

# 考虑设备转换成本的 MRI 检查预约调度优化

林 晖<sup>1</sup>, 王 杉<sup>2</sup>

(1.上海交通大学 安泰经济与管理学院, 上海 200030; 2.中山大学 管理学院, 广东 广州 510275)

**【摘要】**核磁共振检查 (MRI) 是现代医学影像诊断的重要手段之一。MRI 设备在扫描不同部位时会产生转换成本, 为同时提高设备的利用率和患者的满意度, 建立了马尔可夫决策过程 (Markov Decision Process) 模型, 以最小化医院的长期成本。该模型是首个对 MRI 设备转换成本和患者日间等候成本进行权衡的优化模型。通过策略迭代算法可得到该模型的最优调度; 为降低计算复杂度, 设计了基于单日规则、开放获取规则和短视规则的多项式时间算法; 除此之外, 通过忽略容量约束, 提出了分解算法, 在 MRI 设备检查能力紧张或充足时, 其表现与最优解接近。通过对比上海某大型综合医院的实际排程, 验证了上述算法能在检查能力、患者拒绝比例、患者平均等待天数、日平均检查类型数等指标上取得显著改善。

**关键词:** 医疗卫生; 预约调度; 马尔可夫决策过程; 组合优化; 近似算法

中图分类号: F272

文献标识码: A

## On Scheduling MRI Appointments with Changeover Cost

LIN Hui<sup>1</sup>, WANG Shan<sup>2</sup>

(1. Antai college of Economics & Management, Shanghai Jiao Tong University, Shanghai 200030, China; 2. School of Business, Sun Yat-sen University, Guangzhou 510275, China)

**【Abstract】** Magnetic resonance imaging (MRI) is one of the important methods of modern medical imaging diagnosis. MRI equipment will incur changeover costs when scanning different organs. To improve the utilization of the equipment and the satisfaction of patients, a Markov Decision Process model was established to minimize the long-term cost. This optimization model is the first one to weigh the changeover cost and the inter-day waiting cost. The optimal schedule can be obtained by Policy Iteration algorithm. To reduce the computational complexity, polynomial time algorithms based on Single Day policy, Open Access policy and Myopic policy are designed. In addition, by ignoring the capacity constraint, the Decomposition algorithm is proposed, and the performance is close to the optimal solution when the inspection capacity is tight or sufficient. Comparing to the actual scheduling of a general hospital in Shanghai, the above algorithms can achieve significant improvement in terms of examination capacity, rejection rate, average waiting, and average number of daily examination types.

**Key words:** Healthcare; Appointment Scheduling; Markov Decision Process; Combination Optimization; Approximation Algorithm

收稿日期: 2023-03-30 修订日期: 2023-02-08

基金项目: 国家自然科学基金资助项目 (72001220, 71931008)

作者简介: 林晖 (1999-), 女, 博士生。研究方向为医疗管理与运营。

通信作者: 王杉 (1991-), 女, 助理教授。E-mail: wangsh337@mail.sysu.edu.cn

核磁共振检查 (MRI) 是医生确认患者病情的重要手段<sup>[1]</sup>, 目前 MRI 检查的等待时间长达 16~36 天<sup>[2]</sup>, 成为了患者得到确诊并治疗的瓶颈环节。一方面, MRI 设备昂贵, 平均 500 万元一台, 最高可达 2000 万元, 是医院的稀缺资源, 因此无法简单地通过增加设备数量来应对快速增长的 MRI 检查需求<sup>[3]</sup>。另一方面, MRI 检查需针对检查部位使用特定的线圈, 一般最常用的线圈就有近十种<sup>[4]</sup>。当前后两名患者检查部位不同时, 更换线圈会产生较长的装卸时间和线圈损耗<sup>[3]</sup>。2018 年的一项研究<sup>[5]</sup>显示, 更换 MRI 线圈的平均时间长达 13.4 分钟, 占平均检查时间的 55%, 减少线圈更换次数可显著提高 MRI 设备的利用率。另外, 一台 MRI 设备的线圈成本可高达 32 万美元<sup>[6]</sup>, 而装卸线圈需要医务人员十分严格且精细地操作<sup>[4]</sup>。已有文献表明, 频繁更换线圈会加速线圈损耗、导致线圈故障, 带来高昂的维修甚至重新购置的费用<sup>[4,7]</sup>。在实践中, 许多医院采用先到先服务的预约规则<sup>[8,9]</sup>, 但由于患者检查需求的异质性, 大大增加了线圈的装卸次数, 产生了非常高的转换成本。目前, 已有医院通过建立 MRI 预约规则来减少更换线圈的次数<sup>[4]</sup>, 比如使用固定的日期或时间段来检查特定部位<sup>[2,4]</sup>, 从而降低更换线圈带来的转换成本。但这又会影响患者预约的灵活性, 使得 MRI 设备空置、患者长时间等待, 导致医疗资源的浪费和患者满意度的降低。因此, 如何对 MRI 检查预约进行合理地调度, 在及时满足患者需求的同时, 降低更换线圈带来的转换成本, 是一个亟待解决的重要问题。

关于医院预约调度的文献有很多, 通常以尽快满足患者需求为目标, 同时考虑所消耗的资源成本。许多文章考虑了患者爽约<sup>[10-14]</sup>、患者到达不及时<sup>[15]</sup>、未预约患者 (walk-in) <sup>[16]</sup>、随机服务时间<sup>[17]</sup>、医患匹配度<sup>[18]</sup>等因素, 但很少考虑到 MRI 设备的特殊性, 即服务不同类型患者时会产生转换成本<sup>[3]</sup>。潘兴薇, 耿娜<sup>[9]</sup>根据某三甲医院的实际调研数据, 建立了 MRI 检查预约仿真模型, Geng 等<sup>[19]</sup>对 MRI 预约调度建立了马尔可夫决策过程模型, 但都没有考虑 MRI 设备的转换成本。Qiu 等<sup>[3]</sup>考虑了 MRI 设备在不同类型患者间的转换成本, 在一个工作日内优化患者的检查次序和检查时间, 但没有考虑跨工作日的预约排程, 而现实中患者很少能在预约当天进行 MRI 检查<sup>[2]</sup>。其他与 MRI 设备调度相关的文献则通过仿真方法对调度规则进行检验, 没有通过数学建模的方式对问题进行分析和求解<sup>[2,20]</sup>。因此, 已有的研究成果并不能够被应用到 MRI 检查的日间预约调度 (inter-day scheduling) 问题上, 同时实现降低 MRI 设备转换成本和及时满足患者需求这两个目标。

考虑到 MRI 检查预约给理论和实践带来的挑战, 本文将研究如何对 MRI 检查需求进行预约调度, 来最小化更换线圈产生的转换成本以及患者等待或被拒绝的成本。针对该问题, 本文建立了马尔可夫决策过程 (Markov Decision Process) 模型<sup>[8,21,22]</sup>, 通过策略迭代 (Policy iteration) 算法<sup>[23]</sup>得到医院的最优排程。为了方便医院实际应用, 本文提出了三种预约调度规则: 单日规则 (Same Day Policy)、开放获取规则 (Open Access Policy) 和短视规则 (Myopic Policy)。但由于转换成本的引入, 即使给定调度规则, 该问题仍是一个复杂的组合优化问题。因此本文分析了各调度规则下的问题性质, 设计并证明了最优解算法。除此之外, 本文还依据分而治之 (Divide and Conquer) 的思想提出了忽略容量约束的分解算法 (Decomposition Algorithm) <sup>[21]</sup>。通过数值实验发现, 在 MRI 设备的检查能力紧张或患者等候成本高时, 即单日容量较小或等待成本较大时, 最优解倾向于不开放未来工作日的预约, 以避免增加等待成本, 此时采用单日规则可实现最优调度。当医院希望尽快满足患者的检查需求或 MRI 设备转换耗时较短时, 即拒绝成本较高或转换成本较低时, 开放获取规则与最优解的性能差距

仅为 1.29%。在患者等待成本较低时,可以采用短视规则算法来将部分患者安排到后续工作日,从而降低 MRI 设备转换成本和拒绝成本,此时与最优解的性能差距为 10.75%。在 MRI 设备的检查能力紧张或非常充足时,医院可以忽略检查能力约束,对每个类型的病人单独通过策略迭代算法进行调度,再通过本文提出的分解算法来构造符合检查能力约束的排程,此时与最优解仅有不超过 1.5%的性能差距。通过使用上海某大型综合医院的实际数据进行模拟,并对比医院的实际运营情况,我们发现,若使用本文设计的规则和算法,能在检查容量配置、患者拒绝比例、患者平均等待天数、日平均患者类型数等指标上取得显著改善。

本文的贡献主要有三点。第一,本文所建立的模型具有较强的一般性,比如仅需假设需求具有马尔可夫性、转换成本和等待成本单调递增即可,能广泛应用到不同环境和场景中。第二,本文设计了易于应用的调度规则与算法,还对不同规则的应用效果及其所适用的场景进行了比较和分析,为医院 MRI 预约调度提供了管理洞见。第三,过往文献在模型建立时没有考虑到 MRI 设备的转换成本,本文同时考虑了设备转换成本、患者等待成本和被拒绝成本,除了直接用于 MRI 检查场景,还可用于其他具有类似转换成本的预约调度问题中,填补了预约调度文献在这方面的空白。

## 1 模型构建与求解

### 1.1 数学模型

该模型以一个工作日为一期<sup>[13]</sup>,为当期所有抵达的检查需求安排服务日期,或者,拒绝某些需求<sup>[24]</sup>。可预约的时间范围为 $T$ 期(如两周),共有 $J$ 类检查需求(不同检查所需线圈不同)。在每期初,医院面临上一期末确定的预约表 $S^t = \{s_{i,j}^t\}$ 和当期新到达患者的检查需求 $D^t = \{d_j^t\}$ 。其中,预约表 $S^t$ 为 $T \times J$ 矩阵, $s_{i,j}^t$ 表示在第 $t$ 期开始前,已被安排在第 $t+i-1$ 期的 $j$ 类检查的数量。检查需求 $D^t$ 为 $J$ 维行向量, $d_j^t$ 表示在第 $t$ 期抵达的 $j$ 类检查需求量(本文不对需求的分布进行任何假设,只需它具有马尔可夫性质即可)。在给定预约表的基础上,医院需要为新到达的需求安排检查日期 $A^t = \{a_{i,j}^t\}$ ,以最小化医院的长期总成本。排程 $A^t$ 为 $(T+1) \times J$ 矩阵,当 $1 \leq i \leq T$ 时, $a_{i,j}^t$ 表示将 $a_{i,j}^t$ 个 $j$ 类检查安排在第 $t+i-1$ 期;当 $i = T+1$ 时, $a_{T+1,j}^t$ 表示被拒绝的 $j$ 类检查需求的数量。

由于 MRI 设备检查不同类型的患者时,需要更换线圈,这会造成一定的时间损耗,从而产生转换成本。而未能及时安排 MRI 检查会降低患者满意度,甚至导致患者病情恶化。因此,医院需权衡转换成本与患者等待或被拒绝的成本。我们对成本函数做出以下假设:

**假设 1** 若在同一天工作日内有 $n$ 种不同的检查类型,则产生转换成本 $f_c(n)$ ,是 $n$ 的一般性单调递增函数;若一个患者从提出需求到接受检查需等待 $m$ 天,则产生患者等待成本 $f_w(m)$ ,是 $m$ 的一般性单调递增函数;每拒绝一位患者产生拒绝成本 $c_r$ <sup>②</sup>。

在实践中,每天能进行的检查数量受 MRI 设备检查能力的限制,假设每天最多进行 $M$ 个检查。这里,我们采用每天检查数量的上限,而非每天检查总时长的上限,有以下两点原因:一、在实践中, MRI 检查所耗费的时间一般是固定的,因此控制检查数量,近似于控制检查总时长;二、本文主要考虑转换成本与延迟检查之间的权衡,避免对检查时长进行建模,能让模型更清晰地捕获问题的实质。此外,在数值实验中,我们将问题拓展为给定每天检查

<sup>②</sup> 医院可通过成本函数 $f_c$ 、 $f_w$ 和参数 $c_r$ 来体现对不同成本的重视程度。例如考虑线性函数 $f_c(n) = c_c n$ 和 $f_w(m) = c_w m$ ,则 $c_w < c_c$ , $c_r < c_c$ 表示医院非常重视 MRI 设备使用效率,反之则表示更重视患者就医体验。

总时长的上限,考虑不同类型检查耗费时间不同的情况,能得到相似的结论(见第 5.4 章节)。

表 1 符号及缩写释义表

| 符号/缩写                                | 释义                             |
|--------------------------------------|--------------------------------|
| $T$                                  | 可预约的时间范围                       |
| $J$                                  | 检查类型的数量                        |
| $M$                                  | 医院每日检查能力上限                     |
| $S^t = \{s_{i,j}^t\}_{T \times J}$   | 第 $t$ 期初的排程表                   |
| $D^t = \{d_j^t\}_{1 \times J}$       | 第 $t$ 期的检查需求                   |
| $A^t = \{a_{i,j}^t\}_{T+1 \times J}$ | 对第 $t$ 期需求的排程表                 |
| $f_c(\cdot)$                         | 转换成本函数                         |
| $f_w(\cdot)$                         | 患者等待成本函数                       |
| $c_r$                                | 患者拒绝成本                         |
| $g^t(S^t, D^t, A^t)$                 | 医院在第 $t$ 期的成本                  |
| $NN$                                 | 医院的总经营周期                       |
| $\delta$                             | 折现率                            |
| $\hat{M}$                            | 充分大的正整数                        |
| $\pi = \{\mu^1, \dots, \mu^{NN}\}$   | 医院选择排程表的策略                     |
| $C_\pi^1$                            | 给定策略 $\pi$ , 医院的总成本            |
| SDP                                  | 单日规则 (Same Day Policy)         |
| OAP                                  | 开放获取规则 (Open Access Policy)    |
| MP                                   | 短视规则 (Myopic Policy)           |
| DA                                   | 分解算法 (Decomposition Algorithm) |

给定第 $t$ 期的初始预约表 $S^t$ 、检查需求 $D^t$ 和排程表 $A^t$ ,可以计算医院在第 $t$ 期产生的成本:

$$g^t(S^t, D^t, A^t) = [f_w(1), \dots, f_w(T-1), c_r] \left[ \sum_{j=1}^J a_{2,j}^t, \dots, \sum_{j=1}^J a_{T,j}^t, \sum_{j=1}^J a_{T+1,j}^t \right]^T + f_c \left( \|[s_{1,1}^t + a_{1,1}^t, \dots, s_{1,J}^t + a_{1,J}^t]\|_0 \right) \#(1)$$

公式(1)中,第一部分表示排程表 $A^t$ 产生的患者等待和拒绝成本;第二部分表示在第 $t$ 期产生的转换成本,其中, $0$ -范数 $\|x\|_0$ 的取值为向量中非零元素的数量,即安排在第 $t$ 期的患者类型的数量。

第 $t$ 期结束后,医院根据当前预约表 $S^t$ 和排程表 $A^t$ 更新预约表:删去当期已经完成的预约,保留未被服务的预约,得到第 $t+1$ 期开始前面临的预约表 $S^{t+1}$ 。则预约表的更新过程为:

$$S^{t+1} = \begin{bmatrix} s_{2,1}^t + a_{2,1}^t & \cdots & s_{2,J}^t + a_{2,J}^t \\ \vdots & \ddots & \vdots \\ s_{T-1,1}^t + a_{T-1,1}^t & \cdots & s_{T-1,J}^t + a_{T-1,J}^t \\ 0 & \cdots & 0 \end{bmatrix} \quad (2)$$

假设医院考虑 $NN$ 期的总成本<sup>③</sup>,折现率为 $\delta$ ,给定初始预约表 $S^1$ 、检查需求 $D^1$ 和策略 $\pi = \{\mu^1, \dots, \mu^{NN}\}$ ,其中 $\mu^t$ 表示在第 $t$ 期,对每种可能的预约表 $S^t$ 及检查需求 $D^t$ ,执行怎样的排程 $A^t$ ,即 $A^t = \mu^t(S^t, D^t)$ ,则医院从第1期到第 $NN$ 期的总成本为 $C_\pi^1(S^1, D^1) = g^1(S^1, D^1, A^1) +$

<sup>③</sup> 当 $NN$ 充分大时,可以视为考虑无穷长时间上的总成本,即医院整个经营周期的(折现)总成本。

$E\{\sum_{t=2}^{NN} \delta^{t-1} g(S^t, D^t, A^t)\}$ 。因此，医院面临的优化问题为(P1)：

$$\text{Min}_{A^t} \quad C_{\pi}^1(S^1, D^1) = g^1(S^1, D^1, A^1) + E\left\{\sum_{t=2}^{NN} \delta^{t-1} g(S^t, D^t, A^t)\right\} \quad (3)$$

$$\text{s. t.} \quad a_{i,j}^t \in Z_+, \forall i = 1, \dots, T+1, j = 1, \dots, J, t = 1, \dots, NN \quad (4)$$

$$\sum_{i=1}^{T+1} a_{i,j}^t = d_j^t, \forall j = 1, \dots, J, t = 1, \dots, NN \quad (5)$$

$$\sum_{j=1}^J (a_{i,j}^t + s_{i,j}^t) \leq M, \forall i = 1, \dots, T, t = 1, \dots, NN \quad (6)$$

其中，式(4)为正整数约束，即 MRI 设备的预约排程以每个检查需求为单位；式(5)为检查需求约束，即每个检查需求都需要被安排或拒绝；式(6)为 MRI 设备检查能力约束，即安排在同一期的检查数量之和不超过  $M$ 。

优化问题(P1)求解的难度来源于三个方面。第一，决策变量有时间和检查类型两个维度，而过往的预约调度文献只有时间一个维度；第二，由于安排的检查日期和患者类型都是整数变量，因此这是一个组合优化问题；第三，由于 0-范数的引入，导致目标函数是非线性的，无法将 DP 问题改写为整数规划问题。

## 1.2 最优解求解

针对中小规模问题，可采用策略迭代 (Policy iteration) 算法得到最优解。策略迭代算法可以用来求解包含有限种状态 (State) 和有限种控制 (Action) 的动态规划问题。首先，选择初始策略  $\pi_1$ ，该策略规定了每种状态下执行的控制。其次，计算在策略  $\pi_1$  下每种状态的性能势 (Potential)，即在策略  $\pi_1$  下每种状态的长期成本。最后，进行策略改进：对每种状态，选择使执行控制产生的当前成本与执行控制后新状态的性能势之和最小控制。若改进后的策略与原策略不同，则计算新策略下每种状态的性能势，并继续进行策略改进。曹希仁<sup>[23]</sup>证明了策略迭代算法能收敛到最优策略。(应用策略迭代算法求(P1)最优策略的伪代码见附录 1 中的附表 1。)

## 2 近似算法

### 2.1 单日规则

虽然策略迭代算法能得到最优解，但由于计算复杂度高，无法求解大规模问题。因此，参考医院在 MRI 预约调度上的管理实践，本文提出三种预约调度规则。

实践中，许多医院为了方便管理，对每个检查需求，只能选择当天检查或拒绝，不开放未来日期的预约，我们将该排程规则记为“单日规则” (Same Day Policy)。在单日规则下，第  $t+1$  期初的预约表  $S^{t+1}$  不受第  $t$  期的决策  $A^t$  的影响。因此，动态规划问题(P1)转换为在每期初求解整数规划问题(P2)：

$$\text{Min}_{a_{1,j}^t, b_j, j \in \{1, \dots, J\}} \quad f_r \left( \sum_{j=1}^J d_j^t - \sum_{j=1}^J a_{1,j}^t \right) + f_c \left( \sum_{j=1}^J b_j \right) \quad (7)$$

$$\text{s. t.} \quad (4), (5), (6),$$

$$\bar{M} b_j \geq a_{1,j}^t, \text{ for } j = 1, \dots, J \quad (8)$$

$$b_j \in \{0, 1\}, \text{ for } j = 1, \dots, J \quad (9)$$

其中，由于不开放未来日期的预约，因此每期只决策当期检查量 $a_{1,j}^t$ 以及拒绝量 $a_{T+1,j}^t$ 。由于决策需满足需求约束 $a_{1,j}^t + a_{T+1,j}^t = d_j$ ，因此每期只需决策当期检查量 $a_{1,j}^t$ 。由于0-范数公式计算复杂，我们引入新的决策变量 $b_j$ 来计算当期服务的患者类型数量。式(7)衡量了决策变量 $a_{1,j}^t$ ， $b_j$ 产生的拒绝成本与转换成本。式(4)，(5)，(6)分别为正整数约束、检查需求约束和医院服务能力约束，式(8)、(9)保证了当 $a_{1,j}^t > 0$ 时， $b_j = 1$ ，而最小化目标函数保证了当 $a_{1,j}^t = 0$ 时， $b_j = 0$ 。因此， $\sum_{j=1}^J b_j$ 可准确衡量医院当期服务的患者类型数量，其中 $\hat{M}$ 为充分大的正整数。

尽管可以通过 Gurobi 等求解器得到(P2)的最优解，但在大规模问题中依然需要较长的求解时间。因此我们根据(P2)最优解的性质，设计了求(P2)最优解的多项式时间算法，记为“单日规则算法”（SDP algorithm）。

**命题 1.1** 给定当前预约表 $S^t$ 、检查需求 $D^t$ 和最优排程 $A^t$ ，若 $a_{1,j}^t > 0$ ，则 $a_{1,j}^t = \min\{d_j^t, M - \sum_{k=1}^J s_{1,k}^t - \sum_{k \neq j} a_{1,k}^t\}$ 。（本文所有证明详见附录 2）

由于在单日规则下只允许当天检查或拒绝，医院一旦选择接受 1 个 $j$ 类检查，那么多接受 1 个 $j$ 类检查不会增加转换和等待成本，只会降低拒绝成本。因此，一旦接受一个 $j$ 类检查，医院应尽可能多接受 $j$ 类检查，直到达到需求或容量约束。

**命题 1.2** 给定当前预约表 $S^t$ 、检查需求 $D^t$ 和可行排程 $A^t$ ，若 $A^t$ 满足 $f_r(n_r) - f_r(|n_r - c_1|) < f_c(n_c + 1) - f_c(n_c)$ ，其中 $n_r$ 为排程 $A^t$ 中被拒绝的检查需求数量， $c_1 = M - \sum_{j=1}^J (s_{1,j}^t + a_{1,j}^t)$ 为排程 $A^t$ 中第一天的剩余容量， $n_c$ 表示在排程 $A^t$ 下第 $t$ 期服务的检查类型数量，则增加第 $t$ 期服务的检查类型数量会增加总成本。

若我们从拒绝全部检查需求开始，逐渐增加当期检查的检查类型，则命题 1.2 提供了这个过程的停止条件。 $f_r(n_r) - f_r(|n_r - c_1|)$ 表示可能降低的拒绝成本的最大值， $f_c(n_c + 1) - f_c(n_c)$ 表示增加一个检查类型所产生的转换成本，若降低的拒绝成本小于增加的转换成本，则我们不再增加当期的检查类型。

**命题 1.3** 给定当前预约表 $S^t$ 、检查需求 $D^t$ 和可行排程 $A^t$ ，对 $\forall j_0 \in \{1, \dots, J\}$ 使得 $a_{1,j_0}^t = 0$ ，若满足 $f_r(n_r) - f_r(|n_r - d_{j_0}^t|) < f_c(n_c + 1) - f_c(n_c)$ ，则接受 $j_0$ 类检查会增加总成本。

命题 1.3 提供了选择要增加的检查类型的条件，若 $f_r(n_r) - f_r(|n_r - d_{j_0}^t|) < f_c(n_c + 1) - f_c(n_c)$ ，则接受 $j_0$ 类检查所降低的拒绝成本，无法覆盖其带来的转换成本的增加，因此不应该接受 $j_0$ 类检查。

**命题 1.4** 给定当前预约表 $S^t$ 和检查需求 $D^t$ ，将 $J$ 类检查按照需求数量从大到小排序，即 $d_{j(1)}^t \geq \dots \geq d_{j(J)}^t$ ，则存在最优排程 $A^t$ 满足 $a_{1,j(1)}^t \geq a_{1,j(2)}^t \geq \dots \geq a_{1,j(J)}^t$ 。

命题 1.4 给出了选择检查类型的顺序，即优先接受需求量更大的检查类型。假如某最优排程有 $a_{1,j(i)}^t > a_{1,j(k)}^t, i > k$ ，则 $d_{j(i)}^t \leq d_{j(k)}^t$ ，于是 $d_{j(k)}^t \geq d_{j(i)}^t > a_{1,j(i)}^t > a_{1,j(k)}^t$ ，因此我们可以将 $j(i)$ 类检查的数量减少到 $a_{1,j(k)}^t$ ，再将空出的 $a_{1,j(i)}^t - a_{1,j(k)}^t$ 个容量用来接受 $j(k)$ 类检查，转换后的排程满足 $a_{1,j(1)}^t \geq a_{1,j(2)}^t \geq \dots \geq a_{1,j(J)}^t$ ，且由于 $d_{j(k)}^t > a_{1,j(k)}^t + (a_{1,j(i)}^t - a_{1,j(k)}^t)$ ，因此该转换是可行的。

**定理 1** 单日规则算法能得到(P2)的最优排程，计算复杂度为 $O(J)$ ，其中 $J$ 为检查类型数量。

表 2 单日规则算法伪代码

| Algorithm: Same Day Policy          |   |
|-------------------------------------|---|
| Input: $S^t, D^t, f_c, f_w, c_r, M$ |   |
| Output: $A^t$                       |   |
| 1.                                  | Initialize $A^t$ : $a_{i,j}^t = 0, \forall i = 1, \dots, T, j = 1, \dots, J, a_{T+1,j}^t = d_j^t, \forall j = 1, \dots, J$  |
| 2.                                  | $n_r = \sum_{j=1}^J a_{T+1,j}^t, n_c = 0, c_1 = M$  |
| 3.                                  | <b>while</b> $f_r(n_r) - f_r( n_r - c_1 ) \geq f_c(n_c + 1) - f_c(n_c)$ <b>do</b>   |
| 4.                                  | $j_m \leftarrow \arg \max \{d_1^t, \dots, d_J^t\}$  |
| 5.                                  | <b>if</b> $f_r(n_r) - f_r( n_r - d_{j_m}^t ) \geq f_c(n_c + 1) - f_c(n_c)$ <b>do</b>  |
| 6.                                  | $a_{1,j_m}^t \leftarrow \min \{d_{j_m}^t, c_1\}$  |
| 7.                                  | $c_1 \leftarrow c_1 - a_{1,j_m}^t, d_{j_m}^t \leftarrow d_{j_m}^t - a_{1,j_m}^t, a_{T+1,j}^t \leftarrow a_{T+1,j}^t - a_{1,j_m}^t,$<br>$n_r \leftarrow n_r - a_{1,j_m}^t, n_c \leftarrow n_c + 1$ |
| 8.                                  | <b>else break</b>   |
| 9.                                  | <b>end if</b>   |

## 2.2 开放获取规则

尽管单日规则简单方便，但不开放未来工作日的预约会导致许多检查需求被拒绝，损害患者就医体验。同时，在检查需求波动很大的情况下，也会导致产能的浪费。为了尽可能减少患者被拒绝的情况和产能的浪费，我们提出“开放获取规则”（Open Access Policy），即只有在前一期安排的检查数量达到容量上限时，才开放下一期的预约，当 $T$ 期全部约满时，才允许拒绝患者。由于开放获取规则要求尽可能将患者往前安排，因此考虑未来成本而调整排程的空间很小，因此我们在开放获取规则下忽略未来成本，得到整数规划问题(P3)：

$$\begin{aligned} \min_{\substack{A^t \\ r_i, i \in \{1, \dots, T+1\} \\ b_j, j \in \{1, \dots, J\}}} & [f_w(1), \dots, f_w(T-1), c_r] \left[ \sum_{j=1}^J a_{2,j}^t, \dots, \sum_{j=1}^J a_{T,j}^t, \sum_{j=1}^J a_{T+1,j}^t \right]^T + f_c \left( \sum_{j=1}^J b_j \right) \quad (10) \\ \text{s.t.} & \quad (4), (5), (6), (8), (9), \end{aligned}$$

$$\sum_{j=1}^J (a_{i,j}^t + s_{i,j}^t) \geq M r_{i+1}, \text{ for } i = 1, \dots, T \quad (11)$$

$$\widehat{M} r_i \geq \sum_{j=1}^J a_{i,j}^t, \text{ for } i = 1, \dots, T+1 \quad (12)$$

$$r_i \in \{0, 1\}, \text{ for } i = 1, \dots, T+1 \quad (13)$$

其中，式(10)衡量了排程 $A^t$ 在第 $t$ 期产生的等待成本、拒绝成本和转换成本。式(4)，(5)，(6)分别为正整数约束、检查需求约束和医院服务能力约束，式(8)、(9)保证了 $\sum_{j=1}^J b_j$ 可准确衡量医院当期服务的检查类型数量，其中 $\widehat{M}$ 为充分大的正整数。式(11)、(12)、(13)为开放获取规则约束，式(12)保证当排程表的第 $i$ 行有安排检查时， $r_i = 1$ ；式(11)保证当 $r_{i+1} = 1$ 时，第 $i$ 行安排的检查数量达到容量约束 $M$ 。通过研究，我们得到了(P3)最优解的性质：

**命题 2.1** 给定当前预约表 $S^t$ ，等待成本与拒绝成本只与当期需求总量 $\sum_{j=1}^J d_j^t$ 有关，与

$\sum_{j=1}^J d_j^t$  在  $J$  类检查中的分配无关。

由于开放获取规则要求只有在前一期排满时才开放下一期预约，因此任一可行排程  $A^t$  都满足：

$$\begin{aligned} \sum_{j=1}^J a_{1,j}^t &= \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j=1}^J d_j^t\}, \\ &\dots \\ \sum_{j=1}^J a_{T,j}^t &= \min \{M - \sum_{j=1}^J s_{T,j}^t, \sum_{j=1}^J d_j^t - \sum_{i=1}^{T-1} \sum_{j=1}^J a_{i,j}^t\}, \\ \sum_{j=1}^J a_{T+1,j}^t &= \sum_{j=1}^J d_j^t - \sum_{i=1}^T \sum_{j=1}^J a_{i,j}^t \end{aligned}$$

换言之，对给定的预约表  $S^t$  和需求总量  $\sum_{j=1}^J d_j^t$ ，当期产生的等待成本与拒绝成本是确定的，我们只需要选择使转换成本最小的排程。

**命题 2.2** 给定当前预约表  $S^t$  和检查需求  $D^t$ ，将所有满足  $s_{1,j}^t > 0$  的检查类型记为集合  $JJ_1$ ，则存在最优排程  $A^t$  满足  $\sum_{j \in JJ_1} a_{1,j}^t = \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j \in JJ_1} d_j^t\}$ 。

若在当前预约表  $S^t$  中，已经安排了  $j$  类检查在当期进行检查，则在当期安排更多  $j$  类检查不会增加转换成本，因此医院的最优选择是优先安排在集合  $JJ_1$  中的检查类型，直到达到容量或需求约束。安排满当期检查后，将剩余检查安排到后续工作日不会在当期产生转换成本，根据命题 2.1，所有符合开放获取规则的排程都产生相同的等待成本与拒绝成本。若集合  $JJ_1$  中的检查需求无法填满当期容量，类似命题 1.4，我们将集合  $JJ_0$  中的检查需求按照从大到小的顺序安排在当期，直到达到容量或需求约束。根据命题 2.1 与命题 2.2，我们设计了求 (P3) 最优解的多项式时间算法。

**定理 2** 开放获取规则算法 (OAP algorithm) 能得到 (P3) 的最优解，计算复杂度为  $O(T \cdot J)$ 。

表 3 开放获取规则算法伪代码

| Algorithm: Open Access Policy       |   |
|-------------------------------------|---|
| Input: $S^t, D^t, f_c, f_w, c_r, M$ |   |
| Output: $A^t$                       |   |
| 1.                                  | Initialize $A^t$ : $a_{i,j}^t = 0, \forall i = 1, \dots, T+1, j = 1, \dots, J$                                      |
| 2.                                  | <b>for</b> $j = 1, \dots, J$ <b>do</b>  |
| 3.                                  | <b>if</b> $s_{1,j}^t > 0$ <b>do</b> $j \in JJ_1$  |
| 4.                                  | <b>if</b> $s_{1,j}^t = 0$ <b>do</b> $j \in JJ_0$  |
| 5.                                  | <b>end if</b>   |
| 6.                                  | <b>end for</b>  |
| 7.                                  | <b>for</b> $j \in JJ_1$ <b>do</b>   |
| 8.                                  | $a_{1,j}^t \leftarrow \min \{M - \sum_{j=1}^J (s_{1,j}^t + a_{1,j}^t), d_j^t\}, d_j^t \leftarrow d_j^t - a_{1,j}^t$ |
| 9.                                  | <b>end for</b>  |
| 10.                                 | <b>for</b> $j \in JJ_0$ <b>do</b>   |



```

11.    $j_m \leftarrow \arg \max \{d_j \mid j \in JJ_0\}$ 
12.    $a_{i,j_m}^t \leftarrow \min \{M - \sum_{j=1}^J (s_{i,j}^t + a_{i,j}^t), d_{j_m}^t\}, d_{j_m}^t \leftarrow d_{j_m}^t - a_{i,j_m}^t$ 
13.   end for
14.   for  $i = 1, \dots, T$  do
15.       for  $j = 1, \dots, J$  do
16.            $a_{i,j}^t \leftarrow \min \{M - \sum_{j=1}^J (s_{i,j}^t + a_{i,j}^t), d_j^t\}, d_j^t \leftarrow d_j^t - a_{i,j}^t$ 
17.       end for
18.   end for
19.   for  $j = 1, \dots, J$  do
20.        $a_{T+1,j}^t \leftarrow d_j^t$ 
21.   end for

```

### 2.3 短视规则

尽管开放获取规则能避免医院产能浪费，但会增加医院成本。比如当天剩余 1 个检查名额，但有 10 个  $j$  类检查尚未分配时，按照开放获取规则，医院要在当天进行其中 1 个  $j$  类检查，再将剩余 9 个检查安排在下一期，产生两次转换成本。显然，当患者等待 1 天的成本小于转换成本时，将 10 个检查都安排在下一期更好。为了避免过度增加转换成本，我们考虑短视规则（Myopic Policy），即忽略未来成本，在每期初最小化等待成本、拒绝成本与转换成本之和，将(P1)转换为整数规划问题(P4)：

$$\begin{aligned} \min_{b_j, j \in \{1, \dots, J\}} \quad & [f_w(1), \dots, f_w(T-1), c_r] \left[ \sum_{j=1}^J a_{2,j}^t, \dots, \sum_{j=1}^J a_{T,j}^t, \sum_{j=1}^J a_{T+1,j}^t \right]^T + f_c \left( \sum_{j=1}^J b_j \right) \quad (14) \\ \text{s.t.} \quad & (4), (5), (6), (8), (9) \end{aligned}$$

(P4)与开放获取规则下的问题(P3)基本相同，只是删除了确保“只有在前一期安排的检查数量达到容量上限时，才开放下一期的预约”的约束(11)、(12)、(13)。同样，我们也通过探究(P4)最优解的性质来设计多项式时间的求解算法。

**命题 3.1** 给定当前预约表  $S^t$ 、检查需求  $D^t$  和排程表的第一行  $a_{1,j}^t, j = 1, \dots, J$ ，则给定  $a_{1,j}^t$  时的最优排程  $A^t$  除了第一行外，都符合开放获取规则。

由于安排在第  $2 \sim T+1$  行的检查不在当期产生转换成本，因此最优选择是尽量把检查往前安排。可以预见，短视规则应该在检查需求不超过检查容量太多的情况下使用，否则与开放获取规则的差异不大，但在检查需求不太大时，短视规则可以通过转移部分需求到下一期来降低转换成本。基于命题 3.1，寻找(P4)最优解等价于找到排程表第一行的最优解。

**命题 3.2** 给定当前预约表  $S^t$  和检查需求  $D^t$ ，将所有满足  $s_{1,j}^t > 0$  的检查类型记为集合  $JJ_1$ ，则最优排程  $A^t$  满足  $\sum_{j \in JJ_1} a_{1,j}^t = \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j \in JJ_1} d_j^t\}$ 。

命题 3.2 与命题 2.2 相同，增加已被安排在当期的检查类型不会增加转换成本。在开放获取规则下，若集合  $JJ_1$  中的检查没有填满排程表的第一行，需继续填入其余类型的患者，直达到容量上限。但在短视规则下，需要通过比较接受多一类检查增加的转化成本与降低的等待成本来决定。

**命题 3.3** 给定当前预约表  $S^t$  和检查需求  $D^t$ ，将所有满足  $s_{1,j}^t = 0$  的检查类型记为集合  $JJ_0$ ，将集合  $JJ_0$  中的检查类型按照需求量从大到小排序，即  $d_{j(1)}^t \geq \dots \geq d_{j(U_0)}^t$ ，则存在最优排程  $A^t$

满足  $a_{1,j(1)}^t \geq a_{1,j(2)}^t \geq \dots \geq a_{1,j(J_0)}^t$ 。

命题 3.3 与命题 1.4 类似，说明了应优先接受需求量更大的检查类型。由于  $s_{1,j}^t = 0$ ，因此只有当接受  $j$  类检查降低的等待成本超过增加的转换成本时，我们才选择接受，而检查需求量大的检查类型有更大的机会实现该条件。

**定理 3** 短视规则算法（MP algorithm）能得到  $(P4)$  的最优解，计算复杂度为  $O(T \cdot J)$ 。

**表 4 短视规则算法伪代码**

|                                     |   |
|-------------------------------------|---|
| Algorithm: Myopic Policy            |   |
| Input: $S^t, D^t, f_c, f_w, c_r, M$ |   |
| Output: $A^t$                       |   |
| 1.                                  | Initialize $A^t$ : $a_{i,j}^t = 0, \forall i = 1, \dots, T+1, j = 1, \dots, J$  |
| 2.                                  | <b>for</b> $j = 1, \dots, J$ <b>do</b>  |
| 3.                                  | <b>if</b> $s_{1,j}^t > 0$ <b>do</b> $j \in JJ_1$  |
| 4.                                  | <b>if</b> $s_{1,j}^t = 0$ <b>do</b> $j \in JJ_0$  |
| 5.                                  | <b>end if</b>   |
| 6.                                  | <b>end for</b>  |
| 7.                                  | <b>for</b> $j \in JJ_1$ <b>derive</b> $A^t$ by Open Access Policy Algorithm <b>end for</b>  |
| 8.                                  | <b>for</b> $j \in JJ_0$ <b>fix</b> $a_{1,j}^t = 0$ , <b>derive</b> $A^t$ by Open Access Policy Algorithm from 1 <sup>st</sup> row to $(T+1)$ <sup>th</sup> row <b>end for</b> |
| 9.                                  | <b>for</b> $j \in JJ_0$ <b>do</b>   |
| 10.                                 | $j_m \leftarrow \arg\max \{d_j^t, j \in JJ_0\}$   |
| 11.                                 | $cost_1 \leftarrow g(S^t, D^t, A^t)$ (cost when postpone demand of type $j_m$ )   |
| 12.                                 | $A'^t \leftarrow A^t, a_{1,j_m}^t \leftarrow \min \{M - \sum_{j=1}^J (s_{1,j}^t + a_{1,j}^t), d_{j_m}^t\}$  |
| 13.                                 | given $a_{1,j_m}^t$ , derive $A'^t$ by Open Access Policy Algorithm from 2 <sup>nd</sup> row to $(T+1)$ <sup>th</sup>   |
| 14.                                 | $cost_2 \leftarrow g(S^t, D^t, A'^t)$ (cost when accept type $j_m$ in first day)  |
| 15.                                 | <b>if</b> $cost_2 < cost_1$ <b>do</b> $A^t \leftarrow A'^t$   |
| 16.                                 | <b>end for</b>  |

## 2.4 分解算法

尽管在单日规则、开放获取规则和短视规则下，医院能快速为新抵达的需求安排检查时间，但没有考虑当期排程对未来成本的影响。而考虑未来成本的策略迭代算法在求  $(P1)$  最优解时，控制 (Action) 的维度是时间  $\times$  类型，算法复杂度非常高。我们根据分而治之思想 (Divide and Conquer)，将原问题按照检查类型分解为  $J$  个子问题，每个子问题的控制 (Action) 只有时间维度，算法复杂度大大降低。据此我们设计了忽略容量约束的分解算法 (Decomposition Algorithm)，可在考虑未来成本的情况下近似求解  $(P1)$ 。其每个子问题  $(P5_j)$  为：

$$\min g_j^1(S_j^1, d_j^1, A_j^1) + E\left\{\sum_{t=2}^{NN} \delta^{t-1} g(S_j^t, d_j^t, A_j^t)\right\} \quad (15)$$

$$s. t. \quad a_{i,j}^t \in Z_+, \forall i = 1, \dots, T+1, t = 1, \dots, NN \quad (4')$$

$$\sum_{i=1}^{T+1} a_{i,j}^t = d_j^t, \forall t = 1, \dots, NN \quad (5')$$

$$a_{i,j}^t + s_{i,j}^t \leq M, \forall i = 1, \dots, T, t = 1, \dots, NN \quad (6')$$

其中， $g_j^t(S_j^t, a_j^t, A_j^t) = [f_w(1), \dots, f_w(T-1), c_r][a_{2,j}^t, \dots, a_{T,j}^t, a_{T+1,j}^t]^T + f_c(\|s_{1,j}^t + a_{1,j}^t\|_0)$ 衡量了 $j$ 类检查在第 $t$ 期的等待成本、拒绝成本与转换成本。式(4')，(5')，(6')分别为对 $j$ 类检查的整数约束、需求约束和容量约束。若 $M = +\infty$ 且 $f(n_1) + f(n_2) = f(n_1 + n_2)$ ，那么求解 $J$ 个子问题(P5 <sub>$j$</sub> )等价于求解原问题(P1)。若 $M < +\infty$ ， $J$ 个子问题的最优解不一定能组合为原问题的可行解。在分解算法（Decomposition Algorithm）中，优先接受检查数量大的类型，将检查数量少的检查类型往后推，能在降低转换成本的同时，把 $J$ 个子问题的最优解转换为原问题(P1)的可行近似解。

表 5 分解算法伪代码

| Algorithm: Decomposition Algorithm  |  |
|---|--|
| Input: $S^t = \{s_{i,j}^t\}_{T \times J}$ , $D^t$ , $f_c$ , $f_w$ , $c_r$ , $M$ |  |
| Output: $A^t$   |  |
| 1.  | Solve (P5 <sub><math>j</math></sub> ) by Policy Iteration, get $A_j^t = [a_{1,j}^t, \dots, a_{T+1,j}^t]^T$ |
| 2.  | Initialization: $A^t = [A_1^t, \dots, A_J^t]$  |
| 3.  | <b>for</b> $i = 1, \dots, T$   |
| 4.  | <b>while</b> $\sum_{j=1}^J (s_{i,j}^t + a_{i,j}^t) > M$ <b>do</b>  |
| 5.  | find $k$ s.t. $0 < a_{i,k}^t \leq a_{i,j}^t$ for all $j \in \{1, \dots, J\}$                               |
| 6.  | update $A^t$ : $a_{i,k}^t = a_{i,k}^t - 1$ , $a_{i+1,k}^t = a_{i+1,k}^t + 1$                               |
| 7.  | <b>end while</b>   |
| 8.  | <b>end for</b>   |

### 3 数值实验及案例分析

为检验上述 4 种算法的效果，我们设计了两组数值实验，分别比较了小规模问题中 4 种算法与最优解的性能差距、大规模问题中各算法在不同参数条件下的表现。最后，我们通过医院实际数据分析比较各算法的效果。

#### 3.1 小规模数值实验

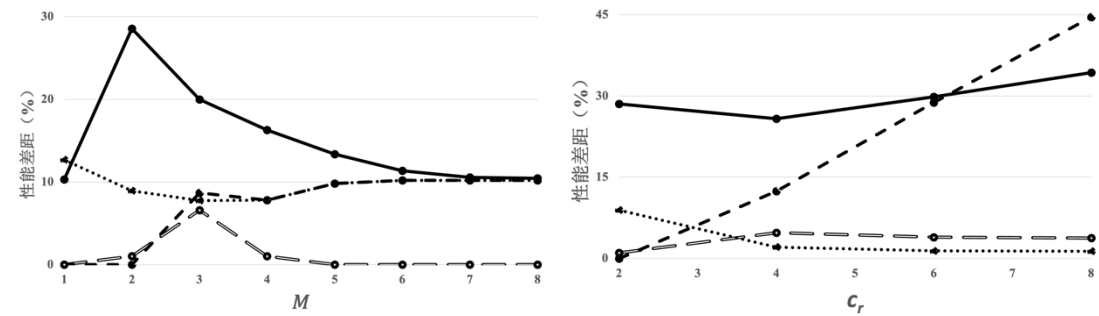
为了控制问题规模，使最优解可以通过策略迭代算法得到，我们设置可预约时间范围  $T = 2$ ，即只能将患者安排在当期、下一期或拒绝；检查类型数量  $J = 2$ ；检查需求服从均值为  $\lambda = 1$  的泊松分布，且限制最大值为 2。在这样的问題规模下，最大需求为 8 人，因此将检查容量按 1 至 8 均匀取值，即  $M \in \{1, \dots, 8\}$ 。至于成本，方便起见我们使用线性函数，即  $f_w(t) = c_w t$ ， $f_c(n) = c_c n$ 。我们将  $c_w$ （一位患者等待一天的成本）的均值标准化为 1；拒绝一位患者的成本应大于让一位患者等待多日，因此把  $c_r$  的均值设为 6；将转换成本  $c_c$  的均值设为 3，使其略大于将一类患者推迟一天产生的等待成本。在实验中， $c_w$ 、 $c_r$ 、 $c_c$  根据其均值按均匀分布取值，即  $c_w \in \{0, 0.5, \dots, 2\}$ ， $c_r \in \{2, 4, \dots, 10\}$ ， $c_c \in \{0, 1, \dots, 6\}$ 。

基于以上参数设置，分别根据策略迭代算法、单日规则算法、开放获取规则算法、短视规则算法和分解算法求解最优策略  $\pi$ ，并使用值迭代（Value iteration）计算各参数组合下的长期平均成本（Long-term average cost），所得结果如表 6 与图 1 所示，其中性能差距为各算

法下长期平均成本超出最优解的百分比，即性能差距=  $\left(\frac{\text{近似算法长期平均成本}}{\text{最优解长期平均成本}} - 1\right) \times 100\%$ 。

表 6 小规模数值实验结果

| 模型参数   |     |           |       |     |       |       | 性能差距 % (运算时间 s) |              |              |              |              |
|--------|-----|-----------|-------|-----|-------|-------|-----------------|--------------|--------------|--------------|--------------|
| $T$    | $J$ | $\lambda$ | $c_w$ | $M$ | $c_c$ | $c_r$ | 最优解             | SDP          | OAP          | MP           | DA           |
| 2      | 2   | 1         | 1     | 1   | 2     | 2     | -(0.07)         | 0.00 (0.02)  | 12.69 (0.04) | 10.30 (0.03) | 0.00 (0.06)  |
| 2      | 2   | 1         | 1     | 2   | 2     | 2     | -(0.22)         | 0.00 (0.02)  | 8.94 (0.05)  | 28.54 (0.05) | 1.03 (0.07)  |
| 2      | 2   | 1         | 1     | 3   | 2     | 2     | -(1.18)         | 8.69 (0.02)  | 7.80 (0.04)  | 19.97 (0.04) | 6.58 (0.07)  |
| 2      | 2   | 1         | 1     | 4   | 2     | 2     | -(1.67)         | 7.80 (0.02)  | 7.80 (0.03)  | 16.30 (0.04) | 1.02 (0.07)  |
| 2      | 2   | 1         | 1     | 5   | 2     | 2     | -(2.35)         | 9.82 (0.02)  | 9.82 (0.06)  | 13.36 (0.04) | 0.00 (0.06)  |
| 2      | 2   | 1         | 1     | 6   | 2     | 2     | -(6.50)         | 10.19 (0.03) | 10.19 (0.04) | 11.35 (0.04) | 0.00 (0.07)  |
| 2      | 2   | 1         | 1     | 7   | 2     | 2     | -(10.60)        | 10.19 (0.02) | 10.19 (0.06) | 10.56 (0.04) | 0.00 (0.08)  |
| 2      | 2   | 1         | 1     | 8   | 2     | 2     | -(17.80)        | 10.19 (0.02) | 10.19 (0.05) | 10.43 (0.04) | 0.00 (0.08)  |
| 2      | 2   | 1         | 1     | 2   | 0     | 4     | -(0.73)         | 58.99 (0.02) | 0.00 (0.03)  | 34.03 (0.03) | 0.00 (0.06)  |
| 2      | 2   | 1         | 1     | 2   | 1     | 4     | -(0.71)         | 22.10 (0.02) | 1.13 (0.03)  | 41.40 (0.03) | 3.18 (0.06)  |
| 2      | 2   | 1         | 1     | 2   | 2     | 4     | -(0.72)         | 12.42 (0.03) | 2.09 (0.03)  | 25.81 (0.04) | 4.74 (0.06)  |
| 2      | 2   | 1         | 1     | 2   | 3     | 4     | -(0.27)         | 8.09 (0.03)  | 2.80 (0.04)  | 15.46 (0.03) | 11.69 (0.07) |
| 2      | 2   | 1         | 1     | 2   | 4     | 4     | -(0.73)         | 9.67 (0.03)  | 7.25 (0.03)  | 13.65 (0.03) | 12.45 (0.07) |
| 2      | 2   | 1         | 1     | 2   | 5     | 4     | -(0.73)         | 2.75 (0.02)  | 11.94 (0.04) | 16.86 (0.03) | 1.13 (0.09)  |
| 2      | 2   | 1         | 1     | 2   | 6     | 4     | -(0.18)         | 0.76 (0.02)  | 19.36 (0.03) | 20.10 (0.03) | 0.14 (0.07)  |
| 2      | 2   | 1         | 1     | 2   | 2     | 2     | -(0.19)         | 0.00 (0.03)  | 8.94 (0.03)  | 28.54 (0.03) | 1.03 (0.06)  |
| 2      | 2   | 1         | 1     | 2   | 2     | 4     | -(0.72)         | 12.42 (0.02) | 2.09 (0.03)  | 25.81 (0.03) | 4.74 (0.06)  |
| 2      | 2   | 1         | 1     | 2   | 2     | 6     | -(0.72)         | 28.78 (0.03) | 1.39 (0.03)  | 29.85 (0.04) | 3.92 (0.08)  |
| 2      | 2   | 1         | 1     | 2   | 2     | 8     | -(0.73)         | 44.55 (0.02) | 1.34 (0.03)  | 34.32 (0.03) | 3.77 (0.08)  |
| 2      | 2   | 1         | 1     | 2   | 2     | 10    | -(0.70)         | 59.16 (0.03) | 1.29 (0.03)  | 38.46 (0.03) | 3.63 (0.09)  |
| 2      | 2   | 1         | 0     | 2   | 4     | 4     | -(0.40)         | 37.10 (0.02) | 18.81 (0.03) | 11.89 (0.03) | 14.05 (0.07) |
| 2      | 2   | 1         | 0.5   | 2   | 4     | 4     | -(0.26)         | 21.46 (0.03) | 12.02 (0.03) | 10.75 (0.03) | 18.70 (0.07) |
| 2      | 2   | 1         | 1     | 2   | 4     | 4     | -(0.71)         | 9.67 (0.03)  | 7.25 (0.03)  | 13.65 (0.03) | 20.82 (0.06) |
| 2      | 2   | 1         | 1.5   | 2   | 4     | 4     | -(0.31)         | 0.63 (0.03)  | 4.03 (0.03)  | 16.83 (0.04) | 53.40 (0.06) |
| 2      | 2   | 1         | 2     | 2   | 4     | 4     | -(0.20)         | 0.00 (0.02)  | 8.94 (0.03)  | 28.54 (0.03) | 56.10 (0.08) |
| 平均性能差距 |     |           |       |     |       |       | -               | 15.42        | 7.53         | 21.07        | 8.89         |
| 平均运算时间 |     |           |       |     |       |       | 1.98            | 0.02         | 0.04         | 0.03         | 0.07         |



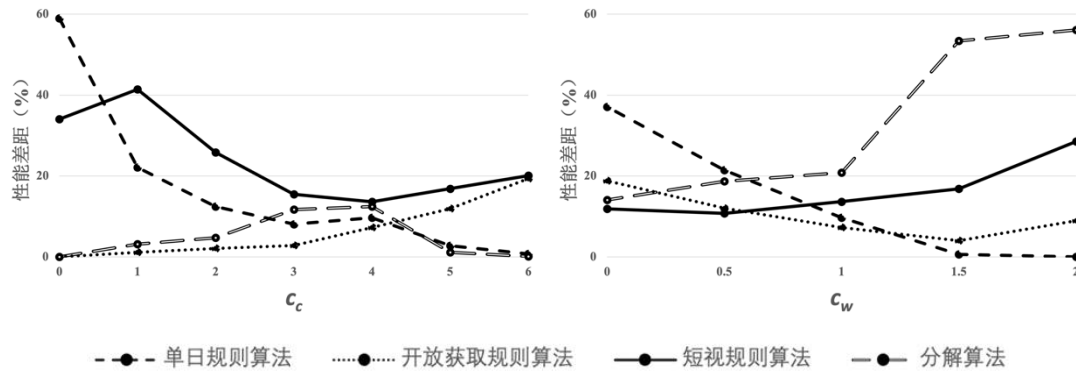


图 1 小规模数值实验结果

从表 6 可以看出，每种近似算法的计算时间都远低于策略迭代算法。随着 MRI 检查单日容量 $M$ 的增加，策略迭代算法的运算时间从 0.07 秒快速上涨到 17.8 秒，而其他近似算法的运算时间始终不超过 0.1 秒。换言之，与策略迭代算法相比，本文提出的近似算法不存在维数灾难问题。从表 6 和图 1 可以看出，每个近似算法适用的情形不同，且在每种参数组合下，都存在至少一种表现优异的算法。

在医院检查能力紧张时，单日规则算法与分解算法表现最佳，长期平均成本达到了最优。这是因为在医院检查能力紧张时，几乎每天新抵达的检查需求都超出检查能力限制，因此最佳选择是不开放未来预约，在每期达到容量限制后拒绝剩余需求，这与单日规则一致。同时，在检查需求大大超出检查能力时，通常存在只需挑选一个类型的检查，其需求就能填满当天的检查容量，因此将原问题按照检查类型分解为 $J$ 个子问题不影响最优解。

随着医院检查能力的提高，会出现需求时而大于容量、时而小于容量的情况，因此最佳选择是将当期抵达的需求安排到未来工作日，以抚平需求波动，这与单日规则存在较大差异，因此其性能差距增大。同样，在医院检查能力提高时，分解算法由于没有考虑不同类型检查间的相互影响，导致其性能差距增大，但其表现依然优于其他算法，性能差距小于 6.6%。当医院检查能力充分大时，排程不受检查能力的限制，且由于数值实验考虑的是等待成本与转换成本线性增加，因此分解算法能得到最优排程。

在拒绝成本较低时，最优排程倾向于拒绝患者而非让其等待并增加下一期的转换成本，这与单日规则相似，因此此时单日规则的表现最好。当拒绝成本变大时，最优排程倾向于不拒绝患者，这与开放获取规则类似，因此开放获取规则的表现最好。

在 MRI 设备转换成本低时，最优排程更有可能通过增加检查类型来降低患者等待和拒绝成本，这与开放获取规则类似，因此其表现最佳。在转换成本高时，只有在某类检查数量较多时，才为该检查花费转换成本。在单日规则中，我们只有在接受某类检查所降低的拒绝成本超过其增加的转换成本时，才接受该类检查，而降低的拒绝成本随检查数量递增，因此单日规则在此情形下表现很好。在分解算法中，求解每个子问题时，都权衡了该类检查带来的拒绝成本和转换成本，因此分解算法在此情形下表现也很好。

在等待成本较低时，由于短视规则受等待成本的影响较小，并且比起单日规则能通过让患者等待来降低拒绝成本、比起开放获取规则能考虑到转换成本的影响、比起分解算法能考虑到不同类型检查间的相互作用，因此性能表现最好。在等待成本适中时，由于等待成本增加，短视规则比起前一种情形更难通过让患者等待来降低拒绝和转换成本；而由于等待成本不够高，因此让患者等待而非直接拒绝的开放获取规则表现最佳。在等待成本较高时，最优

排程倾向于不让患者等待，与单日规则较为相似，因此其表现最佳。

通过小规模问题的数值实验可以看出，在不同参数情况下，都能找到效果较好的近似解，因此医院可以根据 MRI 检查能力的紧张与否，拒绝成本、等待成本及转换成本的高低来选择合适的算法为患者预约检查日期。尽管单日规则算法和短视规则算法在某些情况表现极佳，但从平均性能差距来看，它们的总体表现劣于开放获取规则算法和分解算法，因此医院在采用单日规则和短视规则时需要仔细辨别适用的场景。

### 3.2 大规模数值实验

通过小规模问题证明了近似算法的有效性后，本文对更方便应用的单日规则、开放获取规则和短视规则进行规模较大的数值实验，以探究 3 种规则在大规模情形下的表现。

大规模数值实验的参数设置如下：一般医院可预约时间范围为一周，即 $T = 7$ ；在实践中，一般会用到 7 种 MRI 线圈，即 $J = 7$ ；上海某三甲医院日均 MRI 检查需求约为 100 左右，故假设每类检查需求服从均值 $\lambda = 15$ 的泊松分布；每日检查能力设为 $M \in \{85, 105, 125\}$ ，分别对应检查能力紧张、均衡和充分的情况；患者单位等待成本的均值标准化为 1，因此设置 $c_w \in \{0.5, 1, 1.5\}$ ；将拒绝成本设为等待成本的 10 倍（略大于将患者安排在最后一天的等待成本），因此设置 $c_r \in \{5, 10, 15\}$ ；转换成本应略大于将某类需求全部推迟一天的等待成本，故设 $c_c = \{10, 20, 30\}$ 。基于以上参数，使用 Matlab 进行仿真求解，部分结果如图 2 所示（详细结果见附录中的附表 2）。

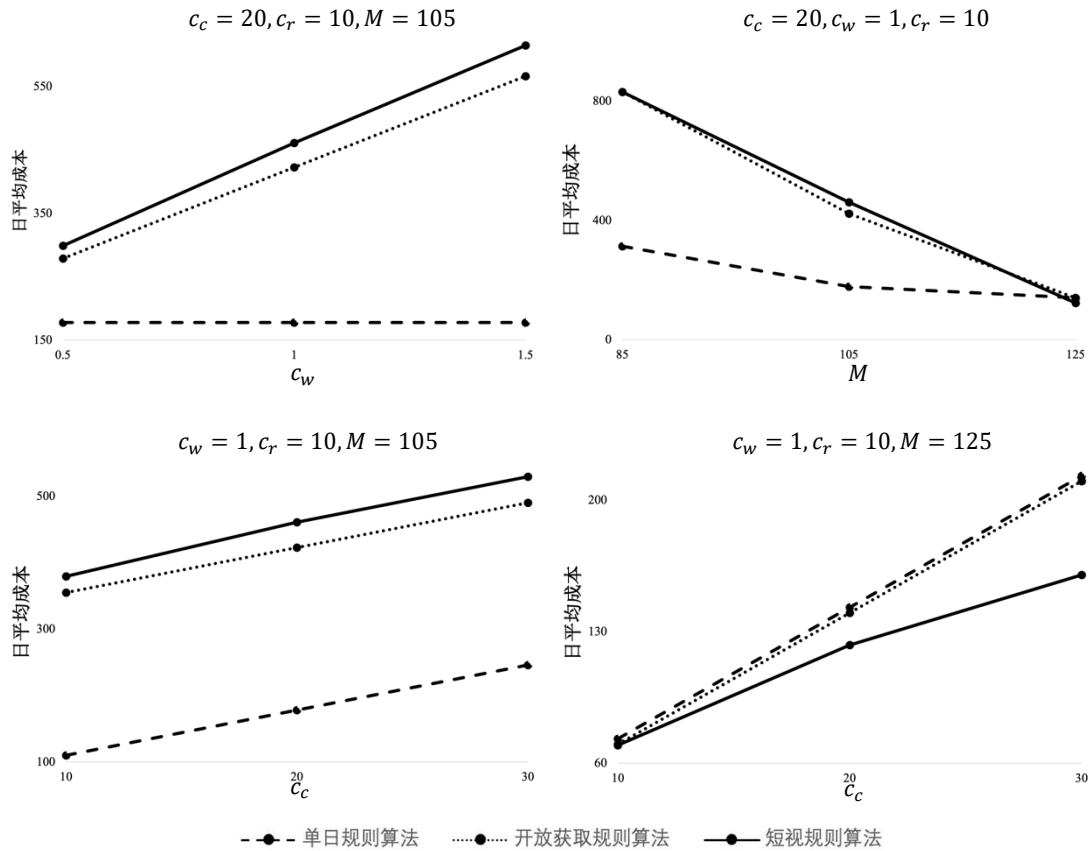


图 2 大规模数值实验结果

首先，单日规则的效果几乎不受患者等待成本的影响，但开放获取规则和短视规则下的

日平均成本会随等待成本的增加而显著增加；并且，等待成本越大，三种规则的表现效果差异越大。此外，随着医院检查能力的增加，所有近似算法的日平均成本都会下降；在医院检查能力紧张和均衡时，单日规则表现最佳；在医院检查能力充足时，短视规则表现更好，但三者差异不大。最后，随着转换成本的增加，三种规则的性能差异越发明显，这说明转换成本对系统性能的影响非常显著，在转换成本较大时，即使医院检查能力充足，三种规则的性能差异也十分明显。总的来说，在患者等待成本越大、医院检查能力紧张、转换成本较大时，医院需要仔细选择所使用的预约规则。

### 3.3 医院实际案例分析

为了验证现实中近似算法的效果，我们搜集并整理了上海某大型综合医院 198 天共 21585 次 MRI 检查的申请日期、检查日期及检查类型的数据，对单日规则算法、开放获取规则算法和短视规则算法的效果进行评价。

医院案例数据中，患者检查需求可按使用线圈的不同分为 7 种类型，而日均检查类型为 5.82 种，检查类型最少时为 1 种，可以看出医院在有意控制每天检查类型的数量，但效果不佳。从每日产生的检查需求来看，均数为 108.43 次，方差为 3119；从每日实际检查数量看，均值为 100.36 次，最大值为 241 次，方差为 3310，这说明 MRI 的预约调度并没有起到平滑需求波动的效果，每天的检查工作量极不平衡，同时存在着设备的闲置和加班。从患者的就医体验看，平均每个患者等待 2.40 天，最长等待时间为 27 天，并且还有 7.47% 的检查需求被拒绝，就医体验不佳。

在案例分析中，将可预约的时间范围设为一周，即  $T = 7$ ；根据 MRI 线圈类型，将  $J$  设为 7；将实际数据中每日检查数量的中位数设为检查容量，即  $M = 118$ ；将每位患者等待一天的成本标准化为 1，即  $c_w = 1$ ；拒绝一名患者的成本应稍大于将患者安排到最后一天，故设  $c_r = 10$ ；转换成本应稍大于将某类需求全部推后一天产生的等待成本，故将  $c_c$  设为 20。根据单日规则算法、开放获取规则算法和短视规则算法，对实际历史数据中的检查需求求解预约调度，并计算各个算法下拒绝率、平均等待天数、日平均检查类型数、日平均检查量、日检查量的方差等指标，并与医院实际情况对比。

开放获取规则和短视规则在每项指标上都优于医院实际情况，在降低医院成本的同时提高了患者就医体验；而单日规则通过减少日平均患者类型数降低了医院成本，但提高了患者拒绝率，可能会损害患者的就医体验，因此更适合在检查需求远超出检查能力的医院中使用。

表 7 实际案例分析结果

| 指标      | SDP    | OAP    | MP     | 医院实际情况 |
|---------|--------|--------|--------|--------|
| 拒绝率 (%) | 16.78  | 0.00   | 0.00   | 7.47   |
| 平均等待天数  | 0.00   | 1.74   | 1.85   | 2.4    |
| 日平均类型数  | 3.82   | 5.43   | 5.22   | 5.82   |
| 日平均检查量  | 90.71  | 109    | 108.85 | 100.36 |
| 日检查量方差  | 1568.7 | 700.13 | 692.05 | 3310   |

### 3.4 拓展模型数值实验

前文使用“每天最多检查的患者数量”来衡量医院 MRI 设备的检查能力，但实践中可能出现不同类型检查耗费的时间不同的情况，从而相同的设备可能每天能进行 10 个 I 类检查，但只能进行 5 个 II 类检查。因此，我们通过简单的数值实验来说明使用“每天最多检查的患者数量”或“MRI 设备总运行时间”来衡量医院检查能力并不会显著影响本文的主要结

论。

参考小规模数值实验中的参数设置,进一步区分两类检查耗费的时间:假设 II 类检查的耗费时间为 1 个单位时间,I 类与 II 类检查的耗费时间之比为分别为 $\{0.7, 0.8, \dots, 2\}$ 。最后,我们得到图 3 所示的结果。从图 3 发现,3 种算法的性能差距都在检查时间之比为 1 时发生跳转,在其他情况保持稳定。这是由于我们设置了检查总时间 $M = 2$ ,即每天 MRI 设备最多运转 2 个单位时间,在检查时间之比小于 1 时,相当于检查容量 $M = 2$ ;在检查时间之比大于 1 时,相当于检查容量 $M = 1$ ,因此性能差距发生改变。可见,本文所提出的近似算法的表现与检查容量有关,而与检查时间之比几乎无关。

## 4 结语

本文考虑了医院核磁共振检查(MRI)在扫描患者不同部位时产生的转换成本,以及未能及时满足患者就医需求而产生的等候及拒绝成本,对 MRI 设备的预约调度问题建立了马尔可夫决策过程模型,以最小化医院长期总成本,并通过策略迭代算法得到最优排程。为了降低计算复杂度,本文考虑了单日规则、开放获取规则和短视规则,但由于转换成本的引入,即使在给定调度规则下,该问题仍是一个组合优化问题。因此,本文分析了各调度规则下的问题性质,设计并证明了求各规则下最优解的多项式时间算法。并且单日规则、开放获取规则和短视规则下的最优解算法的计算时间与患者总量无关,只随患者类型数量 $J$ 和预约时间范围 $T$ 线性增长,大大降低了计算复杂度。本文还依据分而治之的思想设计了分解算法,得到原问题的近似解。从数值模拟结果来看,本文提出方法均能有效近似最优解。从上海某大型综合医院的实际案例看,本文设计的近似算法在检查容量配置、患者拒绝率、平均等待天数、日平均检查类型数等指标上取得显著改善。

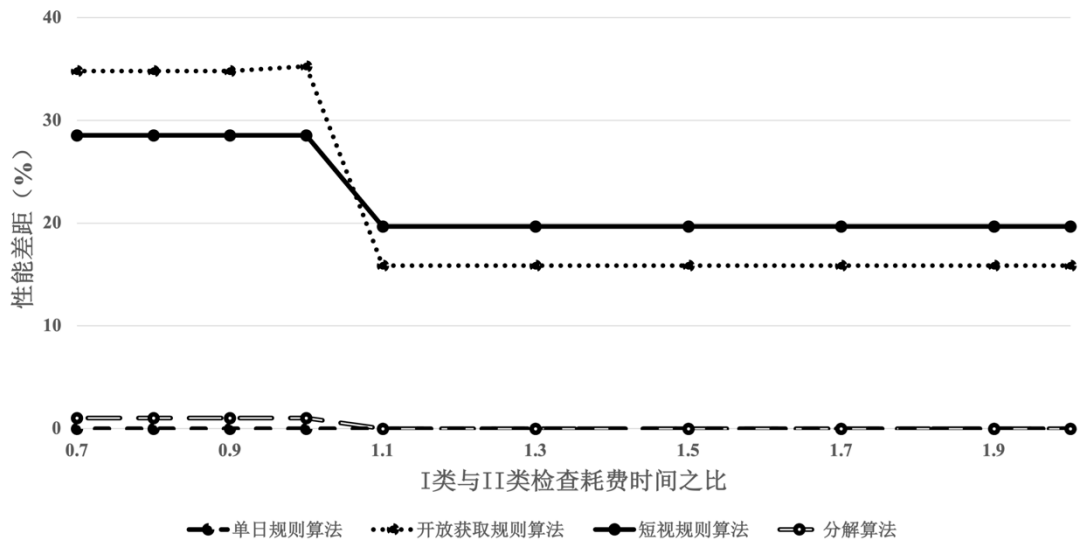


图 3 拓展模型数值实验结果

本文提出的算法对医院 MRI 设备预约调度有实践指导意义。一方面,我们提出的多项式时间算法计算复杂度低,能快速得到对应调度规则下的最优排程;另一方面,我们只要求成本函数递增,没有对设备转换成本和等待成本之间的关系做出假设,因此医院可以根据实际情况灵活选择适合的调度规则及算法。当医院的 MRI 设备检查能力紧张时,应采用单日规则,因为无法通过开放后续工作日预约来抚平需求波动,反而会增加转换成本与患者等待



成本。当医院不希望拒绝患者的检查需求或 MRI 设备转换成本低时,可以采用开放获取规则,充分利用 MRI 设备的检查能力。当医院的 MRI 设备检查能力充足,并且患者检查类型多、可预约时间范围长时,应采用短视规则,能通过将部分检查需求移动到后续工作日来降低 MRI 设备的转换成本,从而降低了总成本。最后,当 MRI 设备的检查能力非常紧张或非常充足时,医院可以忽略检查能力约束,采用分而治之的办法,对每个类型单独通过策略迭代算法进行排程,再组合为可行的排程表,此时与最优解仅有不超过 1.5%的性能差距。从直觉上看, MRI 设备的检查能力越紧张,容量约束的影响越大。但在 MRI 设备的预约排程问题中,为了降低患者等待和拒绝成本,医院应尽量将患者排在靠前的日期,而为了降低转换成本,医院在每期应尽量先处理完一个类型的检查、再考虑下一个类型的检查,这与分解算法的思想是类似的,因此可以很好地近似最优解。当 MRI 设备的检查能力非常充足时,容量约束冗余,因此分解算法也表现良好。

虽然本文研究的是 MRI 设备的预约排程问题,但提出的模型、算法和管理内涵同样能应用在存在转换成本的其他预约排程问题中。例如,在场地出租企业中,若前后预约场地的顾客都用于开会,则无需重新布置场地;若前一位顾客预约场地用于开会,后一位顾客预约场地用于演出,则产生了场地布置成本。

本文的研究有许多可拓展方向。首先,本文考虑 MRI 设备的检查能力存在硬性约束,规定了每天患者检查数量的上限,未来研究可以考虑软检查能力约束,允许超出 MRI 设备检查能力,但产生超时工作成本。另外,本文假设不同检查需求的患者的等待成本函数一致,但在现实中,特定类型的患者的就医需求可能更为迫切。本文的模型、排程规则及算法可以作为未来更多相关问题的研究工具。

#### 参考文献:

- [1] O. ALROWAILI M, AHMED A E, AREABI H A. Factors associated with No-Shows and rescheduling MRI appointments [J]. BMC Health Services Research, 2016, 16(1): 1-7.
- [2] VAN SAMBEEK J R, JOUSTRA P E, DAS S F, *et al.* Reducing MRI access times by tackling the appointment-scheduling strategy [J]. BMJ quality & safety, 2011, 20(12): 1075-1080.
- [3] QIU H, WANG D, WANG Y, *et al.* MRI appointment scheduling with uncertain examination time [J]. Journal of Combinatorial Optimization, 2019, 37(1): 62-82.
- [4] 尹余平, 张春峰. 影响核磁共振检查效率的原因和对策[J]. 医药前沿, 2013, 000(034): 359-359.
- [5] SARTORETTI E, SARTORETTI T, BINKERT C, *et al.* Reduction of procedure times in routine clinical practice with Compressed SENSE magnetic resonance imaging technique[J]. PLoS One, 2019, 14(4): e0214887.
- [6] WALD L L, MCDANIEL P C, WITZEL T, *et al.* Low-cost and portable MRI[J]. Journal of Magnetic Resonance Imaging, 2020, 52(3): 686-696.
- [7] 王利艳, 马玉霞. GE HDx1.5T 磁共振系统线圈故障维修 3 例[J]. 医疗卫生装备, 2015, 36(3):2.
- [8] PANG B, XIE X, JU F, *et al.* A dynamic sequential decision-making model on MRI real-time scheduling with simulation-based optimization [J]. Health Care Management Science, 2022: 1-15.
- [9] 潘兴薇, 耿娜. 面向核磁共振检查的预约调度仿真优化研究[J]. 工业工程与管理, 2016, 21(6): 17-23.
- [10] LIN J, MUTHURAMAN K, LAWLEY M. Optimal and approximate algorithms for sequential clinical scheduling with no-shows [J]. IIE Transactions on Healthcare Systems Engineering, 2011, 1(1): 20-36.

- [11] JIANG R, SHEN S, ZHANG Y. Integer programming approaches for appointment scheduling with random no-shows and service durations [J]. *Operations Research*, 2017, 65(6): 1638-1656.
- [12] LIU N, ZIYA S, KULKARNI V G. Dynamic scheduling of outpatient appointments under patient no-shows and cancellations [J]. *Manufacturing & Service Operations Management*, 2010, 12(2): 347-364.
- [13] FELDMAN J, LIU N, TOPALOGLU H, *et al.* Appointment scheduling under patient preference and no-show behavior [J]. *Operations Research*, 2014, 62(4): 794-811.
- [14] ZACHARIAS C, PINEDO M. Appointment scheduling with no-shows and overbooking [J]. *Production and Operations Management*, 2014, 23(5): 788-801.
- [15] JIANG B, TANG J, YAN C. A stochastic programming model for outpatient appointment scheduling considering unpunctuality [J]. *Omega*, 2019, 82: 70-82.
- [16] WANG S, LIU N, WAN G. Managing appointment-based services in the presence of walk-in customers [J]. *Management Science*, 2020, 66(2): 667-86.
- [17] 王建军, 缪鸿儒, 李播, 等. 随机急诊需求下基于分支定价算法的手术计划研究 [J]. *系统工程理论与实践*, 2018, 38(9): 2348-2362.
- [18] ZHOU S, LI D, YIN Y. Coordinated appointment scheduling with multiple providers and patient-and-physician matching cost in specialty care [J]. *Omega*, 2021, 101: 102285.
- [19] GENG N, XIE X, AUGUSTO V, *et al.* A Monte Carlo optimization and dynamic programming approach for managing MRI examinations of stroke patients [J]. *IEEE Transactions on Automatic Control*, 2011, 56(11): 2515-2529.
- [20] GENG N, XIE X, JIANG Z. Implementation strategies of a contract-based MRI examination reservation process for stroke patients [J]. *European Journal of Operational Research*, 2013, 231(2): 371-380.
- [21] BERTSEKAS D P. Separable Dynamic Programming and Approximate Decomposition Methods [J]. *IEEE Transactions on Automatic Control*, 2007, 52(5): 911-916.
- [22] DAI J G, SHI P. Inpatient overflow: An approximate dynamic programming approach [J]. *Manufacturing & Service Operations Management*, 2019, 21(4): 894-911.
- [23] 曹希仁. 随机学习与优化:基于灵敏度的方法:a sensitivity-based approach [M]. 北京: 清华大学出版署, 2011.
- [24] GENG N, XIE X. Managing Advance Admission Requests for Obstetric Care [J]. *INFORMS Journal on Computing*, 2022, 34(2): 1224-1239.

## 在线附录

### 附录 1 策略迭代算法

假设每个工作日的患者检查需求 $D^t$ 相互独立，且所有可能的检查需求 $D^t$ 包含于有限集合 $DD$ ，即 $D^t \in DD$ 。假设第 $t$ 期检查需求为 $D^t$ 的概率为 $P(D^t)$ ，且 $\sum_{D^t \in DD} P(D^t) = 1$ 。由于预约表 $S^t$ 需满足正整数约束与医院服务能力约束，因此存在有限集合 $SS$ 包含所有可能的预约表 $S^t$ 。同理，由于排程 $A^t$ 需满足正整数约束、检查需求约束和医院服务能力约束，因此给定 $S^t \in SS$ 、 $D^t \in DD$ ，存在有限集合 $AA(S^t, D^t)$ 包含所有满足约束(4)、(5)、(6)的排程 $A^t$ 。通过表 1 所示的伪代码，我们能求解优化问题(P1)的最优策略。

附表 1 策略迭代算法伪代码

| Algorithm: Policy Iteration                 |  |
|---|--|
| Input: $DD, SS, AA(S, D), f_c, f_w, c_r, M$ |  |
| Output: $\pi^*$                             |  |
| Process                                     |  |
| 1.  | $n = 100$ (maximum iteration time)   |
| 2.  | $k =  SS $ (Number of matrixes in the set $SS$ )   |
| 3.  | $I = I_k$ (unit matrix of size $k$ )   |
| 4.  | $e = (1, \dots, 1)^T \in \mathbb{R}^{k \times 1}$  |
| 5.  | Initialize $\pi_1$ : randomly choose $A_1(S, D) \in AA(S, D), \forall S \in SS, D \in DD$  |
| 6.  | <b>for</b> $i = 1, \dots, n$ <b>do</b>   |
| 7.  | Calculate one-step transition probability matrix $T$   |
| 8.  | Calculate one-period cost matrix $R$ by $P(D)$ and $g(S, D, A)$  |
| 9.  | Calculate steady probability $\omega$  |
| 10.   | Calculate potential under $\pi_i$ : $G \leftarrow (I - T + e\omega)^{-1}R$   |
| 11.   | <b>for</b> $S \in SS, D \in DD$ <b>do</b>  |
| 12.   | Update $\pi_{i+1}(S, D)$ : $A_{i+1}(S, D) \leftarrow \underset{A \in AA(S, D)}{\operatorname{argmin}} g(S, D, A) + \sum_{S' \in SS} T'_{S, S'} G_{S'}$ |
|   | where $T'$ is one-step transition probability matrix under action $A \in A(S, D)$ , $G_{S'}$ is potential of $S'$ under $\pi_i$                        |
| 13.   | <b>end for</b>   |
| 14.   | <b>if</b> $\pi_{i+1} = \pi_i$ <b>break for</b>   |
| 15.   | <b>end if</b>  |
| 16.   | <b>end for</b>   |

附录 2 大规模数值实验详细结果

附表 2 大规模数值实验结果

| 模型参数 |     |           |     |       |       |       | 日平均成本  |         |          |
|------|-----|-----------|-----|-------|-------|-------|--------|---------|----------|
| $T$  | $J$ | $\lambda$ | $M$ | $c_w$ | $c_c$ | $c_r$ | SDP    | OAP     | MP       |
| 7    | 7   | 15        | 85  | 1     | 20    | 10    | 313.93 | 830.86  | 831.36   |
| 7    | 7   | 15        | 105 | 1     | 20    | 10    | 178.03 | 422.60  | 460.93   |
| 7    | 7   | 15        | 125 | 1     | 20    | 10    | 141.23 | 140.10  | 122.95   |
| 7    | 7   | 15        | 85  | 1     | 20    | 15    | 414.72 | 930.44  | 931.02   |
| 7    | 7   | 15        | 105 | 1     | 20    | 15    | 198.95 | 422.65  | 461.03   |
| 7    | 7   | 15        | 125 | 1     | 20    | 15    | 141.86 | 140.10  | 122.95   |
| 7    | 7   | 15        | 85  | 0.5   | 20    | 10    | 313.93 | 576.45  | 576.81   |
| 7    | 7   | 15        | 85  | 1     | 20    | 10    | 313.93 | 830.86  | 831.36   |
| 7    | 7   | 15        | 85  | 1.5   | 20    | 10    | 313.93 | 1085.60 | 1085.60  |
| 7    | 7   | 15        | 105 | 0.5   | 20    | 10    | 178.03 | 278.88  | 298.98   |
| 7    | 7   | 15        | 105 | 1     | 20    | 10    | 178.03 | 422.60  | 460.93   |
| 7    | 7   | 15        | 105 | 1.5   | 20    | 10    | 178.03 | 566.33  | 614.77   |
| 7    | 7   | 15        | 125 | 0.5   | 20    | 10    | 141.23 | 140.04  | 98.10    |
| 7    | 7   | 15        | 125 | 1     | 20    | 10    | 141.23 | 140.10  | 122.95   |
| 7    | 7   | 15        | 125 | 1.5   | 20    | 10    | 141.23 | 140.17  | 135.37   |
| 7    | 7   | 15        | 85  | 1     | 10    | 10    | 257.76 | 769.42  | 769.49   |
| 7    | 7   | 15        | 85  | 1     | 20    | 10    | 313.93 | 830.86  | 831.36   |
| 7    | 7   | 15        | 85  | 1     | 30    | 10    | 369.32 | 892.29  | 892.85   |
| 7    | 7   | 15        | 105 | 1     | 10    | 10    | 109.93 | 355.07  | 379.55   |
| 7    | 7   | 15        | 105 | 1     | 20    | 10    | 178.03 | 422.60  | 460.93   |
| 7    | 7   | 15        | 105 | 1     | 30    | 10    | 245.81 | 490.14  | 529.33   |
| 7    | 7   | 15        | 125 | 1     | 10    | 10    | 71.24  | 70.12   | 69.62    |
| 7    | 7   | 15        | 125 | 1     | 20    | 10    | 141.23 | 140.10  | 122.95   |
| 7    | 7   | 15        | 125 | 1     | 30    | 10    | 211.20 | 210.09  | 160.29   |
| 7    | 7   | 15        | 85  | 1     | 20    | 5     | 211.98 | 731.27  | 731.70   |
| 7    | 7   | 15        | 85  | 1     | 20    | 10    | 313.93 | 830.86  | 831.36   |
| 7    | 7   | 15        | 85  | 1     | 20    | 15    | 414.72 | 930.44  | 931.02   |
| 7    | 7   | 15        | 105 | 1     | 20    | 5     | 156.61 | 422.56  | 460.83   |
| 7    | 7   | 15        | 105 | 1     | 20    | 10    | 178.03 | 422.60  | 460.9286 |
| 7    | 7   | 15        | 105 | 1     | 20    | 15    | 198.95 | 422.65  | 461.0276 |
| 7    | 7   | 15        | 125 | 1     | 20    | 5     | 140.58 | 140.10  | 122.95   |
| 7    | 7   | 15        | 125 | 1     | 20    | 10    | 141.23 | 140.10  | 122.95   |
| 7    | 7   | 15        | 125 | 1     | 20    | 15    | 141.86 | 140.10  | 122.95   |

## 附录 3 证明

### 命题 1.1 证明:

假设是  $A^t = \{a_{i,j}^t\}$  是预约表  $S^t$  和检查需求  $D^t$  下的最优排程, 存在  $j$  使得  $a_{1,j}^t \neq \min\{d_j^t, M - \sum_{k=1}^J s_{1,k}^t - \sum_{k \neq j} a_{1,k}^t\}$ 。

$$1) \quad a_{1,j}^t > \min\{d_j^t, M - \sum_{k=1}^J s_{1,k}^t - \sum_{k \neq j} a_{1,k}^t\}$$

若  $a_{1,j}^t > d_j^t$ , 则安排的患者数量超出患者需求,  $A^t$  不可行; 若  $a_{1,j}^t > M - \sum_{k=1}^J s_{1,k}^t - \sum_{k \neq j} a_{1,k}^t$ , 则  $\sum_{k=1}^J (s_{1,k}^t + a_{1,k}^t) > M$ , 当天安排的患者数量超出检查能力限制,  $A^t$  不可行。

$$2) \quad a_{1,j}^t < \min\{d_j^t, M - \sum_{k=1}^J s_{1,k}^t - \sum_{k \neq j} a_{1,k}^t\}$$

构造  $A'^t = A^t, a_{1,j}'^t = a_{1,j}^t + 1, a_{T+1,j}'^t = a_{T+1,j}^t - 1$ , 由于  $a_{1,j}'^t \leq \min\{d_j^t, M - \sum_{k=1}^J s_{1,k}^t - \sum_{k \neq j} a_{1,k}^t\}$ , 因此  $A'^t$  是可行排程, 且  $g^t(S^t, D^t, A^t) - g^t(S^t, D^t, A'^t) = c_r > 0$ , 因此  $A'^t$  是比  $A^t$  成本更低的可行排程, 与  $A^t$  是最优排程矛盾。

综上,  $a_{1,j}^t = \min\{d_j^t, M - \sum_{k=1}^J s_{1,k}^t - \sum_{k \neq j} a_{1,k}^t\}$ 。

### 命题 1.2 证明:

假设是  $A^t = \{a_{i,j}^t\}$  是预约表  $S^t$  和检查需求  $D^t$  下的可行排程, 当天的剩余容量  $c_1 = M - \sum_{j=1}^J (s_{1,j}^t + a_{1,j}^t) > 0$ , 当天的检查类型数量  $n_c = f_c(n_c) < J$ , 其中  $n_c = \|[s_{1,1}^t + a_{1,1}^t, \dots, s_{1,J}^t + a_{1,J}^t]\|_0$  拒绝量  $n_r = \sum_{j=1}^J (d_j^t - a_{1,j}^t)$ 。

由于  $n_c < J$ , 存在类型  $j_0$  使得  $a_{1,j_0}^t = 0$ 。若构造  $A'^t = A^t, a_{1,j_0}'^t = \min\{d_{j_0}^t, M - \sum_{k=1}^J s_{1,k}^t - \sum_{k \neq j_0} a_{1,k}^t\}$ 、 $a_{T+1,j_0}'^t = d_{j_0}^t - a_{1,j_0}^t$ , 则  $n_c' = n_c + 1, n_r - n_r' \leq c_1$ , 从而  $g^t(S^t, D^t, A^t) - g^t(S^t, D^t, A'^t) = f_r(n_r) - f_r(n_r') + f_c(n_c) - f_c(n_c + 1)$ 。

由于  $n_r' \geq n_r - c_1, n_r' \geq 0$  且  $f_r$  是递增函数, 则  $f_r(n_r') \geq f_r(n_r - c_1)$ , 因此:  $g^t(S^t, D^t, A^t) - g^t(S^t, D^t, A'^t) \leq f_r(n_r) - f_r(n_r - c_1) + f_c(n_c) - f_c(n_c + 1)$ 。

当  $f_r(n_r) - f_r(n_r - c_1) < f_c(n_c + 1) - f_c(n_c)$  时,  $g^t(S^t, D^t, A^t) - g^t(S^t, D^t, A'^t) < 0$ , 因此增加检查类型数量会增加总成本。

### 命题 1.3 证明:

同命题 1.2 的构造, 得  $n_c' = n_c + 1, n_r - n_r' \leq d_{j_0}^t$ , 则  $f_r(n_r') \geq f_r(n_r - d_{j_0}^t)$ , 因此当  $f_r(n_r) - f_r(n_r - d_{j_0}^t) < f_c(n_c + 1) - f_c(n_c)$  时, 接受  $j_0$  类检查会增加总成本。

### 命题 1.4 证明:

假设是  $A^t = \{a_{i,j}^t\}$  是预约表  $S^t$  和检查需求  $D^t$  下的最优排程, 且存在  $a_{1,j(i)}^t > a_{1,j(k)}^t, d_{j(i)}^t < d_{j(k)}^t$ 。构造  $A'^t = A^t, a_{1,j(i)}'^t = a_{1,j(k)}^t, a_{1,j(k)}'^t = a_{1,j(i)}^t$ , 则  $a_{T+1,j(i)}'^t = d_{j(i)}^t - a_{1,j(k)}^t, a_{T+1,j(k)}'^t = d_{j(k)}^t - a_{1,j(i)}^t$ 。

$$1) \quad A'^t \text{ 是可行排程}$$

由于  $a_{1,j(i)}'^t = a_{1,j(k)}^t < a_{1,j(i)}^t \leq d_{j(i)}^t, a_{1,j(k)}'^t = a_{1,j(i)}^t \leq d_{j(k)}^t < d_{j(k)}^t$ ,  $A'^t$  满足需求数量约束。由于  $\sum_{j=1}^J (s_{1,j}^t + a_{1,j}^t) = \sum_{j=1}^J (s_{1,j}^t + a_{1,j}^t) \leq M$ ,  $A'^t$  满足检查能力约束, 因此  $A'^t$  可行。

$$2) \quad g^t(S^t, D^t, A^t) = g^t(S^t, D^t, A'^t)$$

由于  $a_{1,j(i)}'^t + a_{1,j(k)}'^t = a_{1,j(i)}^t + a_{1,j(k)}^t$ , 即安排在当天的总量不变, 则  $A^t$  与  $A'^t$  拒绝的检查需求数量相同。由于  $a_{1,j(i)}'^t > 0, a_{1,j(k)}'^t > 0, a_{1,j(k)}^t > 0, a_{1,j(i)}^t > 0$ , 即  $A^t$  与  $A'^t$  当天的检查类型数量相同, 因此  $g^t(S^t, D^t, A^t) = g^t(S^t, D^t, A'^t)$ 。

综上，可以通过构造找到满足  $a_{1,j(1)}^t \geq a_{1,j(2)}^t \geq \dots \geq a_{1,j(J)}^t$  的最优排程。

**定理 1 证明：**

单日规则算法得到  $A^t$ ，假设存在可行排程  $A^{t'}$ ，使  $g^t(S^t, D^t, A^t) > g^t(S^t, D^t, A^{t'})$ ，即：

$$f_r \left( \sum_{j=1}^J a_{T+1,j}^t \right) + f_c \left( \| [s_{1,1}^t + a_{1,1}^t, \dots, s_{1,J}^t + a_{1,J}^t] \|_0 \right) > f_r \left( \sum_{j=1}^J a_{T+1,j}^{t'} \right) + f_c \left( \| [s_{1,1}^t + a_{1,1}^{t'}, \dots, s_{1,J}^t + a_{1,J}^{t'}] \|_0 \right)$$

$$1) \quad \| [s_{1,1}^t + a_{1,1}^t, \dots, s_{1,J}^t + a_{1,J}^t] \|_0 < \| [s_{1,1}^t + a_{1,1}^{t'}, \dots, s_{1,J}^t + a_{1,J}^{t'}] \|_0$$

若当天剩余容量  $c_1 = M - \sum_{j=1}^J (s_{1,j}^t + a_{1,j}^t) = 0$ ，无法增加检查类型数量，因此满足  $\| [s_{1,1}^t + a_{1,1}^{t'}, \dots, s_{1,J}^t + a_{1,J}^{t'}] \|_0$  的排程  $A^{t'}$  不可行。若当天剩余容量  $c_1 = M - \sum_{j=1}^J (s_{1,j}^t + a_{1,j}^t) > 0$ ，单日规则算法保证了对  $\forall j_0 \in \{1, \dots, J\}$  使得  $a_{1,j_0}^t = 0$ ，有  $f_r(n_r) - f_r(|n_r - d_{j_0}^t|) < f_c(n_c + 1) - f_c(n_c)$ 。由命题 1.3 知，增加检查类型数量导致总成本增加。

$$2) \quad \| [s_{1,1}^t + a_{1,1}^t, \dots, s_{1,J}^t + a_{1,J}^t] \|_0 = \| [s_{1,1}^t + a_{1,1}^{t'}, \dots, s_{1,J}^t + a_{1,J}^{t'}] \|_0$$

将  $a_{1,j}^t$ 、 $a_{1,j}^{t'}$  分别从小到大排列，即  $a_{1,(1)}^t \geq a_{1,(2)}^t \geq \dots \geq a_{1,(J)}^t$ ， $a_{1,(1)}^{t'} \geq a_{1,(2)}^{t'} \geq \dots \geq a_{1,(J)}^{t'}$ 。由于单日规则算法从需求量最大的检查类型开始安排，因此  $d_{(1)}^t \geq \dots \geq d_{(J)}^t$ 。由于  $a_{1,j}^t \leftarrow \min \{d_j^t, c_1\}$ ，其中  $c_1$  为当期剩余容量，因此  $a_{1,(1)}^t \geq a_{1,(1)}^{t'}$ ， $a_{1,(2)}^t \geq a_{1,(2)}^{t'}$ ， $\dots$ ， $a_{1,(J)}^t \geq a_{1,(J)}^{t'}$ ，即在检查类型数量相同时，排程  $A^t$  在当期安排的检查数量更多，因此  $\sum_{j=1}^J a_{T+1,j}^t \leq \sum_{j=1}^J a_{T+1,j}^{t'}$ ，排程  $A^{t'}$  的总成本更高，与假设矛盾。

$$3) \quad \| [s_{1,1}^t + a_{1,1}^t, \dots, s_{1,J}^t + a_{1,J}^t] \|_0 > \| [s_{1,1}^t + a_{1,1}^{t'}, \dots, s_{1,J}^t + a_{1,J}^{t'}] \|_0$$

将所有满足  $a_{1,j}^t > 0$  的类型  $j$  记为集合  $J_1$ ，所有满足  $a_{1,j}^{t'} > 0$  的类型  $j$  记为集合  $J_1'$ ，单日规则算法保证了对  $\forall j_1 \in J_1, j_2 \notin J_1$ ，将  $d_{j_1}^t \geq d_{j_2}^t$ 。

构造  $A^{t''} = A^{t'}$ ，对  $\forall j \in J_1$  且  $j \notin J_1'$ ，可以找到不同的  $k \in J_1'$  且  $k \notin J_1$  与之对应，令  $a_{1,j}^{t''} = a_{1,k}^{t'}$ ， $a_{1,k}^{t''} = 0$ 。由于对  $\forall j_1 \in J_1, j_2 \notin J_1$ ，将  $d_{j_1}^t \geq d_{j_2}^t$ ，因此  $A^{t''}$  构造的是可行的排程。由于构造  $A^{t''}$  没有改变  $A^{t'}$  在当期安排的检查数量和检查类型数量，因此排程  $A^{t''}$  与  $A^{t'}$  的成本相同。将所有满足  $a_{1,j}^{t''} > 0$  的类型  $j$  记为集合  $J_1''$ ，由于  $\| [s_{1,1}^t + a_{1,1}^t, \dots, s_{1,J}^t + a_{1,J}^t] \|_0 > \| [s_{1,1}^t + a_{1,1}^{t'}, \dots, s_{1,J}^t + a_{1,J}^{t'}] \|_0$ ，有  $J_1'' \in J_1$ 。根据单日规则算法，存在类型  $j$  使得  $f_r(n_r) - f_r(|n_r - d_j^t|) \geq f_c(n_c + 1) - f_c(n_c)$ ，其中  $n_r$  为排程  $A^{t''}$  拒绝的需求数量， $n_c$  为  $A^{t''}$  接受的检查类型数量。根据命题 1.2，增加一个患者类型，将排程  $A^{t''}$  转化为  $A^t$  不会增加总成本。由于排程  $A^{t''}$  与  $A^{t'}$  的成本相同，因此与假设矛盾。

**命题 2.1 证明：**

对当前预约表  $S^t$  和不完全相同的检查需求  $D^{t1}, D^{t2}$ ，且  $\sum_{j=1}^J d_j^{t1} = \sum_{j=1}^J d_j^{t2}$ ，假设分别存在可行排程  $A^{t1}, A^{t2}$ 。根据开放获取规则，只有在前一期排满时才开放下一期预约，则：

$$\begin{aligned} \sum_{j=1}^J a_{1,j}^{t1} &= \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j=1}^J d_j^{t1}\} = \sum_{j=1}^J a_{1,j}^{t2} \\ \sum_{j=1}^J a_{2,j}^{t1} &= \min \{M - \sum_{j=1}^J s_{2,j}^t, \sum_{j=1}^J d_j^{t1} - \sum_{j=1}^J a_{1,j}^{t1}\} = \sum_{j=1}^J a_{2,j}^{t2} \\ &\dots \dots \\ \sum_{j=1}^J a_{T,j}^{t1} &= \min \{M - \sum_{j=1}^J s_{T,j}^t, \sum_{j=1}^J d_j^{t1} - \sum_{i=1}^{T-1} \sum_{j=1}^J a_{i,j}^{t1}\} = \sum_{j=1}^J a_{T,j}^{t2} \\ \sum_{j=1}^J a_{T+1,j}^{t1} &= \sum_{j=1}^J d_j^{t1} - \sum_{i=1}^T \sum_{j=1}^J a_{i,j}^{t1} = \sum_{j=1}^J a_{T+1,j}^{t2} \end{aligned}$$

因此，当期需求总量相同的排程  $A^{t1}, A^{t2}$  安排在每期的检查/拒绝总数相同，则：

$$[f_w(1), \dots, f_w(T-1), c_r] \left[ \sum_{j=1}^J a_{2,j}^t, \dots, \sum_{j=1}^J a_{T,j}^t, \sum_{j=1}^J a_{T+1,j}^t \right]^T$$

$$= [f_w(1), \dots, f_w(T-1), c_r] \left[ \sum_{j=1}^J a_{2,j}^t, \dots, \sum_{j=1}^J a_{T,j}^t, \sum_{j=1}^J a_{T+1,j}^t \right]^T$$

因此, 给定当前预约表 $S^t$ , 等待成本与拒绝成本的最小值只与当期需求总量 $\sum_{j=1}^J d_j^t$ 有关, 与 $\sum_{j=1}^J d_j^t$ 在 $J$ 类检查中的分配无关。

**命题 2.2 证明:**

由命题 2.1 知, 给定预约表 $S^t$ 和检查需求 $D^t$ , 最优排程 $A^t$ 是使当期转换成本最小的可行排程。假设对最优排程 $A^t$ , 有 $\sum_{j \in JJ_1} a_{1,j}^t \neq \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j \in JJ_1} d_j^t\}$ :

1)  $\sum_{j \in JJ_1} a_{1,j}^t > \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j \in JJ_1} d_j^t\}$

违反需求约束或检查能力约束, 因此 $A^t$ 不可行。

2)  $\sum_{j \in JJ_1} a_{1,j}^t < \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j \in JJ_1} d_j^t\}$

由于 $\sum_{j=1}^J a_{1,j}^t = \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j=1}^J d_j^t\}$ , 因此 $\sum_{j \notin JJ_1} a_{1,j}^t > 0$ 。

构造开放获取规则下的可行排程 $A'^t$ 满足 $\sum_{j \in JJ_1} a'_{1,j} = \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j \in JJ_1} d_j^t\}$ , 则 $\sum_{j \notin JJ_1} a'_{1,j} \leq \sum_{j \notin JJ_1} a_{1,j}^t$ 。由于只有集合 $JJ_1$ 以外的需求在当期产生转换成本, 因此排程 $A'^t$ 对应的转换成本小于等于 $A^t$ 对应的转换成本。

综上, 存在最优排程满足 $\sum_{j \in JJ_1} a_{1,j}^t = \min \{M - \sum_{j=1}^J s_{1,j}^t, \sum_{j \in JJ_1} d_j^t\}$ 。

**定理 2 证明:**

根据命题 2.1, 最优排程是使当期转换成本最小的可行排程。假设开放获取规则得到的解为排程 $A^t$ 。

1)  $\sum_{j \in JJ_1} d_j^t \geq M - \sum_{j=1}^J s_{1,j}^t$

根据开放获取规则算法,  $\sum_{j \in JJ_1} a_{1,j}^t = M - \sum_{j=1}^J s_{1,j}^t$ , 因此排程 $A^t$ 产生的转换成本为 0, 不存在转换成本更小的排程。

2)  $\sum_{j \in JJ_1} d_j^t < M - \sum_{j=1}^J s_{1,j}^t$

根据开放获取规则算法,  $\sum_{j \in JJ_1} a_{1,j}^t = \sum_{j \in JJ_1} d_j^t$ , 当期剩余检查容量为 $M - \sum_{j=1}^J s_{1,j}^t - \sum_{j \in JJ_1} d_j^t$ 。由于安排集合 $JJ_1$ 中的需求不产生转换成本, 根据定理 1 的证明可知, 按照需求量从大到小的顺序填满当期剩余检查容量带来的转换成本最小, 与开放获取规则算法相同。

**命题 3.1 证明:**

假设是 $A^t = \{a_{i,j}^t\}$ 是预约表 $S^t$ 和检查需求 $D^t$ 下的最优排程, 且 $A^t$ 的第 $2 \sim T+1$ 行不完全符合开放获取规则, 即存在 $i$ 使得 $\sum_{j=1}^J (s_{i,j}^t + a_{i+1,j}^t) < M, \sum_{j=1}^J a_{i+1,j}^t > 0$ 。

由于 $\sum_{j=1}^J a_{i+1,j}^t > 0$ , 因此存在检查类型 $j$ 使得 $a_{i+1,j}^t > 0$ 。构造 $A'^t = A^t$ , 令 $a'_{i,j} = a_{i,j}^t + 1, a'_{i+1,j} = a_{i+1,j}^t - 1$ 。由于 $f_w$ 为增函数, 因此 $g^t(S^t, D^t, A') - g^t(S^t, D^t, A^t) = f_w(i+1) - f_w(i) > 0$ , 且 $A'^t$ 是可行排程, 与 $A^t$ 是最优排程矛盾。因此, 最优排程的第 $2 \sim T+1$ 行符合开放获取规则。

**命题 3.2 证明:** 同命题 2.2 的证明。

**命题 3.3 证明：** 同命题 1.4 的证明。

**定理 3 证明：**

假设短视规则算法得到 $A^t$ 。根据命题 3.1 和命题 2.1，给定排程在第一行的取值 $a_{1,j}^t, j = 1, \dots, J$ 后，该排程能取到的最小等待成本和拒绝成本之和是固定的，且 $\sum_{j=1}^J a_{1,j}^t$ 越大，等待成本和拒绝成本之和越小。记所有满足 $s_{1,j}^t > 0$ 的检查类型记为集合 $JJ_1$ ，则对 $\forall j \in JJ_1$ ，增加 $a_{1,j}^t$ 不产生转换成本，只降低等待成本和拒绝成本之和。

$$1) \quad \sum_{j \in JJ_1} d_j^t \geq M - \sum_{j=1}^J s_{1,j}^t$$

集合 $JJ_1$ 中的需求超过当期剩余容量，短视规则算法令 $\sum_{j \in JJ_1} a_{1,j}^t = M - \sum_{j=1}^J s_{1,j}^t$ ，因此不产生转换成本，且受检查容量 $M$ 的限制，此时 $\sum_{j=1}^J a_{1,j}^t$ 最大，等待成本和拒绝成本之和最小，因此 $A^t$ 是最优排程。

$$2) \quad \sum_{j \in JJ_1} d_j^t < M - \sum_{j=1}^J s_{1,j}^t$$

假设存在可行的修改使排程 $A^t$ 转化为排程 $A^{t'}$ ，且 $g^t(S^t, D^t, A^t) > g^t(S^t, D^t, A^{t'})$ 。

由短视规则算法知，排程 $A^t$ 满足：

$$a) \quad a_{1,j}^t = d_j^t, \forall j \in JJ_1$$

$$b) \quad a_{1,j}^t = 0, \forall i = \{2, \dots, T+1\}, \forall j \in JJ_1$$

对排程 $A^t$ 的第一行进行的满足约束的更改有以下几种：

$$a) \quad \text{任取 } j \in JJ_1, \text{ 令 } a_{1,j}^{t'} < a_{1,j}^t$$

对线圈转换成本不产生影响，但增加了患者等待成本，因此 $g^t(S^t, D^t, A^{t'}) > g^t(S^t, D^t, A^t)$ 。

$$b) \quad \text{任取 } j \notin JJ_1, \text{ 令 } a_{1,j}^{t'} < a_{1,j}^t$$

若 $a_{1,j}^{t'} > 0$ ，则线圈转换成本不产生影响，但增加了患者等待成本，因此 $g^t(S^t, D^t, A^{t'}) > g^t(S^t, D^t, A^t)$ ；

若 $a_{1,j}^{t'} = 0$ ，由于在短视规则算法中判断了 $cost_2 < cost_1$ ，因此减少的线圈转换成本不足以覆盖增加的患者等待成本，因此 $g^t(S^t, D^t, A^{t'}) > g^t(S^t, D^t, A^t)$ 。

$$c) \quad \text{任取 } j \notin JJ_1, \text{ 令 } a_{1,j}^{t'} > a_{1,j}^t$$

由于在短视规则算法中判断了 $cost_2 \geq cost_1$ ，因此减少的患者等待成本不超过增加的线圈转换成本，因此 $g^t(S^t, D^t, A^{t'}) \geq g^t(S^t, D^t, A^t)$ 。

综上，无法通过修改排程 $A^t$ 来得到成本更低的可行排程，因此 $A^t$ 是最优排程。