# 基于答案集编程的医疗预约排程问题解决方案

Alina Vozna比萨大学，意大利 信息工程、计算机科学与数学系，拉奎拉大学，意大利
alina.vozna@student.univaq.it

Andrea Monaldini
意大利比萨大学
阿奎拉大学信息工程、计算机科学与数学系，意大利；
andrea.monaldini@student.univaq.it

Stefania Costantini
信息工程、计算机科学与数学系，阿奎拉大学，意大利
stefania.costantini@univaq.it

Valentina Pitoni
信息工程、计算机科学与数学系，阿奎拉大学，意大利
valentina.pitoni@univaq.it

Dawid Pado
信息工程、计算机科学与数学系，阿奎拉大学，意大利
dawid.pado@student.univaq.it

本文提出了一种基于答案集编程（ASP）的医疗预约排程框架，旨在提高效率、减少行政负担，并加强以患者为中心的照护。该框架通过整合 Blueprint Personas，为弱势群体提供个性化排程。通过在 ASP逻辑模型中集中规划操作，该框架确保实时更新可用性、无冲突分配，并与现有医疗保健平台实现无缝互操作。

## 1 引言

在现代医疗保健系统中，高效管理医疗预约是确保患者能够及时、公平地获得医疗服务的一项关键挑战。老龄人口增加、慢性病患病率上升以及医疗资源有限，推动医疗服务提供日益复杂，也凸显了传统排程方法的局限性。这些系统往往未能考虑预约分配中涉及的诸多临床、后勤和个人约束。减少等待时间、提高患者满意度和优化资源利用率等关键目标，常因静态的最早可用分配规则以及缺乏个性化安排而难以实现。

M. Gebser、D. Inclezan、F. Ricca、M. Carro 和 M. Truszczynski：第41届国际逻辑编程会议（ICLP 2025），EPTCS439，2026年，第367–382页，doi:10.4204/EPTCS.439.25

© Vozna、Monaldini、Costantini、Pitoni 与 Pado 本作品采用知识共享署名许可协议。

意大利的集中式预约基础设施（Centro Unico di Prenotazione – CUP）等传统系统，正体现了这些不足。分配预约时，通常不会考虑医疗紧急程度、患者所在地或个人偏好。这导致排程效率低下、爽约率高，并增加了重新安排预约和人工调整的行政负担。此外，僵化的框架缺乏灵活性，无法应对人员短缺或紧急事项优先处理等实时变化，导致医疗服务延迟和资源分配不当。

为应对这些挑战，本文提出一种基于约束的自动化排程系统，采用答案集编程（ASP）[6, 20, 9, 21]。ASP是一种声明式逻辑编程范式，非常适合解决涉及复杂动态约束的NP完全问题。利用ASP强大的表达能力，所提出的系统能够自动排除不可行的解，例如残障患者无法使用无障碍设施，或时间段彼此不兼容的情况；并生成最优排程，以尽量减少与距离、等待时间、感官不适和偏好违背相关的惩罚。

本研究的一项独特之处在于整合了Blueprint Personas [23], 结构化模型，用于刻画不同患者原型的临床、社会和行为属性。Blueprint Personas最初源自欧洲数字健康倡议，通过将认知局限、无障碍需求和社会经济状况等现实约束纳入排程过程，支持以患者为中心的护理。使用这些模型能够提升公平性和个性化程度，并促进患者遵循护理计划，尤其有利于弱势群体。

本文开发的基于答案集编程的系统旨在部署于集中式或半集中式医疗基础设施，因此尤其适用于 CUP 等国家级系统。尽管 CUP 支持集中式访问，但其预约分配逻辑较为简单，往往无法充分考虑紧急程度、患者偏好或无障碍性约束等关键因素 [14]。相比之下，我们的方法支持实时重新排程、紧急情况优先排序和伦理决策，这些能力对于现代医疗环境至关重要，尤其是在资源匮乏或需求旺盛的情况下 [4]。尽管该系统非常适用于意大利 CUP 这类集中式基础设施，但也可以通过将排程逻辑分散至多个机构，适配联邦式系统，并通过共享约束保持一致性。

此外，本研究展示了答案集编程如何支持预约排程中的可解释多目标优化。与传统的基于规则或启发式方法不同，答案集编程提供了完整的声明式建模能力，使系统能够在可扩展且适应性强的框架内，系统地整合患者偏好、医疗优先级和组织约束。该解具有模块化和通用性，也适用于劳动力管理、教育课程排程等其他领域。

本文结构如下：第 2 节讨论医疗预约排程及基于答案集编程的优化的背景与相关工作；第 3 节介绍问题的形式化描述，详细说明排程模型的约束和目标；第 4 节描述基于答案集编程的系统架构与实现；第 5 节对该方法进行实证评估，并将其性能与传统排程方法进行比较；最后，第 6 节总结全文并讨论未来的研究方向。

## 2 相关工作

预约排程已通过排队论、整数规划和仿真模型得到广泛研究，旨在应对服务时间变异性、患者爽约以及医疗资源有限等常见挑战 [14]。近期研究引入了决策支持工具

以优化门诊场景中的排程 [25],，并整合数据驱动和基于逻辑的技术来管理复杂约束。

多种优化方法，例如混合整数线性规划（MILP）、答案集编程（ASP）和元启发式算法，在处理超额预约策略、资源平衡和患者流管理方面展现出良好成效。尤其是随机规划，在提高患者就诊机会的同时维持资源的高效利用 [19]。模糊蚁狮优化（FALO）等启发式技术，在不影响资源利用效率的前提下提升了公平性并缩短了患者等待时间 [3]。离散事件仿真（DES）结合患者行为建模与遗传算法，也增强了排程对偏好、取消事项和动态需求的适应能力 [12]。

答案集编程（ASP）在资源受限的医疗场景中已证明尤为有效，包括手术室排程 [10] 和护士排班 [5]。值得注意的是，Cappanera 等人 [8] 在 ASP 中采用基于逻辑的 Benders 分解（LBBD），管理患有合并症的慢性病门诊患者的排程，并通过将日级规划与每日日程编制分离，提高了可扩展性。同样，Kanias [17] 开发了一种公平性感知的基于答案集编程的系统，通过 Python-CLORM 集成 SQL 数据库，并支持大规模医疗预约重新排程，重点在于最大化社区受益。

人工智能（AI）方法进一步提升了排程能力。强化学习和深度学习已被用于从历史预约数据中学习最优策略，从而能够根据不断变化的患者行为进行动态调整。特别是，Ala 等人 [2] 提出了一种基于双向长短期记忆网络（Bi-LSTM）和强化学习的深度学习框架，用于在不确定性条件下优化预约排程。该双向长短期记忆网络通过分析历史预约日志，预测患者行为，例如到诊可能性或偏好的时间窗口。随后，这些预测结果被用于强化学习循环中，以动态更新排程策略。该模型旨在最大限度地减少等待时间，并降低复杂临床系统中的闲置资源。

尽管这些基于学习的方法在发现时间模式方面十分强大，但它们通常需要大量结构化训练数据，并且决策逻辑缺乏可解释性。此外，它们无法原生支持丰富的符号约束，例如无障碍需求、公平规则或伦理优先事项。我们基于答案集编程的方法通过支持对患者档案、偏好和临床政策进行透明的基于规则的推理，弥补了这些局限，从而实现更好的可解释性，并更好地融入约束密集型场景。

混合人工智能模型——例如将进化算法与两阶段随机规划相结合的模型——已用于磁共振成像排程等高不确定性环境 [27]。此外，采用分布式约束优化问题（DCOPs）的去中心化方法将排程构建为多智能体协商过程 [16]。

然而，尽管这些技术颇为复杂，理论模型与现实医疗运营之间仍存在长期未能弥合的差距。许多框架忽略了关键的实际因素，例如临床医生可用性的波动、患者异质性，以及重新排程带来的行政负担[18]。我们工作的一个重要区别在于，它与现实医疗基础设施和政策相契合，尤其是从国家级系统的角度出发。例如，在意大利，预约排程主要由 Centro Unico di Prenotazione（CUP）管理；这一集中式系统会分配最早可用的预约，而不考虑紧急程度、无障碍性或患者偏好。这会导致工作流程僵化、爽约率偏高，并在临床医生缺勤或临近取消等意外事件发生时造成效率低下。相比之下，一些国际系统则采用了不同程度的灵活性和个性化服务。例如，英国的 e-RS 平台允许患者

以选择他们偏好的日期、时间和地点；法国的 Doctolib 提供视频咨询的数字化访问和个性化预约设置；德国的 eTerminservice 则确保紧急转诊获得优先级排序。这些系统虽然适应性更强，但通常依赖定制集成，且缺乏答案集编程所具备的逻辑严谨性和约束表达能力。

与启发式方法、机器学习或随机方法不同，我们的基于 ASP 的排程框架能够确保完全遵守硬约束，提供可解释的决策，并支持实时适应不断变化的医疗保健需求。此外，许多以往的系统都侧重于优化吞吐量或资源利用率，却很少全面考虑以患者为中心的约束，例如感官偏好、无障碍要求、多次就诊规划和经济限制。我们的解决方案通过将排程建模为声明式逻辑问题，弥补了这一不足，确保在切合实际的运营边界内实现合乎伦理且个性化的医疗服务获取。

## 3 问题描述与形式化

有效管理医疗预约是一项复杂的多维问题，需要在临床紧急程度、有限的医疗资源、患者特定偏好和后勤约束之间取得平衡。传统排程方法往往难以胜任，因为它们难以在统一且灵活的框架内协调这些相互竞争的因素。

为克服这些局限，我们将预约排程表述为一个多目标约束优化问题，并使用答案集编程（ASP）求解。答案集编程的声明式特性使领域知识能够以简洁优雅的方式编码，从而可以在庞大的可能解空间中进行自动推理，同时严格执行所有约束。

该问题围绕三个核心实体展开：诊所，在固定预算限制下提供预约时段；医生，其可用性和工作量必须得到平衡；以及患者，他们的紧急程度各不相同（高、中、低），并可能表达对时间窗口、诊所地点、特定医师乃至环境条件（例如噪声或拥挤程度）的偏好。

排程的目标是为每位患者分配一个或多个可行的预约时段，同时确保严格遵守所有硬约束，例如临床适配性、医生可用性、无障碍性和资源限制。与此同时，系统还力求优化软目标，包括尽量缩短患者等待时间、满足表达的偏好、减轻临床人员超负荷，并提高医疗保健基础设施的整体利用率。

在我们的 ASP模型中，这项任务被建模为一个经过优化准则增强的约束满足问题（CSP）。硬约束会剔除无效分配，而加权软约束则引导搜索，找到最理想的结果——实现自动化、可解释且以伦理为基础的调度策略，以适应现代医疗服务提供在现实世界中的复杂性。

### 3.1 问题形式化表述

我们将医疗预约排程任务形式化为一个约束优化问题，定义如下。

给定：

- 一个患者集合 P，其中每位患者由优先级 $\text{priority}(p) \in \{\text{高、中、低}\}$、临床需求，以及对时间段、医生、诊所和环境条件的一组偏好所表征。
- 一组医生 D，每位医生都具备特定的专业技能并受到容量约束。

- 一组诊所 C，每家诊所都有可用时间段、无障碍功能和预算限制。
- 一组可用时间段，分布在各诊所和医生之间。

约束：

- 必须为每位患者安排符合其治疗计划所要求数量的预约。
- 分配必须遵守硬约束：无障碍要求、诊所预算、就诊兼容性以及临床医生可用性。
- 同一位医生、同一家诊所和同一时间段不得重复预约。

目标：找出一个分配函数 $f : P \rightarrow (D \times C \times S)$，使其：

1. 最小化患者总体等待时间。2. 最大化患者偏好的满足度（偏好的诊所、医生、时间窗口和环境条件）。3. 确保医生之间公平的工作量分配。4. 最小化运营惩罚总量（感官不匹配、资源过度使用）。

这一形式化过程自然对应于一个答案集编程（ASP）模型，其中可行分配对应于满足约束和优化目标的答案集。

## 3.2 用户建模与人物画像

有效的预约排程不仅需要管理临床数据，还要求理解具有独特需求、偏好和约束的患者个体。因此，我们采用
Blueprint Personas，这是一种最初在欧洲数字健康倡议（EIP on AHA） [11], 中开发的结构化原型，用于概括结合临床、社会、认知和行为属性的真实世界档案。Blueprint Personas 在原始数据与决策逻辑之间建立了一层抽象，使我们的系统能够以灵活且以人为本的方式推理个性化排程需求。

患者画像：每个患者画像由健康状况、社会环境因素和数字素养构成；该模型使系统能够推理复杂且个性化的排程需求。尽管目前数字素养被纳入患者档案的一部分，但它不会直接影响排程逻辑。纳入数字素养旨在支持更广泛的系统集成，例如指导界面选择（如基于应用程序的访问或照护者介导的访问），或筛选可行选项，例如远程医疗。患者复杂程度分为三个级别：

- 一般健康人群：偏好便捷的排程方式（如清晨时段、就近诊所），但没有严格的临床约束。
- 慢性病患者：需要定期随访、预算敏感型分配和尽可能短的等待时间；排程必须符合多次就诊间隔和费用上限。
- 具有复杂需求的患者：可能表现出感官敏感性（例如对噪声、光照敏感），需要无障碍设施，或依赖照护者出行；系统必须尊重个性化偏好和环境限制。

关键个人数据用于识别患者，并可表达其对诊所、时间段、感官状况和医生的偏好。同时还会考虑出行距离，其中居家护理和远程医疗的固定值为 0。

```
patient(p1,"Mario","Rossi","L'Aquila").disabled(p1).pr
```

列表 1：包含偏好和约束的患者档案

临床医生角色画像：对医疗专业人员进行建模同样至关重要；每位临床医生的人物画像至少描述其专科、技术投入和运营挑战。这使系统能够预判实际的工作负荷约束，并优化医疗协调。

对于每家可用的诊所，就诊类型代表该诊所所提供治疗的特征。每次就诊由以下信息标识：名称、费用，以及用于表示慢性程度的分类（0 =非慢性， 1 = 慢性），还有是否需要现场就诊。有些就诊需要多个预约时段，且必须遵循最短和最长间隔：

```
就诊_类型(v1,"心脏病学","心脏病发作",0,0,0).就诊_费用(v1,
```

清单 2：就诊类型定义与约束

诊所可用性由 Unix 时间戳定义，用于标记医疗预约的具体时间。每个患者请求都包含一个紧急程度（1–3）和偏好的预约时段间隔，从而能够根据个人需求和预算进行最优排程。

```
patient_interval(p1,v1,21,28).需求
```

清单 3：患者需求偏好与诊所可用性

这种建模使系统能够针对患者特定约束进行推理，并确保解决方案在医学上适当且以患者为中心。人物画像还有助于基于场景的验证：可以让模拟患者通过求解器运行，以评估系统对边缘案例和弱势用户的适应能力。

## 3.3 推理规则

基于答案集编程的模型中的推理规则从基础事实推导出中间知识，从而在简单的约束过滤之外进一步提升优化效果。这些规则不作为硬约束，也不用于选择规则，而是用于为特定偏好赋予效用值，并在优化阶段汇总这些效用值。

患者偏好（例如偏好的诊所）会被转换为加权指标：

```
  诊所_偏好_效用（患者，诊所，1）：-2偏好（患者，诊所）。诊所_
```

```
偏 好 ( 患者 , 诊所 ).
```

列表 4：诊所偏好对患者分配的影响

同样，医生偏好根据专科和工作年限进行评估：

```
医 生 _ 偏 好 _ 影 响 ( 患者 ,                    医生 ,     1）  :-
      患者(患者,              _ ,  _ ,  _),
      医生(医生,          _ ,  _ ,  _ ,  _ ,  类型),
      医生_经验(医生,                    专门化,              年 限 经 验 ) ,
      医生_偏好（患者,                    类型,    专 业 化 ,              所需年限）,
      工作年限  >=所需年限。
医生_ 偏好_ 效果（患者,                    医生,     0）  :-
      患者（患者,              _ ,  _ ,  _）,
      医生（医生,          _ ,  _ ,  _ ,  _ ,  类型）,
      医 生 _经 验 ( 医生 ,                    专 业 领 域 , _) ,
      非 医 生 _ 偏 好 ( 患者 ,                    类型 ,   专 业 领 域 , _) .
医 生 _ 偏 好 _ 影 响 ( 患者 ,                    医生,     0）  :-
      患 者( 患者,              _ ,  _ ,  _),
      医生 ( 医生,          _ ,  _ ,  _ ,  _ ,  类型 ),
      医 生 _经 验 ( 医生,                    专门化,              年 限 经 验 ) ,
      医生_偏好（患者,                    类型,    专 业 化 ,              所需年限）,
      所需年限  <年。
```

列表 5：患者偏好对医生分配的影响

同样的逻辑也适用于时段偏好和感官偏好，从而生成 appointment preference effect/4 和 sensory penalty/4 等谓词。

appointment preference effect/4 用于判断给定的预约时间是否符合患者的偏好时间范围：如果该时间落在指定范围（开始时间至结束时间）内，则返回 1；否则返回 0。时间会转换为使用小时和分钟表示的加权分数，以便与偏好区间进行比较；此值可用于优化，以优先安排患者偏好的时间段。

```
预约_ 偏好_效果( 患者 , 时间 , 诊所 , 1) : - 2 患者( 患者 , _,  _,  _) , 3 诊所 ( 诊所 , _) , 4 可用性 ( 诊所
,
1
时
偏
6
7
```

```
    预约排程_偏好(患者,_,                    开始时间, 结束时间),
    X = (((时间 \86400) * 3600) * 100) + (((时间 \3600)/60)/3)*
        5,
    X < 开始时间。
```

列表 6：时间偏好对预约排程的影响

感官惩罚/4规则有助于确定诊所环境条件与患者表达的偏好之间的差距。

```
sensory_penalty(患者,诊所,时间,级别):-sensory_preference(
患
时
时
```

列表 7：基于环境条件的感官惩罚

上文定义的所有推理规则都用于以下所示的优化语句中。它们并非用于强制实施约束，而是作为优化的软指标。ASP 求解器在最大限度满足患者偏好的同时，最小化出行距离和等待时间。对于环境不匹配情况会增加惩罚，而满足偏好时则会给予奖励项。

```
#最小化{
    (距离*10000) +等待时间 +(惩罚                    *1000)-
    (诊所偏好*10000)-
    (医生偏好*1000)-
    (预约偏好*1000)                    :
        distance(患者 , 诊所 , 距离)  ,
        appointment(患者, 诊所, 医生, _, 时间),
        当前_时间(当前时间)  ,
        等待时间 =时间-当前时间  ,
        感知_惩罚(患者, 诊所, 时间, 惩罚),
        诊所_偏好_效应(患者, 诊所, 诊所偏好),
        医生_偏好_效应(患者, 医生, 医生偏好)
        预约_偏好_效果(患者, 时间, 诊所,
        预约偏好)
}.
```

清单8：基于偏好和约束、使费用最小化的目标函数 s

该优化函数在硬约束（第3.4节）与以患者为中心的效用值之间取得平衡，使系统能够生成高质量、个性化且符合伦理原则的日程安排。

## 3.4 约束

约束定义了答案集编程（ASP）模型生成的每个解都必须遵守的逻辑规则。它们确保预约按照临床、个人和后勤要求进行分配。

首先，必须根据每位患者的具体医疗状况，为其安排各项治疗所需的确切预约次数。这可确保涉及多个预约时段的治疗能够正确分配到可用时间段。

```
场 次{ 预 约（患者, 诊所, 医生, 就诊, 时间） :
可用性（诊所, 医生, 就诊, 时间) } 课程：-
需求（患者, 就诊, _), 必需的_次就诊 次就诊）。
```

代码清单 9：基于患者需求的预约分配选择规则

此外，为避免排程冲突，不能同时为两位患者安排相同的诊所、就诊类型和时间段。

```
:- 预约 ( P1 , 诊所 , 医生 , 就诊 , 时间 ) , 预约 ( P2 , 诊所 , 医生 , 就诊 , 时间 ) , P1 ! = P2 。
```

代码清单 10：防止同一时段重复预约的约束

对于紧急程度高的患者也给予优先安排，必须在预定义时间阈值内为其安排较早的时段。

```
:- 需求 ( P1 , 就诊 , Urg1 ) , 需求 ( P2 , 就诊 , Urg2 ) , Urg1 >Urg2 , 预约 ( P1 , 诊所 ,
```

清单 11：优先安排紧急就诊的约束

同时也会考虑无障碍需求：残障患者只能被安排到明确标注为无障碍的诊所。

```
:- 残障(患者) , 预约(患者 , 诊所 , _ , _ , _ ) , 非无障碍(诊所)。
```

清单 12：防止分配到无法进入的诊所的约束

该模型通过确保已安排的慢性病护理预约总费用不超过各医疗机构分配的资源，来落实诊所方面的预算限制。这反映了公共医疗中的现实约束，例如补贴服务的限制。

目前尚未对患者方面的财务约束（例如支付能力）进行建模，但这类约束在私营或混合医疗体系中具有重要意义。要纳入这些约束，需要扩展患者档案，这项工作计划作为未来工作开展。

```
:- c h r o n i c _ c o s t ( 诊所 , T ot al Co st ) , budget ( 诊所 ,
```

清单 13：遵守诊所预算限制的约束

一些就诊可以通过远程医疗或居家护理进行，但另一些则要求医生亲自到场，必须在现场进行；此约束可防止将就诊安排到类型不匹配的诊所。

```
:- a p p o i n t m e n t ( 患者 , 诊所 , 医生 , 就诊 , 时间 ) , visit_t y p e (
```

```
    就诊_类型(就诊,     _, _, _, _, 1),
    诊所(诊所       , "远程医疗").
```

清单 14：不兼容的就诊类型和诊所模式的约束

## 4 系统架构

该系统由一组松耦合的微服务构成，用于管理预约请求，并基于答案集编程逻辑生成最优排程。系统分为三个主要层：用户界面、业务逻辑和数据管理，以确保可扩展性、可维护性，并能与现有医疗基础设施集成。

用户层包括网页界面（例如浏览器）和物联网设备，它们可能会在健康参数达到阈值时自动触发预约请求。

业务逻辑层由两个基于 Flask 的微服务组成 [24]:

- MS1：处理传入的 API 请求、用户身份验证（通过 JWT 令牌 [15]），并将数据格式化为 ASP 事实。
- MS2：将事实收集到批处理队列中，并每隔 60 秒将其编译成完整的 ASP 程序；随后使用 Clingo 求解该程序 [13]。

数据层依托 MySQL 数据库来维护关键信息，例如患者偏好、诊所特征、预约可用性、环境条件和预算约束。所有数据交互均遵循 ACID 原则。

系统提供多个用于后端交互的 REST API，包括提交预约请求、查看进行中的预约状态，以及查看各诊所最新可用性的端点。异步线程可防止高峰时段系统过载，同时为用户提供即时反馈并延迟交付结果。

凭借其分层架构、通过 ASP 实现的集中式约束管理，以及与 Python 后端的无缝集成，该系统展现出处理复杂排程场景的有效性。此外，基于微服务的结构允许独立更新或扩展各个组件（例如求解器、用户界面或数据库），而不会影响整个系统。完整源代码已在网上公开提供 [1]。

## 5 实验评估

为展示所提出的基于 ASP 的排程框架的灵活性和稳健性，我们介绍了三个受真实医疗排程需求启发的现实排程场景。实验在一台配备 Intel Core i7-9750HF CPU @ 2.60GHz、16GB RAM 和 Windows 11 的计算机上运行，使用 Clingo 5.7.2 版。 [2]。

本节展示的场景基于合成生成数据。尽管这些数据并非源自真实世界的记录，但每个示例都经过精心设计，旨在反映患者偏好、临床要求和环境条件的真实组合。所有测试输入均编码为

[1]https://github.com/DawidPado/An-ASP-based-Solution-to-the-Medical-Appointment-Scheduling-Problem/tree/main[2]https://github.com/potassco/clingo/releases/tag/v5.7.0

ASP 事实，并且完全可复现。这些场景旨在展示系统在不同配置下兼顾伦理与约束的行为。

每个场景都突显了系统处理无障碍性、紧急程度和患者个性化偏好等关键约束的能力。清单8所述的优化模型在指导排程决策方面发挥着核心作用。

## 5.1 场景 1：有无障碍要求的高优先级患者

在第一个场景中，一名有残障且医疗需求优先级较高的患者需要预约心脏科。患者 p1 必须安排就诊类型 v1（例如心脏科会诊），紧急程度为 3。由于患者的身体状况，该预约只能安排在无障碍诊所。清单 15 以压缩形式概述了场景 1 的关键 ASP 事实。

```
% Scenario 1: High-priority disabled patient (compressed facts)医生 (
```

清单 15：场景 1 的 ASP 事实——高优先级残障患者

为确保 患者仅被安排到无障碍诊所，使用了清单 12 中的约束 .

结果：优化器选择了无障碍诊所中尽可能早的时间，且出行距离最短。由于未提供感官偏好，惩罚为零；该解在硬约束和最小化函数方面均为最优。计算时间：0.036s 最优性：是

## 5.2 场景 2——感官偏好

在此案例中，患者表达了对低噪声环境的偏好。虽然有一家距离更近的诊所，但优化器选择了距离更远、符合环境标准的诊所：

```
% Patient and sensory preference
患者(p2, "Giulia", "Bianchi"). sensory_preference(p2, "光").
% Clinics and environmental conditions
诊所(c3 , "诊所 A")。诊所（c4 , "诊所 B")。
环境条件_（c1, "光", 3, 1727480000, 1727489000).
环境_条件（c2, “光”, 1, 1727480000, 172748900）。
距离(p2, c3, 10)。 距离(p2, c4, 25)。
% Type of visit requested
就诊_类型(v2, "骨科", "脊柱后凸", 0, 1, 1).
次就诊_费用（v2, 1500）。需求（p2, v2, 2).
% Slot availability
可用性（c3, m1, v2, 1727481600). % 28.09.2024, 12:00
%...
```

列表 16：场景 2 的 ASP 事实——亮度感官偏好

为评估环境条件对患者满意度的影响，我们应用了列表 7 中已介绍的相同感官惩罚规则。这些规则用于评估患者感官偏好与诊所环境条件之间的匹配程度，确保在排程过程中尽量减少不适因素（例如光、噪声）。因此，优化函数能够正确地优先安排符合患者感官需求的预约。

结果：由于感官惩罚较低，该患者被正确分配至诊所 B。尽管距离成本较高，优化逻辑仍优先考虑感官舒适度，这体现在目标函数中的 sensory penalty/4 项。计算时间：0.017s 最优性：是

## 5.3 场景 3 – 群组优先级排序

五位患者请求在同一家诊所进行心脏科就诊。他们在以下方面有所不同：紧急程度、偏好（诊所、感官）以及与诊所的距离。模型必须根据紧急程度确定优先级，同时尽量减少惩罚。

```
% Patients and urgency levels
patient(p1,"Mario","Rossi").needs
```

清单 17：场景 3 的 ASP 事实——在时段有限的情况下安排 5 位患者的优先级 s

结果：所有患者均已正确安排，同时兼顾紧急程度和偏好，并尽量降低总成本。计算时间：0.017 秒 目标函数值：1,610,000 结果分析：

紧急程度较高（3）的 Mario（p1）被赋予最高优先级，并安排在 12:00 的第一个可用时段。紧急程度均为中（2）的 Giulia（p2）和 Anna（p4）被安排在接下来的时段，其中 Giulia 因出行距离较短而优先安排。紧急程度较低（1）的 Luca（p3）和 Paolo（p5）被安排在最后两个时段。尽管 Luca 的出行距离略长于 Paolo，但由于符合他的诊所偏好，安排 Luca 优先就诊降低了总成本。

Clingo 返回的解的优化得分为 1,610,000。该数值是 #minimize 指令中定义的加权和的结果，反映了所提议日程安排的总成本。其中包括出行距离、延迟和感官不匹配带来的惩罚，以及满足个人偏好所获得的奖励项。得分越低，解越好。

## 5.4 结果概述

实验评估证明了基于答案集编程的排程系统在各种现实场景中的有效性和适应能力。主要发现凸显了该系统能够兼顾

| 患者 | 紧急程度 | 偏好 | 分配时段 | 惩罚 | 优先级正常 |
|---|---|---|---|---|---|
| p1 | 高 (3) | 诊所 | 12:00 | 0 | ✓ |
| p2 | 中 | 感官 | 13:00 | 0 | ✓ |
| p3 | Low | 诊所 | 15:00 | 0 | ✓ |
| p4 | 中 | 感官 | 14:00 | 0 | ✓ |
| p5 | Low | – | 16:00 | 0 | ✓ |

表1：患者分配摘要（场景 3）

相互冲突的目标，同时遵守复杂约束，确保效率并以患者为中心提供护理。

| 场景 | 满足的约束 | 满足的偏好 | 时间（秒） | 最优 |
|---|---|---|---|---|
| 残障患者 | ✓ | ✓ | 0.036 | ✓ |
| 感官偏好 | ✓ | ✓ | 0.017 | ✓ |
| 多名患者 | ✓ | ✓ | 0.017 | ✓ |

表2：实验结果汇总

这些实验验证了该模型处理以下情况的能力：

• 复杂多约束调度：在所有场景中，系统都严格遵守了硬约束。无效解（例如，将残障患者安排到无法进入的诊所）会由 ASP 求解器自动排除。

• 个性化偏好：即使与距离最小化相冲突，系统也会尊重感官偏好（场景 2）和诊所选择，这体现了系统能够根据个体患者档案定制日程安排。

• 高效优化：即使在多患者排程（场景 3）中，计算时间仍然很短（$<0.04$s），这证实了系统适用于实时决策。ASP 求解器始终能够生成最优且符合约束的解。

• 公平性与优先级排序：高紧急度患者（例如场景 1）会优先获得最早可用时段，从而最大限度地缩短等待时间。在群组调度（场景 3）中，时段分配由紧急程度决定，同时对偏好和惩罚进行集体优化，以确保公平的资源分配。

基于 ASP 的系统展现出稳健性、灵活性，并适用于现实世界的医疗调度任务。结果验证了该系统在医疗场景中的适用性；在这些场景中，快速应对取消事项、紧急情况或资源可用性的波动至关重要。在所有测试案例中，ASP 求解器均在 0.04 秒内生成最优排程，这证实了即使面对复杂约束，实时决策也是可行的。

尽管所呈现的场景仅用于说明，但该 ASP 编码旨在通过受控实例化支持大规模排程。由于大多数规则都针对特定患者和时段，且不涉及跨患者聚合，实例化原子的数量在最坏情况下会以 $O(p \cdot c \cdot s)$ 的速度增长，其中 p 为患者数量，c 为诊所数量，s 为时段数量。

在实践中，通过为每位患者筛除不相关的诊所和时间段，可以减少基化规模。在使用自定义 Python 生成器进行的测试中，设置为 50 位患者、6 家诊所和 500 个时间段时，基化原子

数量保持在 10,000 以下，求解时间不到 0.1 秒。在竞争程度较低且就诊类型多样化的情况下，性能依然稳定。

虽然已计划开展完整的可扩展性研究，但这些初步测试支持这样一个结论：该模型在现实条件下呈线性扩展。

# 6 结论与未来工作

医疗预约排程仍然是医疗运营中的关键瓶颈，在满足患者需求与优化资源之间取得平衡始终面临挑战。本文提出了一个基于ASP的框架，通过完全声明式的约束建模和多目标优化，系统地应对这些挑战。该框架展现出快速的解算速度，能够始终在0.04秒内生成最优排程，验证了其在实时医疗场景中的可行性。

通过整合Blueprint Personas、患者优先级、感官偏好、无障碍要求和诊所预算，该框架支持为弱势群体制定个性化的医疗服务路径，例如患有慢性病或具有感官敏感性的患者。这些结果弥合了理论优化模型与实际医疗保健需求之间的差距，为由启发式方法驱动的方案提供了一种可扩展、透明且符合伦理原则的替代方案。

尽管当前评估侧重于受约束的场景，但仍需进一步开展实验，以评估模型在极端负载（例如数百个并发排程请求）和高度动态环境（例如频繁发生紧急情况）下的性能。此外，纳入用于估算爽约情况的预测分析，可以进一步增强排程的稳健性。未来的工作还包括扩展该框架，以支持神经多样性患者在城市通勤期间的需求，从而扩大其社会影响。

总而言之，基于答案集编程的方法为预约排程提供了灵活、高效且符合伦理的解，在弥补现有系统关键不足的同时，促进了临床有效性和以患者为中心的护理。

可扩展性与未来方向。我们的实验结果表明，该框架能够以较短的解计算时间有效处理现实世界中复杂的排程问题。未来研究将重点关注扩展系统，以适应涉及数百或数千条排程请求的更大规模实例。计划中的改进包括增量式答案集编程求解、求解器组合技术，以及分层排程分解，从而确保在高需求环境中的计算效率。

推广潜力。尽管当前系统面向医疗预约排程，但其底层的基于答案集编程的框架与具体领域无关，具有良好的适应性。通过修改输入事实和约束编码，同一架构可应用于广泛的排程任务，包括：

- 医疗保健、制造业和服务行业中的人员排班。
- 学校和大学的课程时间表编排，同时纳入偏好和资源约束。
- 医疗机构预订和活动排程，并进行后勤优化。

这种普适性凸显了基于 ASP 的声明式优化在动态、真实世界规划领域中的广泛适用性。

## 致谢

本研究部分由 PNRR项目 CUP E13C24000430006“用于构建宜居环境的智能代理增强网络 - ENABLE”资助；同时也由 PRIN 2022 CUP E53D23007850001 项目 TrustPACTX——人类—自主系统混合社会的设计：架构、可信度、信任、伦理与可解释性（患者护理案例）资助；此外还获得 PRIN PNNR CUP E53D23016270001 ADVISOR——用于可信健康指导的自适应易读机器人资助。

## 参考文献

[1] Ali Ala、Fawaz E Alsaadi、Mohsen Ahmadi 和 Seyedali Mirjalili（2021）：基于鲸鱼优化算法和 NSGA-II，针对医疗保健系统中的预约排程问题进行优化，以提升服务公平质量。《Scientific Reports》11(1)，第 19816 页，doi:10.1038/s41598-021-98851-7。[2] Ali Ala 和 Feng Chen（2022）：医疗保健服务复杂性系统中的预约排程问题：全面综述。《Journal of Healthcare Engineering》2022(1)，第 5819813 页，doi:10.1155/ 2022/5819813。[3] Ali Ala、Vladimir Simic、Dragan Pamucar 和 Erfan Babaee Tirkolaee（2022）：医疗保健服务中基于公平政策的预约排程问题：模糊蚁狮优化器。《Expert Systems with Applications》207，第 117949 页，doi:10.1016/j.eswa.2022.117949。[4]Mahmoud H Alrefaei 和 Ali Diabat（2015）：门诊预约排程的建模与优化。《RAIRO-Operations Research-Recherche Op´erationnelle》49(3)，第 435–450 页，doi:10.1051/ro/ 2014041。[5] Mario Alviano、Carmine Dodaro 和 Marco Maratea（2019）：通过答案集编程进行护士排班。《Intelligenza Artificiale》12(2)，第 109–124 页，doi:10.3233/ia-170030。[6] Chitta Baral（2003）：知识表示、推理与声明式问题求解。剑桥大学出版社。[7] J¨urgen Branke、Su Nguyen、Christoph W Pickardt 和 Mengjie Zhang（2015）：生产排程启发式算法的自动化设计：综述。《IEEE Transactions on Evolutionary Computation》20(1)，第 110–124 页，doi:10.26686/wgtn.13151315。[8] Paola Cappanera、Marco Gavanelli、Maddalena Nonato 和 Marco Roma（2023）：答案集编程中用于慢性病门诊患者排程的基于逻辑的 Benders 分解。《Theory and Practice of Logic Programming》23(4)，第 848–864 页，doi:10.1017/s147106842300025x。[9] Stefania Costantini 和 Andrea Formisano（2010）：包含资源的答案集编程。《Journal of Logic and Computation》20(2)，第 533–571 页，doi:10.1093/logcom/exn082。[10] Carmine Dodaro、Giuseppe Galat`a、Martin Gebser、Marco Maratea、Cinzia Marte、Marco Mochi 和 Marco Scanu（2024）：通过答案集编程进行手术室排程：改进编码并使用真实数据进行测试。《Journal of Logic and Computation》34(8)，第 1556–1579 页，doi:10.1093/logcom/exae041。[11] 欧盟委员会（2024）：积极健康老龄化欧洲创新伙伴关系。https://ec. europa.eu/eip/ageing/home_en.html。访问日期：2024-09-23。[12] Xuanzhu Fan、Jiafu Tang、Chongjun Yan、Hainan Guo 和 Zhongfa Cao（2021）：考虑患者选择行为的门诊预约排程问题：数据建模与仿真优化。《Journal of Combinatorial Optimization》42，第 677–699 页，doi:10.1007/s10878-019-00487-x。[13] Martin Gebser、Roland Kaminski、Benjamin Kaufmann 和 Torsten Schaub（2014）：Clingo= ASP+ control：初步报告。arXiv 预印本 arXiv:1405.3694。参见 https://arxiv.org/abs/1405.3694。[14] Diwakar Gupta 和 Brian Denton（2008）：医疗保健中的预约排程：挑战与机遇。《IIE Transactions》40(9)，第 800–819 页，doi:10.1080/07408170802165880。

[15] Muhamad Haekal 等（2016）：在 SIKASIR RESTful Web服务中使用 JSON Web令牌进行基于令牌的身份验证。载于：2016 年国际信息学与计算会议（ICIC），IEEE，第 175–179 页，doi:10.1109/iac.2016.7905711。[16] Markus Hannebauer 和 Sebastian M¨uller（2001）：医疗预约排程的分布式约束优化。载于：第五届自主代理国际会议论文集，第 139–140 页，doi:10.1145/375735.376026。[17] Stavros Kanias（2023）：一种使用答案集编程进行大规模医疗预约排程的方法。载于：ENIGMA@KR，第 35–42 页。[18] Alex Kuiper、Jeroen de Mast 和 Michel Mandjes（2021）：门诊诊所预约排程问题：一项临床实践多案例研究。《Omega》98，第 102122 页，doi:10.1016/j.omega.2019. 102122。[19] Yong-Hong Kuo、Hari Balasubramanian 和 Yan Chen（2020）：医疗预约超额预订与最优排程：排程效率与服务可及性之间的权衡。《Flexible Services and Manufacturing Journal》32，第 72–101 页，doi:10.1007/s10696-019-09336-z。[20] Vladimir Lifschitz（2008）：什么是答案集编程？载于 Frank van Harmelen、Vladimir Lifschitz 和 Bruce Porter 编：《知识表示手册》，《人工智能基础》第 3 卷，爱思唯尔，第 889–910 页，doi:10.1016/S1574-6526(07)03027-1。[21]Vladimir Lifschitz（2019）：答案集编程。第 3 卷，施普林格 Cham，doi:10.1007/ 978-3-030-24658-7。[22] 意大利劳动、卫生与社会政策部（2009）：CUP 系统：国家指南。https://www.salute.gov.it/imgs/C_17_pubblicazioni_1577_allegato.pdf。访问日期：2025- 04-02。[23] Andrea Monaldini、Alina Vozna 和 Stefania Costantini（2024）：数字健康转型中的 Blueprint Personas。载于：WorkshopHC@AIxIA2024，第 40–49 页。参见 https://ceur-ws.org/Vol-3880/ paper4.pdf。[24] Sam Newman（2021）：构建微服务：设计细粒度系统。” O’Reilly Media, Inc.”。[25] Tiantian Niu、Bingyin Lei、Li Guo、Shu Fang、Qihang Li、Bingrui Gao、Li Yang 和 Kaiye Gao（2023）：系统预约排程优化研究综述。《Axioms》13(1)，第 16 页，doi:10.3390/ axioms13010016。[26] Farid Nouioua 和 Pascal Nicolas（2006）：在自然语言语义的基于推理的方法中使用答案集编程。arXiv 预印本 cs/0607088。参见 https://arxiv.org/abs/cs/ 0607088。[27] Huaxin Qiu、Dujuan Wang、Yanzhang Wang 和 Yunqiang Yin（2019）：检查时间不确定情况下的 MRI预约排程。《Journal of Combinatorial Optimization》37，第 62–82 页，doi:10.1007/ s10878-017-0210-5。