x) \) from Equation 4.19, and rearranging terms we get

\[
I'_t(x) - I_{tl(p+1)}(x) \geq I'_t(x_w) - I_t(x_w) + I'_t(x) - I_{tl(p+1)}(x), x \geq x_w
\] (4.20)

From Equations 4.6 and 4.7 and the working of Algorithm TONGUEAND-GROOVE - ID, it follows that

\[
I_{tl(p+1)}(x) - I_t(x) = I_{tl(p+1)}(x_w) - I_{tl}(x_w), x \geq x_w
\] (4.21)
From Equations 4.20, 4.21 and 4.18, we get

$$I_{tl}'(x) - I_{tl(p+1)}(x) \geq I_{tl}'(x_w) - I_{tl(p+1)}(x_w) \geq 0, x \geq x_w$$

(4.22)

Therefore,

$$I_{tl}'(x) \geq I_{tl(p+1)}(x), x \geq x_w$$

(4.23)

case 3: $I_u(x_w) = 0$.

The proof is similar to that of case 2.

4.1.3 Efficient Implementation of the Algorithms

In the remainder of this section we will use ‘algorithm’ to mean Algorithm TONGUEANDGROOVE or Algorithm TONGUEANDGROOVE-ID and ‘violation’ to mean tongue-and-groove constraint violation or tongue-and-groove-id constraint violation (depending on which algorithm is considered) unless explicitly mentioned.

The execution of the algorithm starts with schedule $M$ at $x = x_0$ and sweeps to the right, eliminating violations from the schedule along the way. The modifications applied to eliminate a violation at $x_w$, prescribed by Equations 4.6 and 4.7, modify one of the violating profiles for $x \geq x_w$. From the unidirectional nature of the sweep of the algorithm, it is clear that the modification of the profile for $x > x_w$ can have no consequence on violations that may occur at the point $x_w$. Therefore it suffices to modify the profile only at $x_w$ at the time the violation at $x_w$ is detected. The modification can be propagated to the right as the algorithm sweeps. This can be done by using an $(n \times m)$ matrix $A$ that keeps track of the amount by which the profiles have been raised. $A(j, k)$ denotes the cumulative amount by which the $j$th leaf pair profiles have been raised at sample point $x_k$ from the schedule $M$ generated using Algorithm MULTIPAIR. When the algorithm has eliminated all violations at each $x_w$, it moves to $x_{w+1}$ to look for possible violations. It first sets the $(w+1)$th column of the modification matrix equal to the $w$th column to reflect rightward propagation of the modifications. It then looks for and eliminates violations at $x_{w+1}$ and so on.
The process of detecting the violations at $x_w$ merits further investigation. We show that if one carefully selects the order in which violations are detected and eliminated, the number of violations at each $x_w$, $0 \leq w \leq m$ will be $O(n)$.

**Lemma 26** The algorithm can be implemented such that $O(n)$ violations occur at each $x_w$, $0 \leq w \leq m$.

**Proof:** The bound is achieved using a two pass scheme at $x_w$. In pass one we check adjacent leaf pairs $(1, 2), (2, 3), \ldots, (n-1, n)$, in that order, for possible violations at $x_w$. In pass two, we check for violations in the reverse order, i.e., $(n-1, n), (n-2, n-1), \ldots, (1, 2)$. So each set of adjacent pairs $(i, i+1), 1 \leq i < n$ is checked exactly twice for possible violations. It is easy to see that if a violation is detected in pass one, either the profile of leaf pair $i$ or that of leaf pair $i + 1$ may be modified (raised) to eliminate the violation. However, in pass two only the profile of pair $i$ may be modified. This is because the profile of pair $i$ is not modified between the two times it is checked for violations with pair $i + 1$. The profile of pair $i + 1$, on the other hand, could have been modified between these times as a result of violations with pair $i + 2$. Therefore in pass two, only $i$ can be a candidate for $t$ (where $t$ is as explained in the algorithm) when pairs $(i, i + 1)$ are examined. From this it also follows that when pairs $(i-1, i)$ are subsequently examined in pass two, the profile of pair $i$ will not be modified. Since there is no violation between adjacent pairs $(1, 2), (2, 3), \ldots, (i, i + 1)$ at that time and none of these pairs is ever examined again, it follows that at the end of pass two there can be no violations between pairs $(i, i + 1)$, $1 \leq i < n$.

**Lemma 27** For the execution of the algorithm, the time complexity is $O(nm)$.

**Proof:** Follows from Lemma 26 and the fact that there are $m$ sample points.

**Theorem 20** (a) Algorithms TONGUEANDGROOVE and TONGUEANDGROOVE-ID terminate.

(b) The schedule generated by Algorithm TONGUEANDGROOVE is free of tongue-and-groove constraint violations and is optimal in therapy time for unidirectional schedules.

(c) The schedule generated by Algorithm TONGUEANDGROOVE-ID is free of interdigitation and tongue-and-groove constraint violations and is optimal in therapy time for unidirectional schedules.
Proof: (a) Lemma 27 provides a polynomial upper bound (\(O(n \times m)\)) on the complexity of Algorithms TONGUEANDGROOVE and TONGUEANDGROOVE-ID. The result follows from this.

(b) When Algorithm TONGUEANDGROOVE terminates, no tongue-and-groove violations remain. From this and Lemma 23, it follows that the schedule generated by Algorithm TONGUEANDGROOVE is optimal in therapy time for unidirectional schedules free of tongue-and-groove violations.

(c) When Algorithm TONGUEANDGROOVE-ID terminates, no tongue-and-groove-id violations remain and from Lemma 24 the final schedule satisfies the tongue-and-groove and interdigitation constraints. From this and Lemma 25, it follows that the schedule generated by Algorithm TONGUEANDGROOVE-ID is optimal in therapy time for unidirectional schedules free of both types of violations.

Theorem 21 The schedule generated by the algorithm of van Santvoort and Heijmen (1996) is free of interdigitation and tongue-and-groove constraint violations and is optimal in therapy time for unidirectional DMLC schedules with this property.

Proof: Similar to that of Theorem 20(c).

4.2 Experimental Validation

The algorithms were validated on a Varian 2100 C/D with 120-leaf MLC (Varian Medical Systems, Palo Alto, CA). The intensity maps of a 7-field head and neck plan from a commercial inverse treatment planning system (CORVUS 5.0, NOMOS Corporation, Cranberry, PA) were sequenced using Algorithm MULTIPAIR, which optimizes the MU efficiency, and Algorithm TONGUEANDGROOVE-ID, which eliminates the tongue-and-groove effect and interdigitation. The intensity maps have a bixel size of 1 cm x 1 cm and a 20% intensity step. Figure 4–7 shows the film measurement of the fluence maps of the AP field. The tongue-and-groove effect is readily seen in Figure 4–7(a), while it is completely eliminated in Figure 4–7(b) using Algorithm TONGUEANDGROOVE-ID. Table 4–1 compares the number of segments and the MU efficiencies of all three algorithms. The MU efficiency is defined as the ratio of the maximum fluence of intensity modulated field per MU to the fluence of an open field per MU. Compared to the leaf sequences with no
constraints, the consideration of tongue-and-groove correction increased both the number of segments and MUs, with an average increase of 21% and 19%, respectively, for the 7 intensity maps considered here. With the additional elimination of interdigitation, the increases were 25% and 24%, respectively. Examination of all the sub fields of the leaf sequences generated with Algorithm TONGUEANDGROOVE-ID verified that no interdigitation constraint has been violated.

![Figure 4–7: Film measurement of the AP field (field ID 1 in Table 1) of a seven-field head and neck plan. The optimized leaf sequences were generated without (Algorithm MULTIPAIR, (a)) and with tongue-and-groove-id correction (Algorithm TONGUEANDGROOVE-ID, (b)).](image)

### 4.3 Comparison with Algorithm of Que et al. (2004))

Recently a new algorithm to eliminate tongue-and-groove effects in step and shoot delivery has been proposed (Que et al. 2004). The algorithm of Que et al. (2004) is designed to eliminate tongue-and-groove effect. Although this algorithm eliminates tongue-and-groove effect on all 1000 random matrices tried in Que et al. (2004), no proof that the algorithm eliminates tongue-and-groove effect on all possible matrices has been provided. Further, it is not known whether or not the algorithm of Que et al. (2004) minimizes therapy time. We analyze the algorithm of Que et al. (2004) and show that it is always successful in eliminating tongue-and-groove effect; the generated leaf sequence is also free of interdigitation. We also perform a theoretical and experimental comparison of this algorithm with our algorithms (Kamath et al. (2004)).
Table 4–1: Comparison of the number of segments and MU efficiency of the three leaf sequencing algorithms (MULTIPAIR, TONGUEANDGROOVE and TONGUEANDGROOVE-ID) for 7 intensity maps of a head and neck treatment plan generated from a commercial treatment planning system. The percent increases in the number of segments and MUs for Algorithms TONGUEANDGROOVE and TONGUEANDGROOVE-ID with respect to Algorithm MULTIPAIR are also shown. The average percent increases in the number of segments are 21% and 25%, respectively. The average percent increases in the number of MUs are 19% and 24%, respectively.

<table>
<thead>
<tr>
<th>Field ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>MULTIPAIR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Segments</td>
<td>11</td>
<td>8</td>
<td>13</td>
<td>15</td>
<td>14</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>MU Efficiency</td>
<td>0.47</td>
<td>0.63</td>
<td>0.40</td>
<td>0.35</td>
<td>0.37</td>
<td>0.40</td>
<td>0.51</td>
</tr>
<tr>
<td>TONGUEANDGROOVE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Segments</td>
<td>14</td>
<td>10</td>
<td>15</td>
<td>21</td>
<td>14</td>
<td>13</td>
<td>11</td>
</tr>
<tr>
<td>MU Efficiency</td>
<td>0.37</td>
<td>0.51</td>
<td>0.35</td>
<td>0.25</td>
<td>0.37</td>
<td>0.40</td>
<td>0.40</td>
</tr>
<tr>
<td>% Segment # increase</td>
<td>27</td>
<td>25</td>
<td>15</td>
<td>40</td>
<td>0</td>
<td>30</td>
<td>10</td>
</tr>
<tr>
<td>% MU increase</td>
<td>26</td>
<td>24</td>
<td>15</td>
<td>37</td>
<td>0</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>TONGUEANDGROOVE-ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of Segments</td>
<td>14</td>
<td>11</td>
<td>16</td>
<td>21</td>
<td>14</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>MU Efficiency</td>
<td>0.37</td>
<td>0.47</td>
<td>0.33</td>
<td>0.25</td>
<td>0.37</td>
<td>0.37</td>
<td>0.37</td>
</tr>
<tr>
<td>% Segment # increase</td>
<td>27</td>
<td>38</td>
<td>23</td>
<td>40</td>
<td>0</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>% MU increase</td>
<td>26</td>
<td>36</td>
<td>22</td>
<td>37</td>
<td>0</td>
<td>7</td>
<td>38</td>
</tr>
</tbody>
</table>

4.3.1 Analysis of the Algorithm of Que et al. (2004)

In this section we analyze the algorithm of Que et al. (2004). They use the ‘sliding window’ method proposed by Bortfeld et al. (1994b) to generate a tentative segment. They then search through the right leaf positions to determine the leftmost right leaf position and position all right leaves at that position. This defines the first segment of the leaf sequence. The residual intensity matrix is calculated and the process is repeated. To obtain the ‘sliding window’ leaf sequence for each leaf pair, horizontal lines are drawn at unit intensity levels to intersect the intensity profile for that leaf pair. The left and right leaf positions are determined from these intersections and are sorted from left to right to give the final unidirectional leaf sequence. The process is repeated for all leaf pairs. For the case where the intensity levels in the map generated by the optimizer are integers, it is possible to show that the algorithms of Ma et al. (1998) and Kamath et al. (2003) (Algorithms SINGLEPAIR and MULTIPAIR for one and multiple leaf pairs respectively) will yield the
same leaf sequence as that obtained using the ‘sliding window’ method of Bortfeld et al. (1994b).

Note that the discrete intensity profile that needs to be delivered, \( I \), is output from the optimizer. Let \( n \) be the number of leaf pairs and \( m \) be the number of sample points for each leaf pair (i.e., for each row of the profile). We denote the rows of \( I \) by \( I_1, I_2, \ldots I_n \). Let \( I_t(x_i) \) denote the number of MUs that need to be delivered at sample point \( i \) (ith column) of leaf pair \( t \) (tth row).

**Lemma 28** The algorithm of Que et al. (2004) generates unidirectional schedules.

**Proof:** During each iteration, the next segment generated using the ‘sliding window’ method is such that the left leaves are positioned at the leftmost non-zero sample point (i.e., the least \( i \) such that \( I_t(x_i) > I_t(x_{i-1}) \), where \( I_t(x_{-1}) = 0 \)) for each row \( t \) in the residual matrix \( I \). The right leaves are positioned at the first columns of the respective rows where there is a decrease in intensity profile (i.e., the least \( j \) for which \( I_t(x_j) < I_t(x_{j-1}) \)). For example, for the single row intensity profile of Figure 4-8, the left leaf will be positioned at \( x_2 \) and the right leaf will be positioned at \( x_6 \). The algorithm of Que et al. (2004) repositions all right leaves to the position of the leftmost right leaf thus obtained. During the delivery of this segment, the intensity values in the matrix in the exposed areas decreases, while the other values remain unaltered. Therefore in the new residual matrix, the leftmost non-zero points of rows either remain at the same positions as in the residual matrix of the previous iteration (or the original intensity matrix for the first iteration) or they move to the right. So the left leaves cannot move to the left between successive segments. It is also easy to verify that there can be no index \( k \) such that in the updated residual intensity matrix \( x_k \) is to the left of the column of right leaf positions in the segment and \( I_t(x_k) < I_t(x_{k-1}) \) for some row. It follows that the right leaves cannot move to the left either. So the leaf movements are unidirectional and from left to right.

Definition 1 and Lemma 24 are from Kamath et al. (2004) and are used in the proof of Theorem 22.

**Theorem 22** The algorithm of Que et al. (2004) generates schedules free of tongue-and-groove effect and interdigitation.
Proof: Let $I'_l(x_i)$ and $I'_r(x_i)$, respectively, be the number of MUs delivered when the left and right leaves of pair $t$ pass $x_i$ in the schedule generated by the algorithm of Que et al. (2004). In the schedule generated, all right leaves pass point $x_i$, $0 \leq i \leq m$ (during their left to right movement) after exactly the same number of monitor units (MUs) are delivered. So $I'_l(x_i) = I'_r(x_{i+1})$, $0 \leq i \leq m$, $1 \leq t < n$. From this equality, Lemma 28, Definition 1, and Lemma 24, it follows that the schedule generated by the algorithm of Que et al. (2004) is free of tongue-and-groove effect and interdigitation.

Theorem 23 Let $T_{tng-id}$ be the optimal therapy time unidirectional leaf sequence that delivers an intensity profile $I$, while eliminating the tongue-and-groove effect and interdigitation. The therapy time for the schedule generated by the algorithm of Que et al. (2004) is at most $n \ast T_{tng-id}$, where $n$ is the number of involved leaf pairs. Further, $n \ast T_{tng-id}$ is a tight bound, i.e., there exist profiles $I$ for which the schedule generated by the algorithm of Que et al. (2004) requires a therapy time of $n \ast T_{tng-id}$.

Proof: Let $\Delta_{jr}(x_i)$ denote the amount of therapy time for which the right leaf of leaf pair $j$ stops at $x_i$ in the schedule obtained for $I$ using Algorithm MULTIPAIR (Kamath et al. 2003). The therapy time for the plan of leaf pair $j$ is the sum of times for which its right leaf stops at all sample points, which is $\sum_{i=0}^{m} \Delta_{jr}(x_i)$. The therapy time of the entire schedule, $T$, is the maximum of the therapy times of all leaf pairs, i.e., $T = \max_j \{ \sum_{i=0}^{m} \Delta_{jr}(x_i) \}$. Clearly, $T_{tng-id} \geq T$. In the schedule generated by the algorithm of Que et al. (2004), all the right leaves stop at each $x_i$ for the same amount of
time, say $\Delta'_r(x_i)$, which is equal to the maximum of the times for which a right leaf stops at $x_i$ in the schedule generated by Algorithm MULTIPAIR, i.e., $\Delta'_r(x_i) = \max_j \{\Delta_{jr}(x_i)\}$. The therapy time for the schedule generated by the algorithm of Que et al. (2004) is therefore $T'_{tng-id} = \sum_{i=0}^{m} \Delta'_r(x_i) = \sum_{i=0}^{m} \max_j \{\Delta_{jr}(x_i)\}$. Since each $\Delta_{jr}(x_i)$, $0 \leq i \leq m$, $1 \leq j < n$ can contribute a term to this expression for $T'_{tng-id}$ at most once, $T'_{tng-id} \leq \sum_{j=1}^{n} \sum_{i=0}^{m} \Delta_{jr}(x_i) \leq n \cdot \max_j \{\sum_{i=0}^{m} \Delta_{jr}(x_i)\} = n \cdot T \leq n \cdot T_{tng-id}$. Note that the algorithm of Kamath et al. (2004) generates schedules that are optimal in therapy time for unidirectional schedules. Hence the algorithm of Que et al. (2004) may generate schedules requiring up to $n$ times the therapy time required by the schedules generated by the algorithm of Kamath et al. (2004).

The above analysis assumes that leaf pairs are allowed to close within the field as defined by the collimator jaws. This is true for certain designs of MLCs. For MLCs with rounded leaf-end design, significant radiation transmission through the closed leaf pairs requires them to be moved under the collimator jaws. In this case, both the algorithms of Kamath et al. (2004) and Que et al. (2004) violate the interdigitation constraint, and only tongue and groove effect is eliminated.

Figure 4–9 shows an intensity map with 4 rows for which the algorithm of Que et al. (2004) requires $4 \cdot 20 = 80$ MUs. The map can be delivered using 20 MUs without violating the tongue-and-groove constraint and interdigitation constraint using Algorithm TONGUEANDGROOVE-ID (Kamath et al. 2004). The example can be generalized for $n$ rows.

```
<table>
<thead>
<tr>
<th>20</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
</tbody>
</table>
```

Figure 4–9: Worstcase example. This intensity map can be delivered using 20 MUs using Algorithm TONGUEANDGROOVE-ID (Kamath et al. 2004). The algorithm of Que et al. (2004) delivers this map using 80 MUs.
4.3.2 Results

We implemented Algorithms TONGUEANDGROOVE and TONGUEANDGROOVE-ID (Kamath et al. 2004) and the algorithm of Que et al. (2004). For performance comparison, we used two separate data sets. The first set consisted of three clinical IMRT plans with 7, 5 and 7 beams, respectively. The first two plans had a 20% fluence step and last plan had a 10% fluence step. Table 4.3.2 gives the total MUs and number of segments required for each of the 19 beams in the 3 clinical plans. On our clinical data set, the algorithm of Que et al. (2004) generated schedules with 2-4 times as many MUs and segments as did the algorithms of Kamath et al. (2004). The second data set consisted of 100,000 randomly generated $15 \times 15$ matrices. The intensity values in these matrices were random integers from 0 to 10. The average MUs and segments for schedules generated using the three algorithms for this set and their respective standard deviations are shown in Table 4.3.2. On this set, the algorithm of Que et al. (2004) generated schedules with about 2.5 times as many MUs and segments as did the algorithms of Kamath et al. (2004). Note that in both cases the number of MUs and segments in the schedules generated using Algorithm TONGUEANDGROOVE-ID (Kamath et al. 2004) are only slightly greater than in those generated using Algorithm TONGUEANDGROOVE (Kamath et al. 2004).

4.4 Conclusion

We have described mathematical formalism and rigorous proofs of leaf sequencing algorithms for segmental multileaf collimation, which maximize MU efficiency while completely eliminating the tongue-and-groove underdosage. Even though it has been shown that for a multiple field IMRT plan ($\geq 5$), the tongue-and-groove effect on the IMRT dose distribution is clinically insignificant (Deng et al. 2001) due to the smearing effect of individual fields, yet it still can be problematic for a small number of fields and for the patient setup with minimal uncertainty. Compared to the unconstrained leaf sequencing algorithms, the presented methods yield leaf sequences, which decreases the MU efficiency a little. But they completely overcome tongue-and-groove underdosages. One of the methods also eliminates leaf interdigitation. Most importantly, mathematical proofs show that these algorithms are optimal in MU efficiency for unidirectional schedules. We have also proved that the algorithm of Que et al. (2004) generates schedules that are free of the tongue-and-groove
Table 4–2: Number of MUs and segments generated for 19 clinical intensity modulated beams from 3 IMRT plans using algorithms A (Algorithm of Que et al. 2004), B (Algorithm TONGUEANDGROOVE (Kamath et al. 2004)) and C (Algorithm TONGUEANDGROOVE-ID (Kamath et al. 2004)). Beams 1-12 have a 20% fluence step, while beams 13-19 have a 10% fluence step.

<table>
<thead>
<tr>
<th>Beam number</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MUs</td>
<td>Segments</td>
<td>MUs</td>
</tr>
<tr>
<td>1</td>
<td>780</td>
<td>38</td>
<td>280</td>
</tr>
<tr>
<td>2</td>
<td>520</td>
<td>26</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>760</td>
<td>33</td>
<td>300</td>
</tr>
<tr>
<td>4</td>
<td>840</td>
<td>40</td>
<td>420</td>
</tr>
<tr>
<td>5</td>
<td>740</td>
<td>32</td>
<td>280</td>
</tr>
<tr>
<td>6</td>
<td>780</td>
<td>34</td>
<td>260</td>
</tr>
<tr>
<td>7</td>
<td>640</td>
<td>29</td>
<td>260</td>
</tr>
<tr>
<td>8</td>
<td>1500</td>
<td>74</td>
<td>380</td>
</tr>
<tr>
<td>9</td>
<td>860</td>
<td>43</td>
<td>240</td>
</tr>
<tr>
<td>10</td>
<td>1500</td>
<td>67</td>
<td>420</td>
</tr>
<tr>
<td>11</td>
<td>1660</td>
<td>78</td>
<td>420</td>
</tr>
<tr>
<td>12</td>
<td>840</td>
<td>39</td>
<td>280</td>
</tr>
<tr>
<td>13</td>
<td>880</td>
<td>78</td>
<td>280</td>
</tr>
<tr>
<td>14</td>
<td>1080</td>
<td>102</td>
<td>300</td>
</tr>
<tr>
<td>15</td>
<td>1070</td>
<td>90</td>
<td>310</td>
</tr>
<tr>
<td>16</td>
<td>1000</td>
<td>90</td>
<td>340</td>
</tr>
<tr>
<td>17</td>
<td>890</td>
<td>71</td>
<td>340</td>
</tr>
<tr>
<td>18</td>
<td>990</td>
<td>75</td>
<td>310</td>
</tr>
<tr>
<td>19</td>
<td>1010</td>
<td>84</td>
<td>330</td>
</tr>
</tbody>
</table>

Table 4–3: Average number of MUs and segments generated over a set of 100,000 random $15 \times 15$ matrices using algorithms A (Algorithm of Que et al. 2004), B (Algorithm TONGUEANDGROOVE (Kamath et al. 2004)) and C (Algorithm TONGUEANDGROOVE-ID (Kamath et al. 2004)). The respective standard deviations are also shown. The intensity values in the matrices were randomly generated integers from 0 to 10.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MUs</td>
<td>Segments</td>
<td>MUs</td>
</tr>
<tr>
<td>Average</td>
<td>114.3</td>
<td>111.6</td>
<td>47.5</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>6.1</td>
<td>6.0</td>
<td>3.4</td>
</tr>
</tbody>
</table>
effect and interdigitation. Our analysis shows that the algorithm of Que et al. generates schedules that may require up to \( n \) times the therapy time required by that for an optimal leaf sequence free of tongue-and-groove effect and interdigitation, where \( n \) is the number of involved leaf pairs. In experiments with clinical and randomly generated data sets we find that the algorithm of Que et al. (2004) generates schedules that require 2 to 4 times the therapy time required by the schedules generated by our algorithms.
The delivery of abutting sub-fields that result from the split of a large field often results in longer delivery times, poor MU efficiency, and field matching problems. Dogan et al. (2003) point out that the uncertainties in leaf and carriage positions cause errors in the delivered dose (hot or cold spots) along the match line of the abutting sub-fields. They observed dose differences of up to 10% along the field split line when the split line crossed through the center of the target for all the fields. The problem of dosimetric perturbation along the field split line has been addressed in several recent publications (Wu et al. 2000, Hong et al. 2002, Dogan et al. 2003). The solutions included automatic feathering of split-fields by modifying the split line position for each gantry position (Hong et al. 2002, Dogan et al. 2003) or by dynamically changing radiation intensity in the overlap region of the split fields. None of the field splitting techniques reported in the literature has addressed the issue of treatment delivery and MU efficiency. We believe that it is equally important to address this issue.

Our optimal field splitting algorithms with and without feathering may be integrated into our previously developed leaf sequencing algorithms to optimally account for interdigitation and tongue-and-groove effect of some multileaf collimators. We provide rigorous mathematical proofs that the proposed schemes for field splitting are optimal in MU efficiency. Experimental results show that our optimal field splitting algorithm without feathering reduces total MUs by up to 26% on clinical cases and up to 63% on synthetic cases compared to a commercial planning system that also splits fields without feathering.

5.2 Field Splitting Without Feathering

5.2.1 Optimal Field Splitting for One Leaf Pair

In this section we deviate slightly from our earlier notation and assume that the sample points are \(x_1, x_2, \ldots, x_m\) rather than \(x_0, x_1, \ldots, x_m\). All other notation remains unchanged.
Delivering a profile using One field. Let $I$ be the desired intensity profile. The problem of delivering the exact profile $I$ using a single field has been extensively studied. Ma et al. (1998) provide an $O(m)$ algorithm for the problem such that the therapy time of the solution is minimized, where $m$ is the number of sample points. Kamath et al. (2003) also describe the algorithm (Algorithm SINGLEPAIR) and give an alternate proof that it obtains a plan $(I_l, I_r)$ with optimal therapy time for $I$, where $I_l$ and $I_r$ denote the left and right leaf movement profiles, respectively. The optimal therapy time for $I$ is given by the following lemma.

Lemma 29 Let $inc_1, inc_2, \ldots, inc_q$ be the indices of the points at which $I(x_i)$ increases, i.e., $I(x_{inc_i}) > I(x_{inc_{i-1}})$. The therapy time for the plan $(I_l, I_r)$ generated by Algorithm SINGLEPAIR is $\sum_{i=1}^{q}[I(x_{inc_i}) - I(x_{inc_{i-1}})]$, where $I(x_{inc_1-1}) = 0$.

Algorithm SINGLEPAIR can be directly used to obtain plans when $I$ is deliverable using a single field. Let $l$ be the least index such that $I(x_l) > 0$ and let $g$ be the greatest index such that $I(x_g) > 0$. We will assume without loss of generality that $l = 1$. So the width of the profile is $g$ sample points, where $g$ can vary for different profiles. Assuming that the maximum allowable field width is $w$ sample points, $I$ is deliverable using one field if $g \leq w$; $I$ requires at least two fields for $g > w$; $I$ requires at least three fields for $g > 2w$.

The case where $g > 3w$ is not studied as it never arises in clinical cases. The objective of field splitting is to split a profile so that each of the resulting profiles is deliverable using a single field. Further, it is desirable that the total therapy time is minimized, i.e., the sum of optimal therapy times of the resulting profiles is minimized. We will call the problem of splitting the profile $I$ of a single leaf pair into 2 profiles each of which is deliverable using one field such that the sum of their optimal therapy times is minimized as the $S2$ (single pair 2 field split) problem. The sum of the optimal therapy times of the two resulting profiles is denoted by $S2(I)$. $S3$ and $S3(I)$ are defined similarly for splits into 3 profiles.

The problem $S1$ is trivial, since the input profile need not be split and is to be delivered using a single field. Note that $S1(I)$ is the optimal therapy time for delivering the profile $I$ in a single field. From Lemma 29 and the fact that the plan generated using Algorithm SINGLEPAIR is optimal in therapy time, $S1(I) = \sum_{i=1}^{q}[I(x_{inc_i}) - I(x_{inc_{i-1}})]$. 
Splitting a profile into two. Suppose that a profile \( I \) is split into two profiles. Let \( j \) be the index at which the profile is split. As a result, we get two profiles, \( P_j \) and \( S_j \). \( P_j(x_i) = I(x_i) \), \( 1 \leq i < j \), and \( P_j(x_i) = 0 \), elsewhere. \( S_j(x_i) = I(x_i) \), \( j \leq i \leq g \), and \( S_j(x_i) = 0 \), elsewhere. \( P_j \) is a left profile and \( S_j \) is a right profile of \( I \).

**Lemma 30** Let \( S_1(P_j) \) and \( S_1(S_j) \) be the optimal therapy times, respectively, for \( P_j \) and \( S_j \). Then \( S_1(P_j) + S_1(S_j) = S_1(I) + \hat{I}(x_j) \), where \( \hat{I}(x_j) = \min\{I(x_{j-1}), I(x_j)\} \).

**Proof:** From Lemma 29, we have \( S_1(I) = \sum_{i=1}^{g} [I(x_{inci}) - I(x_{inci-1})] \). For the left profile, \( S_1(P_j) = \sum_{inci < j} [I(x_{inci}) - I(x_{inci-1})] \). The optimal therapy time of the right profile \( S_j \) is equal to the sum of the increments in the intensities of successive sample points of the right profile. Adding these increments, we get, \( S_1(S_j) = S_j(x_j) - S_j(x_{j-1}) + \sum_{inci > j} [I(x_{inci}) - I(x_{inci-1})] = I(x_j) + \sum_{inci > j} [I(x_{inci}) - I(x_{inci-1})] \) (since \( S_j(x_{j-1}) = 0 \) and \( S_j(x_j) = I(x_j) \)). If \( I(x_j) > I(x_{j-1}) \), this can be written as \( S_1(S_j) = (I(x_j) - I(x_{j-1})) + \sum_{inci > j} [I(x_{inci}) - I(x_{inci-1})] + I(x_{j-1}) \). If \( I(x_j) \leq I(x_{j-1}) \), \( S_1(S_j) = \sum_{inci > j} [I(x_{inci}) - I(x_{inci-1})] + I(x_{j-1}) \) (since \( S_j(x_{j-1}) = 0 \)). Therefore \( S_1(S_j) = \sum_{inci > j} [I(x_{inci}) - I(x_{inci-1})] + I(x_{j-1}) = \sum_{inci > j} [I(x_{inci}) - I(x_{inci-1})] + I(x_{j-1}) \). By addition, \( S_1(P_j) + S_1(S_j) = \sum_{i=1}^{g} [I(x_{inci}) - I(x_{inci-1})] + \min\{I(x_{j-1}), I(x_j)\} = S_1(I) + \hat{I}(x_j) \).

We illustrate Lemma 30 using the example of Figure 5–1. The optimal therapy time for the profile \( I \) is the sum of increments in intensity values of successive sample points. However, if \( I \) is split at \( x_3 \) into \( P_3 \) and \( S_3 \), an additional therapy time of \( \hat{I}(x_3) = \min\{I(x_2), I(x_3)\} = I(x_3) \) is required for treatment. Similarly, if \( I \) is split at \( x_4 \) into \( P_4 \) and \( S_4 \), an additional therapy time of \( \hat{I}(x_4) = \min\{I(x_3), I(x_4)\} = I(x_3) \) is required.

Algorithm S2

(1) Compute \( \hat{I}(x_i) = \min\{I(x_{i-1}), I(x_i)\} \), for \( g - w < i \leq w + 1 \).

(2) Split the field at a point \( x_j \) where \( \hat{I}(x_j) \) is minimized for \( g - w < j \leq w + 1 \).

It is evident from Lemma 30 that if the width of the profile is less than the maximum allowable field width \( (g \leq w) \), the profile is best delivered using a single field. If \( g > 2w \) two fields are insufficient. So it is useful to apply Algorithm S2 only for \( w < g \leq 2w \). Once
Figure 5–1: Splitting a profile (a) into two. (b) and (c) show the left and right profiles resulting from a split at $x_3$; (d) and (e) show the left and right profiles resulting from a split at $x_4$. 
the profile $I$ is split into two as determined by Algorithm $S_2$, the left and right profiles are
delivered using separate fields. The total therapy time is $S_2(I) = S_1(P_j) + S_1(S_j)$, where
$j$ is the split point.

**Splitting a profile into three.** Suppose that a profile $I$ is split into three profiles. Let
$j$ and $k$, $j < k$, be the indices at which the profile is split. As a result we get three profiles
$P_j, M_{(j,k)}$ and $S_k$, where $P_j(x_i) = I(x_i), 1 \leq i < j$, $M_{(j,k)}(x_i) = I(x_i), j \leq i < k$, and
$S_k(x_i) = I(x_i), k \leq i \leq g$. $P_j, M_{(j,k)}$ and $S_j$ are zero at all other points. $P_j$ is a left profile,
$M_{(j,k)}$ is a middle profile of $I$ and $S_k$ is a right profile.

**Lemma 31** Let $S_1(P_j), S_1(M_{(j,k)})$ and $S_1(S_k)$ be the optimal therapy times, respectively,
for $P_j, M_{(j,k)}$ and $S_k$. Then $S_1(P_j) + S_1(M_{(j,k)}) + S_1(S_k) = S_1(I) + \min\{I(x_{j-1}), I(x_j)\} + \min\{I(x_{k-1}), I(x_k)\} = S_1(I) + \hat{I}(x_j) + \hat{I}(x_k)$.

**Proof:** Similar to that of Lemma 30

Lemma 31 motivates the following algorithm for $S_3$.

**Algorithm $S_3$**

1. Compute $\hat{I}(x_i) = \min\{I(x_{i-1}), I(x_i)\}$, for $1 < i \leq w + 1, g - w < i \leq g$.
2. Split the profile at two points $x_j, x_k$ such that $1 \leq j \leq w + 1, g - w < k \leq g$,
   $0 < k - j \leq w$, and $\hat{I}(x_j) + \hat{I}(x_k)$ is minimized.

Note that for Algorithm $S_3$ to split $I$ into three profiles that are each deliverable in one
field, it must be the case that $g \leq 3w$. Once the profile $I$ is split into three as determined
by Algorithm $S_3$, the resulting profiles are delivered using separate fields. The minimum
total therapy time is $S_3(I) = S_1(P_j) + S_1(M_{(j,k)}) + S_1(S_k)$. Algorithm $S_3$ examines at
most $g^2$ candidates for $(j, k)$. So the complexity of the algorithm is $O(g^2)$.

**Bounds on optimal therapy time ratios.** We prove the following bounds on ratios of
optimal therapy times.

**Lemma 32**

(a) $1 \leq S_2(I)/S_1(I) \leq 2$

(b) $1 \leq S_3(I)/S_1(I) \leq 3$

(c) $0.5 < S_3(I)/S_2(I) < 2$

**Proof:**

(a) $S_2(I) = \sum_{i=1}^{q} [I(x_{inci}) - I(x_{inci-1})] + \min\{I(x_{j-1}), I(x_j)\} = S_1(I) + \min\{I(x_{j-1}), I(x_j)\}$, where $j$ is the best point to split the field as determined by Algorithm $S_2$. This implies $S_2(I)/S_1(I) \geq 1$ and so splitting a field into two never
improves optimal therapy time. For an upper bound on the ratio, note that $S_1(I) \geq \min\{I(x_{j-1}), I(x_j)\}$ since at least $\min\{I(x_{j-1}), I(x_j)\}$ MUs are required to deliver $I$. So $S_2(I) \leq 2 \times S_1(I)$. The example of Figure 5–2 shows that the upper bound is tight. The profile $I$ has $2w$ sample points, i.e., it has a width $2w\Delta x$. So it has to be split exactly at $x_{w+1}$. The resulting left and right profiles each have an optimal therapy time equal to that of $I$.

\[ S_3(I) = S_1(I) + \min\{I(x_{j-1}), I(x_j)\} + \min\{I(x_{k-1}), I(x_k)\}, \] where $j$ and $k$ are as in Algorithm $S_3$. Clearly, $S_3(I)/S_1(I) \geq 1$. Also, $S_1(I) \geq \min\{I(x_{j-1}), I(x_j)\}$ and $S_1(I) \geq \min\{I(x_{k-1}), I(x_k)\}$. Therefore, $S_3(I) \leq 3 \times S_1(I)$. Once again the upper bound is tight as shown in the Figure 5–3. The profile shown has width $3w\Delta x$ and needs to be split at $x_{w+1}$ and at $x_{2w+1}$. Each of the resulting profiles has optimal therapy time equal to $S_1(I)$.

From (a) and (b), $S_3(I) \geq S_1(I)$ and $S_2(I) \leq 2 \times S_1(I)$. So $S_3(I)/S_2(I) \geq 0.5$. $S_3(I)/S_2(I) = 0.5$ only if $S_3(I) = S_1(I)$ and $S_2(I) = 2 \times S_1(I)$. Suppose
that $S_3(I) = S_1(I)$. Then there exist indices $j,k$ such that $\min\{I(x_{j-1}), I(x_j)\} + \min\{I(x_{k-1}), I(x_k)\} = 0$, i.e., $\min\{I(x_{j-1}), I(x_j)\} = 0$ and $\min\{I(x_{k-1}), I(x_k)\} = 0$. This and the fact that $I(x_1) \neq 0, I(x_g) \neq 0$ implies that the profile has at least two disjoint components separated by a sample point at which the desired intensity is zero. Sample points in the two disjoint components cannot be exposed at the same time and so there does not exist a point $x_i$ such that $I(x_i) = S_1(I)$. So $S_2(I) = S_1(I) + \min_{g-w<i\leq w+1} \min\{I(x_{i-1}), I(x_i)\} < 2 \cdot S_1(I)$. It follows that $S_3(I)/S_2(I) > 0.5$. Figure 5–4 shows an example where the ratio can be made arbitrarily close to 0.5. In this example, $S_1(I) = I_2$. The profile has a width of $2w\Delta x$ and therefore needs to be split at $x_{w+1}$. The resulting profiles each have an optimal therapy time of $S_1(I)$ so that $S_2(I) = 2 \cdot S_1(I)$. $S_3(I) = S_1(I) + 2I_1$ and so $S_3(I) \rightarrow S_1(I)$ as $I_1 \rightarrow 0$.

![Figure 5–4: Tight lower bound for Lemma 32c](image)

To obtain an upper bound note that the best split point for $S_2$ (say $x_j$) is always a permissible split point for $S_3$. By selecting this as one of the two split points for $S_3$, we can construct a split into three profiles such that the total therapy time of profiles resulting from this split is $S_2(I) + \min\{I(x_{k-1}), I(x_k)\}$, where $k$ is the second split point defining that split. Since $\min\{I(x_{k-1}), I(x_k)\} \leq S_1(I) \leq S_2(I)$, the total therapy time of the split $\leq 2 \cdot S_2(I)$. So $S_3(I)/S_2(I) \leq 2$. The ratio can be arbitrarily close to 2 as demonstrated in Figure 5–5. One can verify that for the profile $I$ in this example, $S_3(I)/S_2(I) \rightarrow 2$ as $I_1 \rightarrow 0$. 

\[ \]
Lemma 32 tells us that the optimal therapy times can at most increase by factors of 2 and 3, respectively, as a result of a splitting a single leaf pair profile into 2 and 3. Also, the optimal therapy time for a split into 2 can be at most twice that for a split into 3 and vice versa.

5.2.2 Optimal Field Splitting for Multiple Leaf Pairs

The input intensity matrix (say $I$) for the leaf sequencing problem is obtained using the inverse planning technique. The matrix $I$ consists of $n$ rows and $m$ columns. Each row of the matrix specifies the number of monitor units (MUs) that need to be delivered using one leaf pair. Denote the rows of $I$ by $I_1, I_2, \ldots, I_n$. For the case where $I$ is deliverable using one field, the leaf sequencing problem has been well studied in the past. The algorithm that generates optimal therapy time schedules for multiple leaf pairs (Algorithm MULTIPAIR) applies algorithm SINGLEPAIR independently to each row $I_i$ of $I$. Without loss of generality assume that the least column index containing a non zero element in $I$ is 1 and the largest column index containing a non zero element in $I$ is $g$. If $g > w$, the profile will need to be split. We define problems $M1$, $M2$ and $M3$ for multiple leaf pairs as being analogous to $S1$, $S2$ and $S3$ for single leaf pair. The optimal therapy times $M1(I)$, $M2(I)$ and $M3(I)$ are also defined similarly.

**Splitting a profile into two.** Suppose that a profile $I$ is split into two profiles. Let $x_j$ be the column at which the profile is split. This is equivalent to splitting each row profile
\( I_i, 1 \leq i \leq n, \) at \( j \) as defined for single leaf pair split. As a result we get two profiles, \( P_j \) (left) and \( S_j \) (right). \( P_j \) has rows \( P_j^1, P_j^2, \ldots, P_j^n \) and \( S_j \) has rows \( S_j^1, S_j^2, \ldots, S_j^n \).

**Lemma 33** Suppose \( I \) is split into two profiles at \( x_j \). The optimal therapy time for delivering \( P_j \) and \( S_j \) using separate fields is \( \max_i \{ S_1(P_j^i) \} + \max_i \{ S_1(S_j^i) \} \).

**Proof:** The optimal therapy time schedule for \( P_j \) and \( S_j \) are obtained using Algorithm MULTIPAIR. The therapy times are \( \max_i \{ S_1(P_j^i) \} \) and \( \max_i \{ S_1(S_j^i) \} \) respectively. So the total therapy time is \( \max_i \{ S_1(P_j^i) \} + \max_i \{ S_1(S_j^i) \} \).

From Lemma 33 it follows that the \( M_2 \) problem can be solved by finding the index \( j \), \( 1 < j \leq g \) such that \( \max_i \{ S_1(P_j^i) \} + \max_i \{ S_1(S_j^i) \} \) is minimized (Algorithm \( M_2 \)).

**Algorithm \( M_2 \)**

1. Compute \( \max_i \{ S_1(P_j^i) \} + \max_i \{ S_1(S_j^i) \} \) for \( g - w < j \leq w + 1 \).
2. Split the field at a point \( x_j \) where \( \max_i \{ S_1(P_j^i) \} + \max_i \{ S_1(S_j^i) \} \) is minimized for \( g - w < j \leq w + 1 \).

From Lemma 29, \( S_1(P_j^i) = \sum_{\text{inci} \leq j} [I(x_{\text{inci}}) - I(x_{\text{inci} - 1})] \). For each \( i \), \( S_1(P_1^i) \), \( S_1(P_2^i) \), \( \ldots \), \( S_1(P_g^i) \) can all be computed in a total of \( O(g) \) time progressively from left to right. So the computation of \( S_1 \)s (optimal therapy times) of all left profiles of all \( n \) rows of \( I \) can be done in \( O(ng) \) time. The same is true of right profiles. Once these values are computed, step (1) of Algorithm \( M_2 \) is applied. \( \max_i \{ S_1(P_j^i) \} + \max_i \{ S_1(S_j^i) \} \) can be found in \( O(n) \) time for each \( j \) and hence in \( O(ng) \) time for all \( j \) in the permissible range. So the time complexity of Algorithm \( M_2 \) is \( O(ng) \).

**Splitting a profile into three.** Suppose that a profile \( I \) is split into three profiles. Let \( j, k, j < k \), be the indices at which the profile is split. Once again, this is equivalent to splitting each row profile \( I_i, 1 \leq i \leq n \) at \( j \) and \( k \) as defined for single leaf pair split. As a result we get three profiles \( P_j, M_{(j,k)} \) and \( S_k \). \( P_j \) has rows \( P_j^1, P_j^2, \ldots, P_j^n \), \( M_{(j,k)} \) has rows \( M_{(j,k)}^1, M_{(j,k)}^2, \ldots, M_{(j,k)}^n \) and \( S_k \) has rows \( S_k^1, S_k^2, \ldots, S_k^n \).

**Lemma 34** Suppose \( I \) is split into three profiles by splitting at \( x_j \) and \( x_k \), \( j < k \). The optimal therapy time for delivering \( P_j, M_{(j,k)} \) and \( S_k \) using separate fields is \( \max_i \{ S_1(P_j^i) \} + \max_i \{ S_1(M_{(j,k)}^i) \} + \max_i \{ S_1(S_k^i) \} \).

**Proof:** Similar to that of Lemma 33.
Algorithm M3 solves the M3 problem.

Algorithm M3

1. Compute \( \max_i \{ S_i(P^i_j) \} + \max_i \{ S_i(M^i_{j,k}) \} + \max_i \{ S_i(S^i_k) \} \) for \( 1 < j \leq w + 1, \)
\( g - w < k \leq g, \) \( 0 < k - j \leq w. \)

2. Split the field at two points \( x_j, x_k, \) such that \( 1 < j \leq w + 1, \) \( g - w < k \leq g, \)
\( 0 < k - j \leq w, \) and \( \max_i \{ S_i(P^i_j) \} + \max_i \{ S_i(M^i_{j,k}) \} + \max_i \{ S_i(S^i_k) \} \) is minimized.

The complexity analysis is similar to that of Algorithm M2. In this case though, \( O(g^2) \) pairs of split points have to be examined. It is easy to see that the time complexity of Algorithm M3 is \( O(n g^2). \)

**Bounds on optimal therapy time ratios.** We prove the following bounds on ratios of optimal therapy times.

**Lemma 35**

(a) \( 1 \leq M2(I)/M1(I) \leq 2 \)

(b) \( 1 \leq M3(I)/M1(I) < 3 \)

(c) \( 0.5 < M3(I)/M2(I) < 2 \)

**Proof:**

(a) \( M2(I) = \max_i \{ S_i(P^i_j) \} + \max_i \{ S_i(S^i_j) \}, \) where \( j \) is as determined by Algorithm M2. \( \max_i \{ S_i(P^i_j) \} + \max_i \{ S_i(S^i_j) \} \geq \max_i \{ S_i(P^i_j) + S_i(S^i_j) \} \geq \max_i \{ S_i(I_i) \} = M1(I). \) This implies \( M2(I)/M1(I) \geq 1 \) and so splitting a field into two never improves optimal therapy time. For an upper bound on the ratio, note that \( \max_i \{ S_i(P^i_j) \} \leq \max_i \{ S_i(I_i) \} \) and \( \max_i \{ S_i(S^i_j) \} \leq \max_i \{ S_i(I_i) \}. \) Hence \( M2(I) = \max_i \{ S_i(P^i_j) \} + \max_i \{ S_i(S^i_j) \} \leq 2 * M1(I). \)

(b) \( M3(I) = \max_i \{ S_i(P^i_j) \} + \max_i \{ S_i(M^i_{j,k}) \} + \max_i \{ S_i(S^i_k) \}, \) where \( j, k \) are as in Algorithm M3. The proof that \( M3(I)/M1(I) \geq 1 \) is similar to that of (a). As in (a), \( M1(I) \geq \) each of the three terms in \( M3(I). \) Therefore, \( M3(I) \leq 3 * M1(I). \)

(c) From (a) and (b), \( M3(I) \geq M1(I) \) and \( M2(I) \leq 2 * M1(I). \) So \( M3(I)/M2(I) \geq 0.5. \) To obtain an upper bound note that the best split point for \( M2 \) (say \( x_j \)) is always a permissible split point for \( M3. \) By selecting this as one of the two split points for \( M3, \) we can construct a split into three profiles such that the total therapy time of profiles resulting from this split is \( \max_i \{ S_i(P^i_j) \} + \max_i \{ S_i(M^i_{j,k}) \} + \max_i \{ S_i(S^i_k) \}, \) where \( k \) is the second split point defining that split. Since \( \max_i \{ S_i(M^i_{j,k}) \} + \max_i \{ S_i(S^i_k) \} \leq 2 * \max_i \{ S_i(S^i_j) \}, \) it follows that the total therapy time of profiles resulting from this
split is max\(i\{S_i(P_j^i)\} + \max\{S_1(M_{i,j,k}^i)\} + \max\{S_1(S_k^i)\}\) \leq \max\{S_1(P_j^i)\} + 2 \ast \max\{S_1(S_k^i)\} \leq 2 \ast M_2(I). \) So \(M_3(I)/M_2(I) \leq 2.\)

Note that the examples used to show tightness of bounds in the proof of Lemma 32 can also be used to show tightness of bounds in this case. □

Lemma 35 tells us that the optimal therapy times can at most increase by factors of 2 and 3, respectively, as a result of splitting a field into 2 and 3. Also, the optimal therapy time for a split into 2 can be at most twice that for a split into 3 and vice versa. These bounds give us the potential benefits of designing MLCs with larger maximal aperture so that large fields do not need to be split.

**Tongue-and-groove effect and interdigitation.** Algorithms \(M_2\) and \(M_3\) may be extended to generate optimal therapy time fields with elimination of tongue-and-groove underdosage and (optionally) the interdigitation constraint on the leaf sequences. Kamath et al. (2004) present algorithms for delivering an intensity matrix \(I\) using a single field with optimal therapy time, while eliminating the tongue-and-groove underdosage (Algorithm TONGUEANDGROOVE) and also while simultaneously eliminating the tongue-and-groove underdosage and interdigitation constraint violations (Algorithm TONGUEANDGROOVERID). Denote these problems by \(M_1'\) and \(M_1''\) respectively (\(M_2', M_2'', M_3'\) and \(M_3''\) are defined similarly for splits into two and three fields). Let \(M_1'(I)\) and \(M_1''(I)\), respectively, denote the optimal therapy times required to deliver \(I\) using the leaf sequences generated by these algorithms. To solve problem \(M_2'\) we need to determine \(x_j\) where \(M_1'(P_j^i) + M_1'(S_j^i)\) is minimized for \(g - w < j \leq w + 1\). Note that this is similar to Algorithm \(M_2\). Using the fact that \(M_1'\) can be solved in \(O(nm)\) time for an intensity profile with \(n\) rows and \(m\) columns (Lemma 7, Kamath et al. (2004)), and by computing \(M_1'(P_j^i)\) and \(M_1'(S_j^i)\) progressively from left to right, it is possible to solve \(M_2'\) in \(O(ng)\) time. In case of \(M_3'\) we need to find \(x_j, x_k\), such that \(1 < j \leq w + 1, g - w < k \leq g, 0 < k - j \leq w,\) and \(M_1'(P_j^i) + M_1'(M_{j,k}^i) + M_1'(S_k^i)\) is minimized. \(M_3'\) can be solved in \(O(n^2g^2)\) time. The solutions for \(M_2''\) and \(M_3''\) are now obvious.

### 5.3 Field Splitting with Feathering

One of the problems associated with field splitting is the field matching problem that occurs in the field junction region due to uncertainties in setup and organ motion (Wu...
et al. 2000). To illustrate the problem we use an example. Consider the single leaf pair intensity profile of Figure 5–6a. Due to width limitations, the profile needs to be split. Suppose that it is split at $x_j$. Further suppose that the left field is delivered accurately and that the right field is misaligned so that its left end is positioned at $x'_j$ rather than $x_j$. Due to incorrect field matching the actual profile delivered may be, for example, either of the profiles shown in Figure 5–6b or Figure 5–6d, depending on the direction of error. In Figure 5–6b, the region between $x'_j$ and $x_j$ gets overdosed and is a hotspot. In Figure 5–6d, the region between $x_j$ and $x'_j$ gets underdosed and is a coldspot.

One way to partially eliminate the field matching problem is to use the ‘feathering’ technique (Wu et al. 2000). In this technique, the large field is not split at one sample point into two non-overlapping fields. Instead the profiles to be delivered by the two fields resulting from the split, overlap over a central feathering region. The beam splitting algorithm proposed by Wu et al. (2000) splits a large field with feathering, such that in the feathering region the sum of the split fields equals the desired intensity profile. Figure 5–7a shows a split of the profile of Figure 5–6 with feathering. Figures 5–7c and 5–7d show the effect of field matching problem on the split with feathering. The extent of field mismatches is the same as those in Figures 5–6b and 5–6d, respectively. Note that while the profile delivered in the case with feathering is not the exact profile either, the delivered profile is less sensitive to mismatch compared to the case when it is split without feathering as in Figure 5–6. In other words, the purpose of feathering is to lower the magnitude of maximum intensity error $e$ in the delivered profile from the desired profile over all sample points in the junction region.

In this section, we extend our field splitting algorithms to incorporate feathering. In order to do so, we define a feathering scheme similar to that of Wu et al. (2000). However, there are two differences between the splitting algorithm we propose and the algorithm of Wu et al. (2000). First, our feathering scheme is defined for profiles discretized in space and in MUs as is the profile generated by the optimizer. Second, the feathering scheme we propose defines the profile values in the feathering region, which is centered at some sample point called the split point for that split. Thus given a split point, our scheme will specify how to split the large field with a feathering region that is centered at that point. The split
Figure 5–6: Field matching problem: The profile in (a) is the desired profile. It is split into two fields at $x_j$. Due to incorrect field matching, the left end of right field is positioned at point $x'_j$ instead of $x_j$ and the fields may overlap as in (c) or may be separated as in (d). In (c), the dotted line shows the left profile and the dashed line shows the right profile. (b) shows these profiles as well as the delivered profile in this case in bold. In (d), the left and right fields are separated and their two profiles together constitute the delivered profile, which is shown in bold. The delivered profiles in these cases, vary significantly from the desired profile in the junction region. $e$ is the maximum intensity error in the junction region, i.e., the maximum deviation of delivered intensity from the desired intensity.
Figure 5–7: Example of field splitting with feathering: (a) shows a split of the profile of Figure 5–6 with feathering. The dotted line shows the right part of the left profile and the dashed line shows the left part of the right profile. The left and right profiles are shown separately in (b). (c) and (d) show the effect of field matching problem on the split with feathering. The extent of field mismatches in (c) and (d) are the same as those in Figure 5–6b and Figure 5–6d, respectively, i.e., the distances between $x_j$ and $x'_j$ are the same as in Figure 5–6. Note that the maximum intensity error $e$ reduces in both cases with feathering.
point to be used in the actual split will be determined by a splitting algorithm that takes
into account the feathering scheme. In contrast, Wu et al. (2000) always choose the center
of the intensity profile as the split point, as they do not optimize the split with respect to
any objective.

We study how to split a single leaf pair profile into two (three) fields using our feathering
scheme such that the sum of the optimal therapy times of the individual fields is minimized.
We will denote this minimization problem by $S_2F$ ($S_3F$). The extension of the methods
developed for the multiple leaf pairs problems ($M_2F$ and $M_3F$) is straightforward and is
therefore not discussed separately.

5.3.1 Splitting a Profile into Two

Let $I$ be a single leaf pair profile. Let $x_j$ be the split point and let $P_j$ and $S_j$ be
the profiles resulting from the split. $P_j$ is a left profile and $S_j$ is a right profile of $I$. The
feathering region spans $x_j$ and $d - 1$ sample points on either side of $x_j$, i.e., the feathering
region stretches from $x_j - d + 1$ to $x_j + d - 1$. $P_j$ and $S_j$ are defined as follows.

\[ P_j(x_i) = \begin{cases} I_j(x_i) & 1 \leq i \leq j - d \\ [I_j(x_i) \ast (j + d - i)/2d] & j - d < i < j + d \\ 0 & j + d \leq i \leq g \end{cases} \] (5.1)

\[ S_j(x_i) = \begin{cases} 0 & 1 \leq i \leq j - d \\ I_j(x_i) - P_j(x_i) & j - d < i < j + d \\ I_j(x_i) & j + d \leq i \leq g \end{cases} \] (5.2)

Note that the profiles overlap over the $2d-1$ points $j-d+1, j-d+2, \ldots, j+d-2, j+d-1$.

Therefore, for the profile $I$ of width $g$ to be deliverable using two fields, it must be the case
that $g \leq 2w - 2d + 1$. Since $P_j$ needs to be delivered using one field, the split point
$x_j$ and at least $d - 1$ points to the right of it should be contained in the first field, i.e.,
$j + d - 1 \leq w \Rightarrow j \leq w - d + 1$. Similarly, since $S_j$ has to be delivered using one field
$j - (d - 1) > g - w \Rightarrow j \geq g - w + d$. These range restrictions on $j$ lead to an algorithm for
the $S_2F$ problem. Algorithm $S_2F$, which solves problem $S_2F$, is described below. Note
that the $P_i$ and $S_i$ can all be computed in a single left to right sweep in $O(d)$ time at each $i$. So the time complexity of Algorithm $S2F$ is $O(d)g$.

Algorithm $S2F$

1. Find $P_i$ and $S_i$ using Equations 5.1 and 5.2, for $g - w + d \leq i \leq w - d + 1$.
2. Split the field at a point $x_j$ where $S1(P_j) + S1(S_j)$ is minimized for $g - w + d \leq j \leq w - d + 1$.

### 5.3.2 Splitting a Profile into Three

Suppose that a profile $I$ is split into three profiles with feathering. Let $j$ and $k$, $j < k$, be the two split points. As a result we get three profiles $P_j$, $M_{(j,k)}$ and $S_k$, where $P_j$ is a left profile, $M_{(j,k)}$ is a middle profile of $I$ and $S_k$ is a right profile. In this case, there are two feathering regions, each of which spans across $2d - 1$ sample points centered at the corresponding split point. One feathering region stretches from $x_{j-d+1}$ to $x_{j+d-1}$ and the other from $x_{k-d+1}$ to $x_{k+d-1}$. $P_j$, $M_{(j,k)}$ and $S_j$ are defined as follows.

\[
P_j(x_i) = \begin{cases} 
I_j(x_i) & 1 \leq i \leq j - d \\
\lceil I_j(x_i) \ast (j + d - i) / 2d \rceil & j - d < i < j + d \\
0 & j + d < i \leq g
\end{cases} \tag{5.3}
\]

\[
M_{(j,k)}(x_i) = \begin{cases} 
0 & 1 \leq i \leq j - d \\
I_j(x_i) - P_j(x_i) & j - d < i < j + d \\
I_j(x_i) & j + d \leq i \leq k - d \\
\lceil I_k(x_i) \ast (k + d - i) / 2d \rceil & k - d < i < k + d \\
0 & k + d \leq i \leq g
\end{cases} \tag{5.4}
\]

\[
S_j(x_i) = \begin{cases} 
0 & 1 \leq i \leq k - d \\
I_j(x_i) - M_{(j,k)}(x_i) & k - d < i < k + d \\
I_j(x_i) & k + d \leq i \leq g
\end{cases} \tag{5.5}
\]

The profiles $P_j$ and $M_{(j,k)}$ overlap over $2d-1$ points, as do $M_{(j,k)}$ and $S_k$. For the profile $I$ to be deliverable using three fields, it must be the case that $g \leq 3w - 2(2d - 1) = 3w - 4d + 2$. Also, it is undesirable for the two feathering regions to overlap. So $g \geq 4d - 2$. For the feathering regions to be well defined and for the split to be useful it can be shown...
that \( g - 2w + 3d - 1 \leq j \leq w - d + 1 \) and that \( g - w + d \leq k \leq 2w - 3d + 2 \). Also, \( k - j + 1 + 2(d - 1) \leq w \Rightarrow k - j \leq w - 2d + 1 \). Using these ranges for \( j \) and \( k \), we arrive at Algorithm \( S3F \), which can be implemented to solve problem \( S3F \) in \( O(dg^2) \) time.

Algorithm \( S3F \)

1. Find \( P_j, M_{(j,k)} \) and \( S_k \) using Equations 5.3, 5.4 and 5.5, for \( g - 2w + 3d - 1 \leq j \leq w - d + 1, g - w + d \leq k \leq 2w - 3d + 2 \) and \( k - j \leq w - 2d + 1 \).
2. Split the field at two points \( x_j, x_k \), where \( S1(P_j) + S1(M_{(j,k)}) + S1(S_j) \) is minimized, subject to \( g - 2w + 3d - 1 \leq j \leq w - d + 1, g - w + d \leq k \leq 2w - 3d + 2 \) and \( k - j \leq w - 2d + 1 \).

5.3.3 Tongue-and-groove Effect and Interdigitation

The algorithms for \( M2F \) and \( M3F \) may be further extended to generate optimal therapy time fields with elimination of tongue-and-groove underdosage and (optionally) the interdigitation constraint on the leaf sequences as is done for field splits without feathering in Section 5.2.2. The definitions of problems \( M2F' (M3F') \) and \( M2F'' (M3F'') \), respectively, for splits into two (three) fields are similar to those made in Section 5.2.2 for splits without feathering.

5.4 Results

The performance of the Algorithms \( M2, M3, M2F \) and \( M3F \) was tested using 27 clinical fluence matrices, each of which exceeded the maximum allowable field width \( w = 14 \), with \( d = 2 \) for feathering. The fluence matrices were generated with a commercial inverse treatment planning system (CORVUS v5.0, NOMOS Corp., Sewickley, PA) for five clinical cases. Algorithm \( M2F \) was used when the profile width was \( \leq 2w - 2d + 1 = 25 \) and algorithm \( M2 \) was used whenever the profile width was \( \leq 2w = 28 \). Algorithms \( M3 \) and \( M3F \) were used in all cases. The optimal MUs for the split fields were calculated assuming that the split fields in each case are delivered by sequencing leaves using Algorithm MULTIPAIR. Table 5–1 displays the resulting total MUs for the field splits obtained using the four algorithms. Also shown are the total MUs obtained using the field split lines as given by the commercial treatment planning system (\( C(I) \)). The MUs are normalized to give a maximum pixel value of 100 of a fluence map. The percent decrease in MUs of \( \min\{M2(I), M3(I)\} \) as a result of optimal field splitting over \( C(I) \) is also shown in the last.
column. MU reductions of up to 26% are seen. In about 30% of the cases the reduction was over 20%. The average decrease in MUs is found to be about 11% for the 27 fluence matrices. Note that the application of optimal splitting algorithms with feathering can reduce MU as compared to the optimal algorithms without feathering as a result of the reduction in intensity values in the feathering region in each field resulting from the split. We observe that $M_2 F(I) < M_2(I)$ in over 34% of the cases and $M_3 F(I) < M_3(I)$ in over 40% of the cases. Examination of the optimal split lines from our algorithms shows that the split lines generally occurred in low fluence columns. Figure 5–8 compares the split line from Algorithm $M_2$ and that from the commercial planning system for one of the fluence matrices. The split line from the commercial planning system occurred at the center of the field, whereas a slight shift in the split line reduces the total MU by 10% in this case (Table 5–1). For extreme (synthetic) cases, more MU reduction can be achieved. For example, consider an intensity profile each row of which consists of the following non-zero pattern: eight 5s followed by thirteen 100s followed by eight 5s. The commercial treatment planning system split the field into three such that each of the resulting fields had at least one intensity value of 100. As a result, the optimal MUs for delivering this split is 300. However, Algorithm $M_3$ split this field into three fields such that the first and third fields contained only 5s and only the middle field contained 100s (note that $w=14$). The optimal MUs for this split is 110. The MU reduction is 63% for this case.

Figure 5–8: Comparison of the field split line obtained with Algorithm $M_2$ (solid line) with the split line from the commercial planning system (dashed line) for the fluence matrix 8 in Table 5–1. The isocenter is marked with the solid circle.
Table 5–1: Total MUs for five clinical cases

<table>
<thead>
<tr>
<th>Matrix $(I)$</th>
<th>Width</th>
<th>$C(I)$</th>
<th>$M2(I)$</th>
<th>$M3(I)$</th>
<th>$M2F(I)$</th>
<th>$M3F(I)$</th>
<th>% MU decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>280</td>
<td>280</td>
<td>330</td>
<td>290</td>
<td>335</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>310</td>
<td>310</td>
<td>350</td>
<td>318</td>
<td>350</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>300</td>
<td>260</td>
<td>340</td>
<td>260</td>
<td>310</td>
<td>13.3</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>400</td>
<td>300</td>
<td>370</td>
<td>290</td>
<td>353</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>350</td>
<td>350</td>
<td>380</td>
<td>360</td>
<td>394</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>340</td>
<td>310</td>
<td>310</td>
<td>325</td>
<td>382</td>
<td>8.8</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>390</td>
<td>310</td>
<td>360</td>
<td>310</td>
<td>338</td>
<td>20.5</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>350</td>
<td>320</td>
<td>340</td>
<td>320</td>
<td>357</td>
<td>8.6</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>400</td>
<td>300</td>
<td>370</td>
<td>310</td>
<td>365</td>
<td>25</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>440</td>
<td>350</td>
<td>390</td>
<td>322</td>
<td>398</td>
<td>20.5</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>320</td>
<td>320</td>
<td>360</td>
<td>310</td>
<td>362</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>400</td>
<td>300</td>
<td>340</td>
<td>280</td>
<td>360</td>
<td>25</td>
</tr>
<tr>
<td>13</td>
<td>17</td>
<td>380</td>
<td>280</td>
<td>320</td>
<td>285</td>
<td>323</td>
<td>26.3</td>
</tr>
<tr>
<td>14</td>
<td>18</td>
<td>280</td>
<td>240</td>
<td>260</td>
<td>240</td>
<td>280</td>
<td>14.3</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>320</td>
<td>320</td>
<td>380</td>
<td>320</td>
<td>375</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>22</td>
<td>400</td>
<td>360</td>
<td>400</td>
<td>360</td>
<td>400</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>22</td>
<td>320</td>
<td>320</td>
<td>360</td>
<td>300</td>
<td>360</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>24</td>
<td>540</td>
<td>480</td>
<td>520</td>
<td>480</td>
<td>500</td>
<td>11.1</td>
</tr>
<tr>
<td>19</td>
<td>24</td>
<td>540</td>
<td>500</td>
<td>500</td>
<td>490</td>
<td>500</td>
<td>7.4</td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>460</td>
<td>420</td>
<td>460</td>
<td>420</td>
<td>425</td>
<td>8.7</td>
</tr>
<tr>
<td>21</td>
<td>24</td>
<td>520</td>
<td>520</td>
<td>540</td>
<td>525</td>
<td>545</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>24</td>
<td>560</td>
<td>520</td>
<td>520</td>
<td>505</td>
<td>505</td>
<td>7.1</td>
</tr>
<tr>
<td>23</td>
<td>25</td>
<td>360</td>
<td>360</td>
<td>380</td>
<td>340</td>
<td>380</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>26</td>
<td>560</td>
<td>440</td>
<td>460</td>
<td>-</td>
<td>430</td>
<td>21.4</td>
</tr>
<tr>
<td>25</td>
<td>29</td>
<td>520</td>
<td>-</td>
<td>480</td>
<td>-</td>
<td>445</td>
<td>7.7</td>
</tr>
<tr>
<td>26</td>
<td>29</td>
<td>580</td>
<td>-</td>
<td>440</td>
<td>-</td>
<td>440</td>
<td>24.1</td>
</tr>
<tr>
<td>27</td>
<td>32</td>
<td>560</td>
<td>-</td>
<td>480</td>
<td>-</td>
<td>470</td>
<td>14.3</td>
</tr>
</tbody>
</table>
5.5 Conclusion

We have developed algorithms to split large intensity-modulated fields into two or three sub-fields. Such a work-around needs to be implemented for MLCs that have a maximum leaf spread limitation, which imposes a field width limitation. We have presented algorithms that split large fields into non-overlapping sub-fields along one or two columns. Also presented are algorithms that split fields with feathering. Feathering of split fields helps reduce the effect of the field matching problem that occurs in the field junction region due to uncertainties in setup and organ motion. We have shown that our algorithms result in field splits for which the MU efficiency is optimal. Application of our optimal field splitting algorithms without feathering to clinical data reduced total MUs by up to 26% and on synthetic data up to 63% compared to a commercial planning system that also splits fields without feathering. We have also shown that our algorithms can easily be extended to split fields resulting in maximal MU efficiency when the MLC model is subject to the interdigitation constraint and/or the tongue-and-groove effect is to be eliminated.
CHAPTER 6
CONCLUSION

We have presented a systematic study of leaf sequencing algorithms for multileaf col-
limation. Algorithms are presented for sequencing leaves without any constraints, with
the intra-pair maximum separation constraint and with the inter-pair minimum separation
constraint for SMLC and without any constraints, with the intra-pair maximum separation
constraint and with the interdigitation constraint for DMLC. Also presented are algorithms
that eliminate the tongue-and-groove underdosage (and optionally the interdigitation con-
straint) for SMLC and a comparison of these algorithms with a recently published algorithm
that also eliminates the tongue-and-groove effect. Finally, algorithms are developed, that
split a large intensity modulated field into two or three subfields. We have shown that
all these algorithms obtain feasible solutions whenever they exist. Further, the solutions
generated are always optimal in therapy time for unidirectional schedules. The algorithms
developed are applicable to some of the popular commercially available delivery systems.
REFERENCES


Chui C-S, LoSasso T and Spirou S 1994 Dose calculation for photon beams with intensity modulation generated by dynamic jaw or multileaf collimations *Medical Physics* **21** 1237-44

Convery D J and Rosenbloom M E 1992 The generation of intensity-modulated fields for conformal radiotherapy by dynamic collimation *PMB* **37** 1359-74


Engel E 2003 A new algorithm for optimal multileaf collimator field segmentation *Preprint, Fachbereich Mathematik, Uni Rostock*

Followill D, Geis P and Boyer A 1997 Estimates of whole-body dose equivalent produced by beam intensity modulated conformal therapy *Int J Rad Oncol Biol Phys* 38 667-672

Kalinowski T 2003 An algorithm for optimal multileaf collimator field segmentation with interleaf collision constraint *Preprint, Fachbereich Mathematik, Uni Rostock*


Intensity Modulated Radiation Therapy Collaborative Working Group 2001 Intensity-modulated radiotherapy: Current status and issues of interest *Int J Radiation Oncology Biol Phys* 51 880-914

Jordan T J and Williams P C 1994 The design and performance characteristics of a multileaf collimator *PMB* 39 231-51

Kallman P, Lind B, Eklof A and Brahme A 1988 Shaping of arbitrary dose distributions by dynamic multileaf collimation, *PMB* 33 1291-300

Kamath S, Sahni S, Li J, Palta J and Ranka S 2003 Leaf sequencing algorithms for segmented multileaf collimation *PMB* 48 307-324


Langer M, Thai V and Papiez L 2001 Improved leaf sequencing reduces segments or monitor units needed to deliver IMRT using multileaf collimators *Medical Physics* 28 2450-2458

Li Y, Yao J, Yao D 2003 Genetic algorithm based deliverable segments optimization for static intensity-modulated radiotherapy *PMB* 48 3353-74

Luan S, Chen D, Zhang L, Wu X and Yu C X 2003 An optimal algorithm for configuring delivery options of a one dimensional intensity-modulated beam *PMB* 48 2321-38
Ma L, Boyer A, Xing L and Ma C-M 1998 An optimized leaf-setting algorithm for beam intensity modulation using dynamic multileaf collimators *PMB* 43 1629-43

Ma L, Boyer A L, Ma C M, and Xing L 1999 Synchronizing dynamic multileaf collimators for producing two-dimensional intensity-modulated fields with minimum beam delivery time, *Int J Radiation Oncology Biol Phys* 44 1147-54

Mohan R 1995 Field shaping for three-dimensional conformal radiation therapy and multileaf collimation *Seminars in Radiat Oncol* 5 86-99


Que W 1999 Comparison of algorithms for multileaf collimator field segmentation *Medical Physics* 26 2390-96

Que W, Kung J and Dai J 2004 “Tongue-and-groove” effect in intensity modulated radiotherapy with static multileaf collimator fields *PMB* 49 399-405

Siochi S 1999 Minimizing static intensity modulation delivery time using an intensity solid paradigm *Int J Radiation Oncology Biol Phys* 43 671-80

Spirou S V and Chui C S 1994 Generation of arbitrary intensity profiles by dynamic jaws or multileaf collimators *Medical Physics* 21 1031-41


Sykes R J and Williams P C 1998 An experimental investigation of the tongue and groove effect for the Philips multileaf collimator *PMB* 43 3157-65

van Santvoort J P C and Heijmen B J M 1996 Dynamic multileaf collimation without “tongue-and-groove” underdosage effects *PMB* 41 2091-105


Xia P and Verhey L J 1998 Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple static segments *Medical Physics* 25 1424-34
Xia P and Verhey L J 2001 Delivery systems of intensity-modulated radiotherapy using conventional multileaf collimators *Medical Dosimetry* **26** 169-77
BIOGRAPHICAL SKETCH

Srijit Kamath completed his high school education from Kendriya Vidyalaya, IIT Madras, India, in 1996. In 2000 he received a B. Tech. degree in information technology from the University of Madras, India. He then moved to the University of Florida (UF) for graduate studies. Since August 2000, he has been a Research Assistant in the Computer and Information Science and Engineering Department. His work at UF has led to five journal papers and two patent applications. He earned a M.S. in 2002 and will receive a Ph.D. in 2005, both from UF.