

调强放疗中强度矩阵分解算法的研究*

齐东洲¹, 吴敏^{1†}, 刘晓慧², 陈宏亮³, 余彬和⁴

(1. 华东师范大学上海市高可信计算重点实验室, 上海 200062; 2. 同济大学医学院, 上海 200092; 3. Elekta 上海研发部, 上海 200122; 4. 上海交通大学上海市信息系统实验室, 上海 200240)

摘要: 强度矩阵分解作为调强放疗(IMRT)中的一个核心技术,是一个 NP-hard 问题。调强一般分为静态模式和动态模式两种。针对静态调强放疗,对总治疗时间主要依赖的四个参数:总机器跳数(TNMU)、子野个数(NS)、多叶准直器叶片移动速率和验证记录(V&R)进行了研究,基于 Xia & Verhey 算法提出了一种高效的算法,能达到总治疗时间尽可能少的效果。经过随机与临床强度矩阵数据的大量测试,提出的算法与同类算法如 Galvin 算法、Bortfeld 算法、Xia & Verhey 算法、Siochi 算法以及 Luan 算法相比,都能取得最小的总治疗时间,从而更具有实际可行性。

关键词: 调强放疗; 强度矩阵; 子野; 总机器跳数; 多叶准直器

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2017)03-0683-06

doi:10.3969/j.issn.1001-3695.2017.03.010

Highly-efficient greedy algorithm for field splitting in intensity-modulated radiation therapy

Qi Dongzhou¹, Wu Min^{1†}, Liu XiaoHui², Chen Hongliang³, Yu Binhe⁴

(1. Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China; 2. Medical College, Tongji University, Shanghai 200092, China; 3. Dept. of R&D, Elekta, Shanghai 200122, China; 4. Shanghai Information System Laboratory, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: Field splitting is one of the core techniques in the intensity-modulated radiation therapy (IMRT), which is an NP-hard problem. Intensity modulation includes two cases: static mode (i. e., step and shoot) and dynamic mode. Regarding the static mode, this paper considered how to reduce the total treatment time of IMRT, which depended on four parameters: the total number of monitor units (TNMU), the number of segments (NS), the MLC's speed, and the verification and recording time (V&R). Based on Xia & Verhey's algorithm, this paper presented an optimal algorithm by taking all these four factors into account. Experiments with random data and clinical data show that, compared with the classic algorithms proposed by Galvin, Bortfeld, Xia & Verhey, Siochi and Luan, the new algorithm always produces optimal results on total treatment time and speeds up significantly.

Key words: intensity modulation; intensity map; field splitting; segmentation; multileaf collimator (MLC)

0 引言

放射治疗经历了立体定位放疗、三维适形放疗和调强放疗(IMRT)等阶段。其中 IMRT 以其精准的放疗效果获得医院的青睐,它可以使肿瘤区域接受较多的射线照射,同时保护周围正常的器官如颈椎、耳膜,避免受到二次伤害。IMRT 的逆向放射治疗系统包含三个模块:影像数据传入模块、计划设计与优化模块、剂量仿真计算模块。计划设计与优化模块的一个核心技术是强度矩阵(intensity matrix or intensity map, IM)分解,也叫做射野分割。有效的强度矩阵分解算法不仅能提高放疗的精准度,而且可以提高放疗效率,这意味着治疗时间缩短,从而降低病人长时间照射带来的风险。因此,强度矩阵分解问题的研究具有重要的应用价值^[1]。

作为计划设计与优化模块产生的一个中间结果,强度矩阵是根据医生处方(如肿瘤器官需要照射的剂量)及其他约束条件,通过数学优化方法计算出的一个非负矩阵,矩阵中元素的值表示各部位的放疗剂量。由于多叶准直器(multi leaf collimator, MLC)的输入需为 0-1 矩阵且每一行中的元素 1 必须连续,所以强度矩阵不能直接作为多叶准直器的输入,需要被分解成符合 0-1 矩阵的和,这个分解过程称做强度矩阵分解^[2]。

单行强度矩阵分解是一个 NP-完全问题,也是 APX 完全问题^[3],因此强度矩阵分解^[4]也属于这类问题。国内外学者提出了不少有效的强度矩阵分解算法。Galvin、Xia、Siochi 和 Luan 等人^[5-8]提出了不同的启发式算法来分解强度矩阵,但有些算法的总治疗时间过长。Bortfeld、Engel 等人^[9,10]提出的算法能够获得最小的总机器跳数(TNMU),但会产生更多的子

收稿日期: 2016-02-03; 修回日期: 2016-04-06 基金项目: 国家自然科学基金面上资助项目(NSFC11371143)

作者简介: 齐东洲(1990-),男,江苏淮安人,硕士研究生,主要研究方向为符号计算;吴敏(1976-),女(通信作者),福建永定人,副教授,博士,主要研究方向为符号计算、自动推理、可信计算(mwu@sei.ecnu.edu.cn);刘晓慧(1992-),女,内蒙古赤峰人,硕士研究生,主要研究方向为妇科;陈宏亮(1988-),男,上海人,工程师,本科,主要研究方向为放射治疗系统;余彬和(1991-),男,江西南昌人,工程师,硕士研究生,主要研究方向为三维可视化。

野个数(NS),总治疗时间也不理想。Biedl 等人^[11]利用 fixed-parameter tractable 方法很大程度上提高了调强效率,但不适用于矩阵包含较大元素的情况。Taskin 等人^[12]提出的 mixed-integer programming 技术,利用矩阵光圈(由顎构成而非多叶准直器)来分解强度矩阵。张鹏程等人^[13]提出的算法组合了 Engel 和 Luan 等人算法的优点,对子野个数和总机器跳数作了优化。景佳等人^[14]将 Galvin、Siochi、Bortfeld 等人的算法组合成一个模块,每次运行都需要运行这三个算法,从中挑出最优结果。近年来还出现了大强度矩阵分解问题^[15-17],即矩阵表示的大小超过了多叶准直器的最大张口范围。这类问题的解决思路在于将大强度矩阵分解成多个常规的邻接强度矩阵,使得每个小的邻接强度矩阵通过强度矩阵分解后被多叶准直器执行,因而本质上也是强度矩阵分解问题。Kamath 等人^[18]提出了一种方法将大强度矩阵分解为至多三个的邻接强度矩阵,但当相邻的强度矩阵不能准确对齐时,这种算法会产生热点问题(hotspots)。Wu 等人^[19]利用 k-连接路径来建模大矩阵分解问题,分解后的每个邻接强度矩阵采用 Engel 或 Kalinowski 算法^[20]进行强度矩阵分解。上述的强度矩阵分解算法,一些是追求最少的子野个数,另一些是寻找最小的总机器跳数,或是二者的权衡。而在实际应用中,总治疗时间还需额外考虑多叶准直器叶片的移动速度和验证记录(verification and recording, V&R)时间这两个因素,实验表明目前已有的算法存在总治疗时间偏长的问题,还有很大的提升幅度。

受 Xia & Verhey 算法^[6]的启发,本文综合考虑以上四个因素提出了一种高效的贪心算法。与之前的算法相比,尽管本文算法并不始终获得最少的总机器跳数或最少的子野个数,但却能权衡总机器跳数和子野个数并给出最优的子野排列顺序,从而保证总治疗时间最小。通过临床和随机数据测试,本文的贪心算法能以最少的时间来完成静态调强。

1 基本概念与表示

在本文中,仅考虑多叶准直器情况下的强度矩阵分解问题。

1.1 多叶准直器

调强放疗是通过多叶准直器来实现。多叶准直器是由几十对钨合金叶片组成,叶片左右移动从而构造出每一个子野的形状,如图 1 所示。多叶准直器的调强主要分为静态模式和动态模式。静态模式是指将射野分成多个均匀强度的子野,其技术称为分布照射,要求每次调整多叶准直器形状时要将射线关闭,且射线照射过程中多叶准直器的叶片不能移动;而在动态模式下,射线一直处于打开的状态,多叶准直器需要动态调整每一对叶片的形状。与动态模式相比,静态模式虽然治疗速度较慢,射线利用率低,但该方式操作更容易控制和实现,剂量分布更准确,从而更能降低对正常器官的伤害。本文中只探讨静态模式下的强度矩阵分解问题。

1.2 强度矩阵

强度矩阵是放疗计划系统中生成的一个中间结果,它是一个非负矩阵。本文中,矩阵的每一行对应着多叶准直器的一对叶片,矩阵的列数表示多叶准直器的张口范围。

1.3 强度矩阵分解

一般情况下,强度矩阵不能被多叶准直器使用,必须分解为多个 0-1 子强度矩阵(也叫做子野,segment),作为多叶

准直器的输入。0-1 子强度矩阵中的元素 1 表示在该位置上射线可以通过,即多叶准直器处于开口状态;反之,若元素为 0 则表示多叶准直器处于遮挡状态。由于多叶准直器的构造,0-1 子强度矩阵的每行中元素 1 必须是连续的。

强度矩阵分解成的每个 0-1 子强度矩阵称做子野。强度矩阵分解是指将强度矩阵 A 分解为多个机器跳数不同的子野,其形式化的描述为

$$A = \sum_{k=0}^n u_k \times S_k$$

其中: S_k 表示第 k 个子野, $u_k (> 0)$ 表示子野 S_k 的机器跳数(monitor unit, MU), n 表示子野个数 NS。特别地,有

$$S_k = S_k(i, j) = \begin{cases} 1 & \text{若该位置有射线通过} \\ 0 & \text{若该位置没有射线通过} \end{cases}$$

例如,当子野矩阵是 $\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$ 时,对应的多叶准直器

形状如图 2 所示。

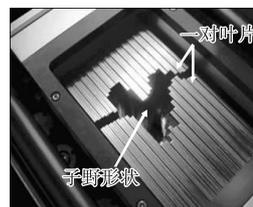


图 1 多叶准直器

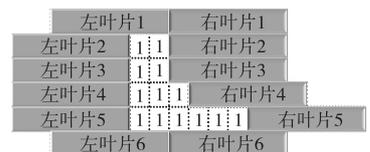


图 2 多叶准直器组合成的叶片形状

1.4 子野个数

强度矩阵分解后的子野数量 n 称为子野个数 NS。

1.5 总机器跳数

机器跳数(MU)是度量直线加速器照射剂量的单位,总机器跳数(TNMU)表示一个强度矩阵实施后所需的总照射剂量。给定一种直线加速器,它每秒输出的机器跳数是确定的,所以总机器跳数 $TNMU = \sum_{i=0}^n u_i$ 可用来衡量调强放疗照射总时间。

1.6 总治疗时间

静态模式调强的总时间主要取决于四个参数:总机器跳数、子野个数、多叶准直器叶片移动速度和验证记录(V&R)时间,其中总机器跳数和子野个数是两个关键参数。较多的总机器跳数不仅会延长照射时间,而且会增加多叶准直器间的射线泄漏,从而加大对正常器官的伤害;过多的子野个数会导致病人治疗时间的增加以及剂量计算的不准确性;叶片移动时间取决于子野个数与子野的排列顺序,优化子野排列的顺序,可以在一定程度上降低总治疗时间;验证记录时间通常是固定的。强度矩阵分解算法的执行时间一般为毫秒级,在计算总时间时可忽略不计。因此,减少总机器跳数与子野个数并优化子野顺序可以提高放疗准确性,提升放疗效率。

在给定强度矩阵的前提下,减少总机器跳数和子野个数显然是最主要的两个优化目标。目前大部分启发式强度矩阵分解算法的目标都是寻找尽可能少的总机器跳数和子野个数。然而,总治疗时间并非仅仅取决于这两个参数,子野的顺序也会影响治疗时间。在给定的多叶准直器叶片移动速率的前提下,当两个子野对应叶片之间的距离较远,子野的形状调整就会需要更多的时间,所以需要子野的执行顺序进行合理的排序。

为提供一种更加实际可行的强度矩阵分解算法,本文对影响调强放疗总治疗时间的四个因素进行了综合考虑。调强放疗的总治疗时间 T 定义如下:

$$T = \frac{\text{TNMU}}{D} + \sum_{i=2}^n \max \left\{ \text{V\&R}, \frac{\max \{ \text{dist}(x_i^j, x_{i-1}^j) \}}{v_{\text{leaf}}} \right\} \quad (1)$$

其中: D 表示剂量速率(dose rate); \max 表示取两个元素中的最大值; v_{leaf} 表示多叶准直器叶片移动速率; x_i^j 表示第*i*个子野的第*j*个叶片位置。为保证算法的可行性,本文采用了真实的机器参数,其中网格大小为1 cm,剂量速率*D*为200 MU/min,多叶准直器叶片移动速率为1.5 cm/s,验证记录(V&R)时间为4 s。

例1 为方便阐述,本文采用被普遍引用的基准强度矩阵 $A^{[10]}$ 。该强度矩阵的分解过程如下所示:

$$A = \begin{pmatrix} 4 & 5 & 0 & 1 & 4 & 5 \\ 2 & 4 & 1 & 3 & 1 & 4 \\ 2 & 3 & 2 & 1 & 2 & 4 \\ 5 & 3 & 3 & 2 & 5 & 3 \end{pmatrix} = 2 \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} + 2 \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} +$$

$$1 \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} + 1 \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} + 2 \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} + 1 \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (2)$$

根据分解结果,可得出总机器跳数 TNMU 为 10(= 2 + 2 + 1 + 1 + 1 + 2 + 1),子野个数 $NS = 7$,总调强放疗时间 $T = 27$ s (= 10/200 × 60 + 4 + 4 + 4 + 4 + 4 + 4)。

2 强度矩阵分解算法

2.1 Xia & Verhey 算法

常用的强度矩阵分解算法包括 Xia & Verhey、Bortfeld、Galvin 等,它们都采用以下的启发式算法框架:

```

R = A;
while (R ≠ 0) {
    找出子野矩阵 S 和对应的机器跳数 u (u > 0);
    R = R - u * S;
    输出 (u, S);
}
    
```

在迭代之前,先将原始强度矩阵赋值给剩余强度矩阵(residual intensity matrix) R ,接着在每一轮迭代中找到一个子野矩阵 S 和对应的机器跳数 u ,更新剩余强度矩阵 R ,即用 R 本身减去本轮的机器跳数与子野矩阵的乘积,并输出一轮迭代的结果 (u, S) 。继续迭代直至剩余强度矩阵 R 变成零矩阵。Xia & Verhey、Bortfeld 和 Galvin 算法的主要区别是如何选择机器跳数 u 。Galvin 算法中的 u 选取满足条件 $0.5 \times u(u+1) \geq L$ 的最小整数;Bortfeld 算法中的 u 接近于输入强度矩阵 A 中非零元素的平均数;Xia & Verhey 算法中的 $u = 2^{\lceil \log_2 L \rceil - 1}$ 。文献[21]中的实验表明,Xia & Verhey 算法通常可以给出较优的分解结果。Xia & Verhey 算法提供了 sliding window 和 reducing level 两种方法,首先找出分解的系数,然后逐步分解直到剩余矩阵变为零矩阵。Sliding window 方法的核心是先找到矩阵最左边非零两列的不同元素,然后计算分解系数;而 reducing level 方法是计算整个强度矩阵的不同元素,再找出整体分解的系数,逐级分解矩阵。Xia & Verhey 算法分解速度快,大部分情况下都能给出不错的分解结果,但 sliding window 方法没有对 NS 和 TNMU 两个参数优化,进而导致每一轮分解的 MU 和 NS 过多,

最终的 TNMU 和 NS 过大;而 reducing level 方法计算的 TNMU 太多,分解的子野开口较小且子野形状切换比较耗时。鉴于 sliding window 方法更加高效,本文采用其基本框架并作出改进。在每一轮分解中不仅优化了 NS 和 TNMU 两个核心参数,并对最终子野的排列顺序作了优化。

2.2 两列强度矩阵分解算法

2.2.1 单列强度矩阵分解

多叶准直器是由多对平行移动的叶片组成,每一对叶片都是相对独立地左右移动,所以强度矩阵分解问题可以归为单列强度矩阵分解问题,再将各行的分解结果组合成需要的结果。Bortfeld 和 Luan 算法正是采用这种方式。同样地,强度矩阵的分解也可从矩阵列方向来考虑,即将一列强度矩阵分解为多个子野矩阵。例如,例1中的第一列 $[4 \ 2 \ 2 \ 5]^T$ 有三个不同的数字 2、4、5,从而可用数组 4、2、1 或 2、2、1 或 2、1、1 来表示。

$$\begin{pmatrix} 4 \\ 2 \\ 2 \\ 5 \end{pmatrix} = 4 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + 2 \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} + 1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (3)$$

$$\begin{pmatrix} 4 \\ 2 \\ 2 \\ 5 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + 2 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + 1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (4)$$

$$\begin{pmatrix} 4 \\ 2 \\ 2 \\ 5 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + 1 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + 1 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + 1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (5)$$

上述分解式(3)~(5)的子野个数分别为 3、3、4,总机器跳数分别为 7、5、5。显然单列强度矩阵可以分解为多种不同的子野。为减少总机器跳数与子野个数,需要找到一组系数,使得它们的数量最少且系数之和最小。这一问题形式化描述为:给定单列强度矩阵 C ,将 C 表示成矩阵 B 及列向量 D 的乘积。

$$C = B \times D \quad \text{or}$$

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_i \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1j} \\ b_{21} & b_{22} & \cdots & b_{2j} \\ \vdots & \vdots & & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{ij} \end{bmatrix}_{i \times j} \times \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_j \end{bmatrix}_{j \times 1} \quad (6)$$

其中: B 为 0-1 矩阵,向量 D 为分解的系数向量, i 表示单列强度矩阵的行数, j 表示子野个数,总机器跳数等于向量 D 的所有元素之和,即 $\sum_{k=0}^j d_k$ 。为加快计算,需要去除单列矩阵中的重复元素,如例1中第一列向量 C 为 $[2 \ 4 \ 5]^T$,根据之前的计算,例1中强度矩阵 C 的分解向量 D 可为 $[4, 2, 1]^T, [2, 2, 1]^T$ 或者 $[2, 1, 1, 1]^T$,其中向量 $[2, 2, 1]^T$ 具有最少的总机器跳数和子野个数,所以是最优的结果。

命题1 若单列强度向量 C_{i1} 可写成 0-1 矩阵 B_{ij} 与列向量 D_{j1} 的乘积(如式(6)所示),为了减少总机器跳数与子野个数, j 应该满足如下条件:

$$\log_2(i+1) \leq j \leq \min \{ i, (\lfloor \log_2 L \rfloor + 1) \}$$

其中: i 表示向量 C 中的元素个数; j 表示向量 D 中的元素个数; L 表示向量 C 中的最大元素。

证明 a) 给定 j 个不同的数,它们最多可以表示 $2^j - 1$ (= $C_j^1 + C_j^2 + \cdots + C_j^j$) 个数,而此个数须大于单列强度矩阵 C 的元素个数,于是 $2^j - 1 > i$,所以 $j \geq \log_2(i+1)$ 。

b) 给定一个 j 位二进制数 $b_j b_{j-1} \dots b_1$, 它能表示数的范围为 $0 \sim (2^j - 1)$, 如一个三位二进制数可以表示 $0 \sim 7$ 中的任何数字。当 $j = (\lfloor \log_2 L \rfloor + 1)$ 时, j 位二进制数能表示的范围是 $0 \sim (2^{(\lfloor \log_2 L \rfloor + 1)} - 1)$, 而 $(2^{(\lfloor \log_2 L \rfloor + 1)} - 1) \geq L$, 故 j 的上界为 $(\lfloor \log_2 L \rfloor + 1)$, 即 $j \leq (\lfloor \log_2 L \rfloor + 1)$ 。又因为表示 i 个不同的元素最多只需要 i 个元素, 所以 $j \leq \min\{i, (\lfloor \log_2 L \rfloor + 1)\}$ 。证毕。

2.2.2 两列强度矩阵分解

为尽可能减少总机器跳数和子野个数, 在强度矩阵分解时应尽可能考虑多列, 但每次最多只能考虑两列。例如, 一个行矩阵 $(1\ 0\ 1)$ 不能用一个子野来表示, 而要拆成两个子野, 这样反而会产生更多的子野。因此在进行强度矩阵分解的时候, 本文同时考虑两列。两列强度矩阵分解问题可以归结为单列强度矩阵分解问题, 此时只需将两列中所有不同元素组成向量 C , 然后应用式(6)求解即可。

2.3 强度矩阵分解算法

强度矩阵分解算法包含数据结构和算法两个部分。数据结构部分描述了子野的一些关键属性, 算法部分拆分为多个步骤, 每个步骤描述了强度矩阵的详细分解过程。

2.3.1 数据结构

子野的数据结构包含两部分: 二维矩阵 matrix, 如例 1 中的 0-1 矩阵, 以及机器跳数 mu, 如例 1 中的 0-1 矩阵的系数。

```
struct segment {
    matrix matrix,
    integer mu,
};
```

2.3.2 算法描述

a) 计算满足式(6)的向量 D 。

输入: 向量 C (不包括重复元素)。

输出: 满足式(6)的 D 向量数组 ArrDs。

说明: ArrDs 表示向量数组, 数组中每个元素是向量 D 。

根据命题 1 计算出 j 的范围, 不失一般性, 设 j 的范围是 $[a, b]$;

for $j = a$ to b do {

if (c_m 可表示成 j 个整数和的形式) then {

将每个形式下的 j 个整数表示成一个向量 D , 并将这些向量 D 加入到向量数组 ArrDs 中;

遍历 ArrDs 并删除不满足式(6)的向量 D ;

if (ArrDs 不为空) then { 退出循环; }

例 2 对于例 1 的强度矩阵 A , 根据命题 1, 可计算出 j 的范围是 $[3, 3]$ 。矩阵 A 的前两列有四个不同元素, 将它们组成向量 $C(2, 3, 4, 5)^T$, 则满足式(6)的 ArrDs 包含两个向量 $D: (1, 1, 3)^T, (1, 2, 2)^T$ 。

b) 计算子野数组 ArrSegs 和剩余矩阵 R 。

输入: 向量 D 及剩余矩阵 R 。

输出: 子野数组 ArrSegs 和剩余矩阵 R 。

说明: NRow, NCol 分别表示强度矩阵的行数与列数。

S 表示一个子野 (segment);

ArrSegs 表示子野数组;

D . first, D . last 表示矩阵 D 的第一个和最后一个元素;

将向量 D 中的元素按照从大到小的顺序排序;

for $k = D$. first to D . last do {

初始化子野 S ;

S . mu = k ;

for $i = 0$ to NRow do {

在剩余矩阵 R 的第 i 行找出满足条件“ $x \geq k$ ”的第一个元素下标 m (x 表示矩阵 R 中的元素);

for $j = m$ to NCol do {

if ($R[i][j] \geq k$) then { S . matrix[i][j] = 1; }

else { 退出本层循环; }

}}

将子野 S 加入到子野数组 ArrSegs 中;

$R = R - S$. matrix; }

例 3 给定向量 $D(1, 2, 2)^T$, 先将其按从大到小排序, 得到向量 $D = (2, 2, 1)^T$ 。初始化时, 剩余矩阵 R 等于强度矩阵 A 。根据步骤 b), 执行第一次循环后,

$$\text{第一个子野 } S_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \mu = 2$$

$$\text{剩余矩阵 } R_1 = \begin{pmatrix} 4 & 5 & 0 & 1 & 4 & 5 \\ 2 & 4 & 1 & 3 & 1 & 4 \\ 2 & 3 & 2 & 1 & 2 & 4 \\ 5 & 3 & 3 & 2 & 5 & 3 \end{pmatrix} - 2 \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 0 & 1 & 4 & 5 \\ 0 & 2 & 1 & 3 & 1 & 4 \\ 0 & 1 & 0 & 1 & 2 & 4 \\ 3 & 1 & 1 & 0 & 3 & 1 \end{pmatrix}$$

执行第二次循环后

$$\text{第二个子野 } S_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \mu = 2$$

$$R_2 = \begin{pmatrix} 2 & 3 & 0 & 1 & 4 & 5 \\ 0 & 2 & 1 & 3 & 1 & 4 \\ 0 & 1 & 0 & 1 & 2 & 4 \\ 3 & 1 & 1 & 0 & 3 & 1 \end{pmatrix} - 2 \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 & 4 & 5 \\ 0 & 0 & 1 & 3 & 1 & 4 \\ 0 & 1 & 0 & 1 & 0 & 2 \\ 1 & 1 & 1 & 0 & 3 & 1 \end{pmatrix}$$

执行第三次循环后,

$$\text{第三个子野 } S_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}, \mu = 1$$

$$R_3 = \begin{pmatrix} 0 & 1 & 0 & 1 & 4 & 5 \\ 0 & 0 & 1 & 3 & 1 & 4 \\ 0 & 1 & 0 & 1 & 0 & 2 \\ 1 & 1 & 1 & 0 & 3 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 4 & 5 \\ 0 & 0 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 3 & 1 \end{pmatrix}$$

步骤 b) 执行结束时, 子野数组 ArrSegs 包含三个子野 S_1, S_2, S_3 , 剩余矩阵 R 等于 R_3 。从示例中可以看出, 经过步骤 b) 的执行, 可以保证剩余矩阵 R 前两列的元素全部变成 0 元素。在进行每一行矩阵分解的过程中, 算法不仅考虑了矩阵的前两列, 如子野 S_1 的最后一行的所有列都变成 1, 这个原因主要是为了让多叶准直器的开口尽可能地大。这样不但不影响前两列的分解, 而且会降低接下来列的复杂度, 有利于减少子野个数。例如剩余矩阵 R_3 , 经过一轮循环, 前三列元素全部都变成了 0, 如果仅考虑前两列, 第三列的元素就会保持原值。因此, 在保证不影响前两列全部变成 0 的情况下, 每行计算时要使 1 的连续个数尽可能多。

c) 强度矩阵分解的贪心算法。

本文给出了如下四种贪心策略:

策略 1 选择 ArrDs 中的一个元素, 它能使剩余矩阵 R 接下来两列中的不同元素个数最少。若相同, 则选择第一个向量 D 。

策略 2 选择 ArrDs 中的一个元素, 它能使剩余矩阵 R 接下来两列中的元素最大值 L 最小。若相同, 则选择第一个向量 D 。

策略 3 选择 ArrDs 中的一个元素, 它能使剩余矩阵 R 接下来两列中的所有元素之和最小。若相同, 则选择第一个向量 D 。

策略 4 选择 ArrDs 中的一个元素, 它能使剩余矩阵 R 接下来两列中零元素的个数最多。若相同, 则选择第一个向量 D 。

由命题 1 可知,向量 D 元素个数 j 的取值范围依赖于向量 C 中元素个数 i 和向量 C 中最大元素 L ,且 i 越小或者 L 越小,对应的 j 范围越小。所以它们可以作为贪心策略的选择依据,分别对应着贪心策略 1 和 2。当剩余矩阵接下来两列的 i 或者 L 值相同时,策略 1 或者 2 就无法发挥自身的贪心作用,所以补充了策略 3、4。剩余矩阵 R 接下来两列中所有元素之和最小或者零元素个数最多都可以表示剩余矩阵 R 分解程度比较高。

输入:强度矩阵 A ,一种贪心策略 G 。

输出:子野数组和总治疗时间。

$R = A$;

for $i = 0$ to N_{Col} do {

 第 i 和 $i + 1$ 列的所有不同元素组成向量 C ;

 利用步骤 a) 计算出 $ArrDs$;

 根据给定的贪心策略 G 从 $ArrDs$ 中选出一个向量 D ;

 利用步骤 b) 计算出子野数组 $ArrSegs$ 并更新剩余矩阵 R ;

$i + = 2$;}

根据式(1)计算出调强放疗的总治疗时间。

例 4 采用例 1 中的强度矩阵 A 和步骤 c) 中的贪心策略作为输入。

第一轮 for 循环。向量 $C = (2, 3, 4, 5)^T$, 根据步骤 a) 得出 $ArrDs$ 有两个向量 $D_1 = (1, 1, 3)^T$ 、 $D_2 = (1, 2, 2)^T$ 。 D_1 、 D_2 对

应的剩余矩阵分别是 $R_{D1} = \begin{pmatrix} 0 & 0 & 0 & 1 & 4 & 5 \\ 0 & 0 & 0 & 2 & 0 & 3 \\ 0 & 0 & 2 & 1 & 2 & 4 \\ 0 & 0 & 0 & 2 & 5 & 3 \end{pmatrix}$, $R_{D2} = \begin{pmatrix} 0 & 0 & 0 & 1 & 4 & 5 \\ 0 & 0 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 3 & 1 \end{pmatrix}$ 。

根据贪心策略 1, 由于 R_{D1} 、 R_{D2} 的第三列和第四列不同元素个数都是 3, 所以选择第一个向量 $D_1 = (1, 1, 3)^T$; 根据贪心策略 2, 由于 R_{D1} 、 R_{D2} 的第三列和第四列最大元素也相同, 则仍选择向量 $D_1 = (1, 1, 3)^T$; 根据贪心策略 3, 因为 R_{D1} 、 R_{D2} 的第三列和第四列元素之和分别为 8、4, 所以选择 $D_2 = (1, 2, 2)^T$; 根据贪心策略 4, 因为 R_{D1} 、 R_{D2} 的第三列和第四列元素中零元素个数分别是 3、5, 所以选择 $D_2 = (1, 2, 2)^T$ 。然后利用步骤 b) 计算出子野数组 $ArrSegs$, 并更新剩余矩阵 R 。接下来的迭代依次类推。

d) 根据输入的贪心策略数组, 找出总治疗时间最少的强度矩阵分解结果。

输入: 贪心策略数组。

输出: 总治疗时间最少的强度矩阵分解 $ArrSegs$ 。

for 策略 1 to 策略 4 do {

 将强度矩阵 A 和贪心策略 G 传给步骤 c), 计算出总治疗时间; }

比较所有策略的总治疗时间, 找出最小值;

输出对应的强度矩阵分解结果。

3 实验

为了测试本文提出算法的性能, 下面将新算法与之前的 Galvin、Bortfeld、Xia & Verhey、Siochi 及 Luan 算法进行对比。为了验证算法的可靠性及鲁棒性, 本文采用了不同类型肿瘤病人的数据(如表 1 的第一列所示), 包括脑组织(brain)、头颈部

表 1 基于临床数据的各算法比较结果

数据	No	matrix L	比较项	Galvin	Bortfeld (sweep)	Bortfeld (close-in)	Xia (reducing level)	Xia (sliding window)	Siochi	Luan	New
Brain	1	17 × 62 103	NS	70	127	134	59	63	127	64	59
			TN MU	466	177	177	563	386	177	221	280
			T	1 515. 13	869. 1	1 479. 1	1 208. 9	1 388. 47	869. 1	1 630. 97	614
	2	18 × 56 106	τ	2. 47	1. 42	2. 41	1. 97	2. 26	1. 42	2. 66	1
			NS	49	95	96	42	38	95	56	38
			TN MU	360	106	106	311	148	106	198	199
HN	3	11 × 54 29	T	853. 333	579. 8	684. 467	655. 967	561. 067	579. 8	960. 067	267. 033
			τ	3. 2	2. 17	2. 56	2. 46	2. 1	2. 17	3. 6	1
			NS	27	46	50	27	26	46	33	26
	4	12 × 46 26	TN MU	112	54	54	117	62	54	103	68
			T	530. 267	400. 2	558. 2	489. 1	445. 933	400. 2	631. 567	251. 733
			τ	2. 11	1. 59	2. 22	1. 94	1. 77	1. 59	2. 51	1
lung	5	10 × 2 183	NS	19	25	24	15	14	25	20	17
			TN MU	64	26	26	63	31	26	54	32
			T	263. 2	154. 467	149. 133	235. 567	198. 633	154. 467	179. 533	95. 6
	6	28 × 55 32	τ	2. 75	1. 62	1. 56	2. 46	2. 08	1. 62	1. 88	1
			NS	32	56	51	25	29	56	31	25
			TN MU	257	216	216	355	339	216	373	308
oropharynx	7	10 × 2 183	T	282. 433	305. 467	292. 8	288. 5	274. 367	305. 467	296. 567	190. 4
			τ	1. 48	1. 6	1. 54	1. 52	1. 44	1. 6	1. 56	1
			NS	40	65	65	30	33	65	40	32
	8	28 × 55 32	TN MU	201	71	71	166	93	71	124	80
			T	930. 3	781. 967	841. 967	670. 467	688. 567	781. 967	1 027. 2	497. 333
			τ	1. 87	1. 57	1. 69	1. 35	1. 38	1. 57	2. 07	1
prostate	9	28 × 58 42	NS	38	56	56	30	30	56	37	31
			TN MU	183	59	59	169	90	59	113	73
			T	704. 9	638. 367	774. 367	545. 367	581. 667	638. 367	735. 233	331. 233
	10	29 × 58 52	τ	2. 13	1. 93	2. 34	1. 65	1. 76	1. 93	2. 22	1
			NS	36	55	54	30	28	55	36	29
			TN MU	194	59	59	160	84	59	100	74
prostate	9	10 × 34 23	T	673. 533	602. 367	711. 7	574. 667	530. 533	602. 367	773. 333	302. 867
			τ	2. 22	1. 99	2. 35	1. 9	1. 75	1. 99	2. 55	1
			NS	28	30	29	28	22	30	24	20
	10	10 × 36 98	TN MU	95	30	30	84	46	30	55	38
			T	329. 167	157. 667	172. 333	273. 867	200. 467	157. 667	289. 833	106. 733
			τ	3. 08	1. 48	1. 61	2. 57	1. 88	1. 48	2. 72	1
10	10 × 36 98	NS	53	81	80	40	39	81	45	37	
		TN MU	368	101	101	346	148	101	151	144	
		T	565. 067	394. 967	407. 633	440. 467	395. 067	394. 967	466. 633	197. 867	
τ	2. 86	2	2. 06	2. 23	2	2	2. 36	1			

(head and neck, HN)、肺器官(lung)、口咽部(oropharynx)和前列腺(prostate)等临床强度矩阵和随机生成的强度矩阵进行测试。表 1 中分别列出了总机器跳数(TNMU)、子野个数(NS)、总治疗时间(T)和加速比(τ)几个参数。为了量化算法的提程度,以本文算法的速度作为基准 1,列出了本文算法与其他算法的运行速度之比,即两者总治疗时间之比的倒数。

$$\text{速度比 } \tau = \frac{V_{\text{new}}}{V_{\text{other}}} = \frac{T_{\text{other}}}{T_{\text{new}}} \quad (7)$$

其中: V_{new} 、 V_{other} 分别表示本文提出算法和其他算法的速度; T_{new} 、 T_{other} 分别表示本文算法和其他算法的总治疗时间。限于篇幅问题,表 1 中仅列出临床数据的部分测试结果,其中 L 表示强度矩阵中的最大值,New 表示本文的算法, τ 表示新旧算法的速度比,即新算法的速度是对比算法的 τ 倍。

从表 1 可以看出,尽管本文算法计算的子野个数(NS)和总机器跳数(TNMU)并不总是最小,但是它们两者的组合使得子野开口尽可能大且子野顺序得到优化,从而能取得最小的总治疗时间。如表格的第一组取自脑癌患者数据,矩阵大小为 17×62 ,最大元素 L 为 103,本文算法的子野个数为 59 个,总机器跳数为 280,总治疗时间为 614 s。以新算法为单位基准,速度是 Luan 算法的 2.66 倍。实验结果表明在临床数据和随机数数据测试下,本文算法与同类算法相比都能取得最优的结果。

本实验的测试平台为 Window 操作系统(Core i7-3610QM CPU, 2.30 GHz, RAM 8 GB, 64 bit Window 7 Operation System)。每个强度矩阵分解的平均时间为 0.1 s,所以本文算法在实际中可行。

4 结束语

针对静态调强方式,本文提出了利用多种贪心策略进行强度矩阵分解的新算法。尽管本文算法与 Galvin、Bortfeld、Xia & Verhey、Siochi 及 Luan 算法相比,不是每次都计算出最少的子野个数和总机器跳数,但本文算法可以获得最小的总治疗时间。通过临床强度矩阵和随机强度矩阵的大量测试,本文算法不但能够在较短的时间内完成强度矩阵分解问题,而且能够找出最优方案,这将进一步提高放疗治疗的效率,降低放疗的成本。效率的提高意味着病人治疗时间缩短,从而降低病人长时间照射带来的风险。因此,本文提出的算法在放疗中将具有更加良好的应用前景。

参考文献:

- [1] 王健. 调强放疗强度矩阵分解研究[D]. 武汉:华中科技大学, 2011.
- [2] Webb S. Intensity-Modulated Radiation Therapy[M]. [S. l.]:CRC Press, 2001.
- [3] Bansal N, Coppersmith D, Schieber B. Minimizing setup and beam on times in radiation therapy[C]//Proc of International Workshop on Approximation Algorithms for Combinatorial Optimization Problem, 2006:27-38.
- [4] Baatar D, Hamacher W, Ehrgott M. *et al.* Decomposition of integer matrices and multileaf collimator sequencing[J]. *Discrete Applied Mathematics*, 2005, 152(1-3):6-34.
- [5] Galvin J M, Chen Xuangen, Smith R M. Combining multi leaf field to modulate fluence distributions[J]. *International Journal of Radiation Oncology Biology Physics*, 1993, 27(3):697-705.
- [6] Xia Ping, Verhey L. Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple static segments[J]. *Medical Physics*, 1998, 25(8):1424-1434.
- [7] Siochi R A C. Minimizing static intensity modulation delivery time using an intensity solid paradigm[J]. *International Journal of Radiation Oncology Biology Physics*, 1999, 43(3):671-680.
- [8] Luan Shuang, Saia J, Young M J. Approximation algorithms for minimizing segments in radiation therapy[J]. *Information Processing Letters*, 2007, 101(6):239-244.
- [9] Bortfeld T, Kahler D L, Waldron T, *et al.* X-ray field compensation with multileaf collimators[J]. *International Journal of Radiation Oncology Biology Physics*, 1994, 28(3):723-730.
- [10] Engel K. A new algorithm for optimal multileaf collimator field segmentation[J]. *Discrete Applied Mathematics*, 2005, 152(1-3):35-51.
- [11] Biedl T C, Durocher S, Engelbeen C, *et al.* Faster optimal algorithms for segment minimization with small maximal value[J]. *Discrete Applied Mathematics*, 2013, 161(3):317-329.
- [12] Taskin Z C, Smith J C, Romeijn H E. Mixed-integer programming techniques for decomposing IMRT fluence maps using rectangular apertures[J]. *Annals of Operations Research*, 2012, 196(1):799-818.
- [13] 张鹏程, 朱健, 李元全, 等. 一种改进的多叶准直器野分割算法[J]. *东南大学学报:自然科学版*, 2012, 42(5):875-879.
- [14] 景佳, 曹瑞芬, 李国丽, 等. 基于多算法比较的多叶准直器叶片序列优化[C]//第 2 届全国核技术及应用研究学术研讨会大会论文集摘要集, 2009.
- [15] Dogan N, Leybovich L B, Sethi A, *et al.* Automatic feathering of split fields for step-and-shoot intensity modulated radiation therapy[J]. *Physics in Medicine Biology*, 2003, 48(9):1133-1140.
- [16] Hong Linda, Alektiar K, Chui C, *et al.* IMRT of large fields: Whole-abdomen irradiation[J]. *International Journal of Radiation Oncology Biology Physics*, 2002, 54(1):278-289.
- [17] Wu Qiuwen, Arnfield M, Tong Shidong, *et al.* Dynamic splitting of large intensity-modulated fields[J]. *Physics in Medicine Biology*, 2000, 45(7):1731-1740.
- [18] Kamath S, Sahni S, Ranka S, *et al.* Optimal field splitting for large intensity-modulated fields[J]. *Medical Physics*, 2004, 31(12):3314-3323.
- [19] Wu Xiaodong, Dou Xin, Bayouth J E, *et al.* An almost linear time algorithm for field splitting in radiation therapy[J]. *Computational Geometry*, 2013, 46(6):673-687.
- [20] Kalinowski T. An algorithm for optimal multileaf collimator field segmentation with interleaf collision constraint[J]. *Discrete Applied Mathematics*, 2005, 152(1-3):52-88.
- [21] Que W. Comparison of algorithms for multileaf collimator field segmentation[J]. *Medical Physics*, 1999, 26(11):2390-2396.