

论文题目: 智能排班问题的数学规划模型与有效求解

学科专业: 计算数学

申请人: 刘嘉

指导教师: 陈志平 教授

## 摘 要

由于其广泛的应用领域与难解性, 对不同形式智能排班问题的研究现已成为运筹学界的一个研究热点。为克服现有模型无法恰当反映人力需求约束实际特征与求解算法效率低等不足, 本文对综合排班问题构建两种不同形式的数学规划模型。

文章第一部分, 针对一类具有复杂约束的大规模排班问题, 运用混合0-1线性规划建模, 智能安排每日工作时间。为了强调模型的通用性, 引入了班次库的概念, 避免了以往模型中难以精确描述休息的困难。建模中, 刻画了大量复杂的约束, 尤其是采用更精细的对人力需求的约束, 并应用于两种不同的背景。另一个与众不同的假定为, 我们预先固定员工的总人数。在求解过程中, 我们引入了一种类似Benders分解的分层方法, 通过引入辅助变量-班次类, 将模型分为较易求解的主问题与从属问题两层。其后, 因为从属问题规模依然较大, 我们根据其特有的块状结构, 使用改进的增广拉格朗日松弛 (ALR) 与对角二次逼近 (DQA) 方法进行分解, 以加速求解。然后, 对强约束条件下长时间难以寻得可行解的问题特别加以讨论, 并提出加入惩罚项的解决方案。

第二部分, 针对确定性规划的种种不足, 构建了概率约束随机规划模型。通过样本平均近似方法和“大M方法”, 我们将概率约束问题转化为大规模整数线性规划问题, 并综合运用了Lagrange松弛法、分解算法与信赖域算法等的思想, 设计出了一种有效的嵌套分解算法。

最后, 通过收敛性分析与数值算例分别说明了本文两个模型的合理性与所设计算法的有效性。

**关键词:** 智能排班; 班次类; Benders分解; 概率约束; 样本平均近似; Lagrange松弛; 信赖域

**论文类型:** 应用研究

**Title:**            **Mathematic Models for Staff Scheduling Problems and  
Their Efficient Solution**  
**Speciality:**   **Computational Mathematics**  
**Applicant:**   **Jia Liu**  
**Supervisor:** **Prof. Zhiping Chen**

## **ABSTRACT**

To aim at the problem of a class of complicated-constrained large-scale tour scheduling, this paper proposes a new solution which is to arrange daily work time intellectually by using the 0-1 MILP model, avoiding the formerly difficulty in accurately describing rest by introducing the concept of shift pool for the sake of highlighting the universality of the model.

We describe lots of complicated constraints, specially the more precise demand constraints, and apply to two situation. A different assumption is that we fix the total employees number previous.

In the process of decomposition, we introduce the layering method similar to Benders, dividing the model into two layers which are easier to solve through auxiliary variable of shift-classes. Because of the large scale of slave problem, we use advanced ALR and DQA based on its peculiar blocky structure to decompose for the acceleration of solving. Then we expressly discuss the problem that it is long hard to find feasible solution in the strong-constrained condition and put forward the solution to which is added penalty term. Lastly, the paper briefly analyzes the astringency of arithmetic with practical instance as well. The instance is carried out in a cycle of a month, precisely satisfying required manpower in any time period everyday. The biggest difference between our model and the current ones is that we do not simplify our established model for the consideration of scales and so on.

Due to their extensive application areas and solution difficulty, different forms of intelligent scheduling problems have now become a research hot in operations research. The current models can not properly reflect practical characteristics of labor requirement constraints and the relevant algorithms lack necessary efficiency. To overcome these shortcomings, two forms of chance constrained programming models are proposed for comprehensive scheduling problems, to flexibly describe the labor requirement constraints. By using the

sample average approximation technique and the Big-M method, we transform the chance constrained programs into large-scale integer programming problems. An efficient nested decomposition algorithm is designed by simultaneously utilizing the Lagrange relaxation technique, the decomposition method and the trust region scheme.

Finally, some numerical results illustrate the reasonability of the proposed models and the efficiency of the proposed algorithm.

**KEYWORDS:** staff scheduling; shift class; Benders decomposition; chance constraint; sample average approximation; Lagrange relaxation; trust region

**TYPE OF DISSERTATION:** Applied Research

# 目 录

1	绪论	1
1.1	智能排班问题简介	1
1.2	研究现状	1
1.2.1	确定性排班模型	2
1.2.2	各类约束条件的建模	4
1.2.3	随机排班模型	5
1.3	本文的主要内容	6
2	新型混合整数规划模型及其有效求解	8
2.1	问题分析与预处理	8
2.2	模型描述	12
2.2.1	参数和数据的定义	12
2.2.2	各类约束的描述方法详述	14
2.2.3	完整的混合整数规划模型	17
2.3	新模型的有效求解	19
2.3.1	基于Benders思想的分层方法	19
2.3.2	内层的增广Lagrange松弛分解方法	25
2.3.3	算法描述	30
2.4	分层方法的改进	31
2.5	数值实验	33
2.5.1	模型规模比较	34
2.5.2	算法效率比较	35
2.6	小结	38
3	两种新型概率约束规划模型与求解	39
3.1	模型介绍	39
3.2	模型求解	40
3.3	数值例子	45
3.4	小结	47
4	结论与展望	49
4.1	结论	49
4.2	展望	50
	致 谢	52

参考文献 .....	53
附录	58
其他	58

# CONTENTS

1.1	智能排班问题简介 .....	1
1.2	研究现状 .....	1
1.2.1	确定性排班模型 .....	2
1.2.2	各类约束条件的建模 .....	4
1.2.3	随机排班模型 .....	5
1.3	本文的主要内容 .....	6
2.1	问题分析与预处理 .....	8
2.2	模型描述 .....	12
2.2.1	参数和数据的定义 .....	12
2.2.2	各类约束的描述方法详述 .....	14
2.2.3	完整的混合整数规划模型 .....	17
2.3	新模型的有效求解 .....	19
2.3.1	基于Benders思想的分层方法 .....	19
2.3.2	内层的增广Lagrange松弛分解方法 .....	25
2.3.3	算法描述 .....	30
2.4	分层方法的改进 .....	31
2.5	数值实验 .....	33
2.5.1	模型规模比较 .....	34
2.5.2	算法效率比较 .....	35
2.6	小结 .....	38
3.1	模型介绍 .....	39
3.2	模型求解 .....	40
3.3	数值例子 .....	45
3.4	小结 .....	47
4.1	结论 .....	49
4.2	展望 .....	50

## 1 绪论

### 1.1 智能排班问题简介

所谓排班问题，即是在给定的一系列要求，如每天的用工人员需求、工作强度要求、劳动保障要求等之下，对给定数量的员工进行合理的工休安排。在过去，排班往往是一个人工统筹的过程，但随着大型呼叫中心等现代服务模式的出现，如此大规模的排班是手工不可能完成的任务。这时，利用计算机找到一种新的智能化方法就成为该问题的解决之道。智能排班旨在利用计算机与现代优化技术解决劳动力管理问题，为客户安排最佳任务，使其在满足未来需求预测的情况下，还能满足运营商的各种人性化需求(包括员工间的均衡)。它在现代服务行业，诸如发电、电信、金融等领域正在被广泛采用。对于劳动密集型行业，其运营成本和运营效率很大程度上取决于如何更有效地安排员工的工作任务，智能排班方法正是可以根据预测估计相应的人力需求，合理安排任务，达到人力资源最有效利用的目的。

智能排班问题的求解可分为三个阶段，第一阶段为预测将来各个时段上的人力需求，第二阶段是根据所预测的话务需求安排合理的员工上班时间，第三阶段为动态地在上班员工中分配当前所接收的业务。本文着重解决第二个比较困难的问题。这一阶段需要通过建立恰当的数学规划模型，并设计有效求解算法来完成。建模时，一方面，我们需要综合考虑客户需求的变化、工作人员的数量、大小班限制、连班限制以及员工对工作时间的人性化要求等约束，使得所建模型更贴近现实，实用性强，这就要求我们的模型必须足够的精细化。另一方面，我们既要顾及模型求解的可行性和难易程度，又要顾及模型的通用性和可拓展性，这就要求我们的模型要尽量简单化。本文中，我们先回顾已有文献的研究结果，继而综合这两方面的因素，分别建立基于确定性线性规划和随机规划的两种智能排班模型。

### 1.2 研究现状

伴随着数学规划，尤其是线性规划在20世纪后期的长足发展，应用数学规划模型、通过计算机来智能求解排班问题已成为一种行而有效的常用方法。之后，再随着整数规划、随机规划等建模方式与求解方法的发展，更多的数学规划模型被应用于智能排班问题的求解中。依据现有文献的特点，我们以下分确定性模型、约束建模、随机模型等三个方面，分别综述已有文献的工作。

### 1.2.1 确定性排班模型

用确定性数学规划求解排班问题的核心在于利用一个线性规划模型或（混合）整数规划模型来描述排班问题，这种方法源于Dantzig<sup>[1]</sup>提出的整数规划模型在道路收费亭问题中的应用。Dantzig模型的基本方法是以资源分配的数目（排班问题中可看作上班的员工数目）为变量，以满足各时段上资源需求，达到资源耗费最少的目标。此后的研究主要基于此模型，在使用更细致的时间划分，提出更人性化的员工上班限制，使员工上班更均衡等方面来做进一步的改进。

现有关于智能排班的研究可分为三个方面：工休排班、班次排班和综合排班<sup>[2]</sup>。工休排班在一个周期内为员工安排工作日和休息日；班次排班仅考虑一天的情况，在这天内为员工安排上班时间和休息时间；综合排班则将这班次和工休排班相结合，既确定一个周期的工作日和休息日，同时为工作日安排上班时间和休息时间。以下分别对这三类文献做一简单的综述。

工休排班重点讨论的是一个周期内（比如一周或一月）如何分配工作与休息日。那么休息日的形式则是讨论的重点，常见的方法有：

1) 经典的每周连续工作5天，连续休息2天的安排方法<sup>[3]</sup>，这种方法最容易处理且应用最广泛，但难以适用于呼叫中心等特殊行业。

2) 另一种灵活的方式为，让员工每周可拥有两天或三天不等的休息天数，并将每周工作日与休息日的各种安排方式固定为模板供员工选择<sup>[4-6]</sup>。

3) 针对不同员工对休息的不同要求，可将员工分类，针对不同类别的员工分配不同类型的休息方式<sup>[7-8]</sup>。

4) 还有一类在固定年度总休息天数的前提下，灵活安排每周休息天数的方法<sup>[9-10]</sup>。

工休排班领域内常见的建模框架与求解算法有整数规划<sup>[9,11]</sup>、动态规划<sup>[6,10]</sup>及各种启发式算法<sup>[4,5]</sup>等。

班次排班在为员工确定上班时间等细节时，通常的做法为先生成候选的可行班次池（potentially large pool of candidates，意为所有班次的集合），然后根据当天各个时段的人力需求来为每个员工选择一个班次<sup>[12]</sup>。这部分研究普遍集中在两个方面，一是如何精确的描述班次并生成可行班次池，二是如何为每个员工选择班次。

关于使用班次的方式，一种为普适的定义，即用每个时段上是否上班作为班次的表示<sup>[13]</sup>，而将可行班次池的生成作为一个单独的部分。另一种方式为自然的想法，直接用上班细节，比如开始上班时间，下班时间，开始休息时间等，在加入简单约束后直接作为班次的描述<sup>[14]</sup>。而这两种方式，都不可避免的难题就是休息的描述。关于休息，已有文献中有以下几种描述方法：

1) 为每个员工安排固定长度，固定次数的休息<sup>[15-17]</sup>，优点在于定义简单操作方便，缺点在于休息的类型太少，有时无法适应特殊环境，比如中间休息间隔长达二到四



个小时的天地班。

2) 将员工分类, 为不同类别员工安排不同类型的休息时间<sup>[18]</sup>, 优点在于从员工的角度去分类更为精细, 但仍难免休息类型太狭窄的缺点。

3) 除加入休息和上班保持连续等少量约束外, 可使用任意班次<sup>[14,19]</sup>, 这样做的优点在于能适应各种复杂环境, 但会带来班次规模较大不易求解的瑕疵。

4) 混合类型, 既考虑休息的普适性, 又通过某些限制条件将休息限定于某些范围之内, 如有考虑天地班等特殊类型<sup>[20]</sup>, 将休息分成数个小段 (fractionable break) <sup>[21]</sup>。这样能结合上述1, 3类描述的优点, 在班次规模与精细程度上加以平衡, 但普适的休息限制条件很难寻找到, 应用时也只能针对实际的问题去人工构造, 这部分依然是研究的焦点。

关于班次的分配, 文献里一般有两种建模思路: 第一种方法以某员工是否上某班次为变量, 以满足每时段人力需求为约束, 以员工人数最省为目标, 构建0-1变量的线性规划<sup>[21]</sup>。第二种方法则分为两个阶段, 第一步沿用Dantzig<sup>[1]</sup>提出的集合覆盖模型确定每个班次所需的数目, 第二步再将员工分配到这些已经确定好数目的班次中去<sup>[14]</sup>。这两种方法的应用范围都很广泛, 因描述班次方式的不同又有所区分。

最后, 综合排班是以上两种模型的结合, 与单纯考虑的工休排班或班次排班模型相较, 其不但考虑到每时段的细节同时又考虑较长的周期, 规模上扩大了很多, 而且, 两方面的约束再交叉考虑会很复杂。已有研究基于规模的考虑通常对每天的细节进行简化或是对天与天之间的约束加以简化, 以控制模型的规模。这样就有三种主流的方法:

一部分文献侧重描述长周期内上班与休息日的刻画, 对于每天上班的细节予以简化, 通常的做法为假设班次仅有数种类型。如文献[23,24]假定员工每天所上班次有早中晚三类班, 在一个周期内排班以拟合三类班的人数需求。这一类模型, 为周期内每天安排一个班次, 同时会着力刻画天与天之间关系, 也就是加入些许约束, 比如连班限制, 即使得某些班次间不能相连<sup>[25]</sup>, 或连续上班限制, 即限制连续上班的最大天数<sup>[26]</sup>等。从另一个角度讲, 这一类模型可看做是工休排班在多班次情况下的推广, 文献[9]将这一类称为多班次模型(several types of activities), 这是相对于一般工休排班中的单一班次模型(one type of activity)而言。

另一部分文献从班次排班出发, 推广到多天的情形。其主要侧重刻画一天内班次的细节, 而对天与天之间的联系则考虑欠佳<sup>[27]</sup>。这一类文献的做法为<sup>[28]</sup>, 将班次的概念推广至一个周期 (比如一周或一月), 而班次的细节也精细到每天的每个小时时间段。这种做法相当于将班次排班中班次的概念加以推广, 将一个长周期看做一个具有多个高峰低谷的一天, 也就是所谓的连续化考虑。

还有一系列的文献, 将上述综合排班模型推广至多区域或多技能的情况。通过额外考虑多个地点间的平衡分配关系<sup>[23]</sup>, 或如何划分可重叠或覆盖的技能组, 使模型满足各种技能的人力需求<sup>[31,32]</sup>。这部分的内容并不在本文的讨论范围之内。

在综合排班中, 常见的算法有Lagrange松弛方法<sup>[26]</sup>, 禁忌搜索<sup>[24]</sup>, Benders分

解<sup>[29]</sup>和基于线性规划的方法<sup>[30]</sup>等。

关于综合排班的更多研究可参考综述文献<sup>[33]</sup>。关于智能排班领域更多的研究可参考文献<sup>[12]</sup>，而在呼叫中心领域排班的综述文章<sup>[34]</sup>中也可以查阅到很多最新的研究文献。

### 1.2.2 各类约束条件的建模

约束条件是建模的重点，不同的约束条件刻画了不同的实际要求，决定了模型的适用范围。将已有文献中出现的各种约束归类，我们总结为以下几类：

1) **宏观统计类约束**。如对每日上班员工的数目，安排上某个班次或班次集合的员工数目，和员工的平均工时等的限制。考虑这一类约束的有文献<sup>[9,22]</sup>等。

2) **每天上班的要求**。包括对员工每天工作中休息和吃饭时间的保障。比如所有员工每隔一段时间就要进行休息，但是为了保证每时刻都有人干活，就要求大家的休息和吃饭时间要分散开，但是又不能因为刻意的分散而让某些员工两次休息之间间隔时间太短。这就需要我们在每天内的排班中加以考虑，可参考文献<sup>[14,20,21]</sup>的做法。

3) **公平性约束**。对于员工来说，总有乐于接受的班次，而将最不受欢迎的班次都安排给个别的员工是不公平的。所以很多公司要求将不同类型的班次均匀地分配给各个员工。这里的描述上也有很多方式，比如限制不同员工之间上各种类型班的比例或次数差。当然，一个更简单的办法是限制一个周期内员工上某个班次集合次数的上下界，就如同文献<sup>[35]</sup>限制员工上某些班次的次数上界的做法。

4) **特殊员工的约束**。对某些特殊的员工，比如孕妇等予以特别的上班限制，这类似于文献<sup>[4,8,26]</sup>将员工分类的做法。

5) **天与天之间的联系**。值得注意的是，我们在工休排班这层面并不仅仅考虑上班与休息的联系，比如最长上班天数，休息天数，休息和工作相连时需满足某些特定的要求。而是拓宽至不同类型班与休息之间的联系，等于是对上班的情形做了一次细化。这里的约束可以再细分为两类，即两天间的连班限制和连续工作或休息天数的限制。连班限制在文献<sup>[26,35]</sup>中有类似考虑，连续工作或休息天数限制则在文献<sup>[3,22]</sup>中也有相近约束。但是，我们这里所要考虑的约束相较已有文献更全面，更普适，从而能够涵盖已有文献中所用约束表达的信息。

6) **对人力需求的拟合**。排班的核心就在于生成能最佳满足各个时段人力需求的班表，我们的模型中将其精细到一个排班周期内每天的每个时段。然后再站在全局的角度使得其总体拟合最佳。这是排班问题的一个核心问题，每个排班的规划都会加以考虑，只是类型和侧重各有不同，目前主要有如下两种描述方式。一种是利用单边不等式或双边不等式，保证各时段所安排人数大于某预定值(如人力需求的预测值)，或限定其处于某个范围内。单边不等式法能有效地保证任意时刻的人力需求都能得以满足，但缺点是不能限定每个时刻应安排人力的上限，这有时会导致人员冗余严重。而双边不等式法在

单边不等式的基础上,对每时刻的人力增加一个上界,从而限制了人员冗余的程度,但如何选取这样的上界是较为困难的问题。如果选取得太小,则可能导致模型无解;若选取得过大,又难以达到将人员冗余平摊的目的。另一种是文献[2,3]等中提出的目标规划方法。这种方法最大的特点在于放松了严格的上、下界要求,将实际安排人力与预测人力的偏离程度作为目标来最小化。

### 1.2.3 随机排班模型

虽然确定性模型有多种人力需求约束的刻画方式,但其确定性需求量的假设严重限制了它的应用范围与应用效果。熟知,智能排班中最核心的内容就是要最优地满足不同的资源需求。然而,未来的资源需求量是事先无法预知的随机量,而预测其最合适的取值又是很困难的<sup>[36]</sup>。鉴于此,一个自然的想法就是直接将资源需求作为随机变量,用随机规划方法来描述需求约束,已有文献中常见的有两种处理方法:一是将人力需求约束不满足程度的期望值作为惩罚加入目标函数,此即随机目标规划法,如文献[37];二是将人力需求约束满足的概率作为关注重点,对满足的概率值加以约束,即概率约束规划法,如文献[38,39]。这两种方法各有其优缺点。随机目标规划从平均的角度去度量人力需求约束的满足情况,它不能有效刻画随机需求分布的尾部特性对约束的影响;概率约束规划将不好的事件发生的概率限定在某个范围之内,这样能够根据实际要求灵活选择概率阈值,但如何设置合适的置信度比较困难,且这类约束没有考虑在约束限定之外的极端情况。

对所建立随机模型的求解,与现有求解概率约束问题的方法一样,通常的做法是从随机变量的分布特性入手,将概率约束转化为确定性约束,再利用求解确定性规划问题的相关算法予以处理。至于如何将概率约束确定化,主要有三种思路:一是在假设正态分布、 $t$ 分布等特殊分布之下,直接将概率约束转化为等价的确定性约束<sup>[40]</sup>;二是利用平均话务率等参数,基于排队论中采用的分布模型来推导出人力需求量服从的分布<sup>[37,38]</sup>,再通过抽样方法将概率约束确定化;三是直接假设随机人力需求服从某个离散分布<sup>[41,42]</sup>,将一个周期内的人力需求模拟为数种不同的需求向量,每组向量代表需求的一组实现,再对每个向量赋予一定的概率(如等概率),使总概率为1。这里,一种典型的方法为文献[43]中的构造法:它将需求分为周期内每日需求模式与每天内每小时需求模式两层,每一层又可根据增长趋势、凹趋势等不同类型构造6种模式,再将这两层结合起来产生离散分布的样本。有了离散分布,就可直接将概率约束转化为通常的确定性约束。

从实际调研我们发现,对于现实中的排班问题,相对于人力需求约束满足程度偏差之大小,客户更为关注的是偏差存在的情形是否过多、能否在大多数情况下使人力需求得到满足、如何恰当描述、如何灵活控制人力需求约束等随机性约束的满足。另外,对于人力需求等约束,硬性的下界约束很容易导致人员冗余严重,如何在某些时段的人力

不足与另一些时段的人员冗余间建立平衡以使人力成本最低，至今还没有好的解决方法。最后，对于现实中复杂的大规模排班问题，现有求解随机模型的算法效率比较差，怎样设计可快速求解的有效算法是亟待解决的难题。

### 1.3 本文的主要内容

分别基于确定性模型与随机规划模型，本文的主要工作可分为两个部分。

在确定性规划模型的应用方面，我们在模型中引入了班次和班次类的概念。一个班次包括上班时间、下班时间、工作时长、中途休息的次数及每次休息的时间和长度等信息，班次类为一些具有相同特征并为相同约束所限制的班次的集合。这样就克服了现有模型难以用线性约束精确描述一日内多次休息，多次上下班的难题，使得整体模型得以线性化。该思想不仅拓宽了模型的应用范围，可解决更加复杂的实际问题，而且使得模型更易于求解。

由于所需求解的排班问题规模很大，且对最终班表的限制要求或约束数目众多，种类各有不同，为了有效处理问题，我们将模型进行分解，根据业务逻辑不同层面约束的特点等将整个系统进行解耦，分解为X层和Z层两个层次的优化问题，并将不同的需求分拆到不同的优化模块里。X层一方面求解班次类层面的人力安排，即每天有哪些员工上班，上哪个班次类的班，另一方面为保证和Z层的相容性，加入了对应班次层面的相关约束。Z层将X层的结果具体到班次，得到每天需要安排那些班次。我们将两层迭代求解，来寻找整体问题的最优解。

为有效求解X层、Z层所对应的大规模混合整数线性规划问题，我们充分利用模型的特性，依耦合性强弱将问题的变量、约束进行分类。通过Lagrange乘子将耦合性强的约束松弛到目标函数中，从而使原问题的整体约束转化为一系列相互独立的子约束块，使问题具有可分性。然而，若仅使用传统的Lagrange松弛法，则会出现分解所得子问题的同构震荡问题。为此，我们在Lagrange函数中引入惩罚项，采用增广Lagrange松弛法，并对原有目标和新增加的每项目标分别赋予不同的权重，通过在求解过程中动态地调整相关权重，保证了可快速收敛到问题的最优解。

由于松弛因子的引入，问题转化为一个双层规划问题。内层通过求解松弛后的规划来更新乘子，外层以乘子为变量寻优。对外层，我们通过改进现有的次梯度算法，以次梯度方向为搜索方向，但用信赖域的方法计算搜索步长，以确定最优乘子。这样就大大提高了求解速度。对内层，由于目标函数的不可微性和不可分性，我们借鉴了Jacobi迭代法的思想，并同时运用与对角二次逼近方法（DQA）类似的绝对值逼近来迭代求解X层。上述一系列求解技巧的运用不仅可将不可微的子问题转化为等价的线性整数规划问题，而且保证了子问题规模相对较小，易于求解。

针对确定性规划模型的不足，本文的第二部分基于随机规划模型就综合排班问题的建模与求解进行研究。为灵活而恰当地描述人力需求约束，我们引入人力不足量的概

念，通过对其设定下界，并选取适当的置信水平，建立两类不同形式的概率约束条件来刻画人力需求约束；为权衡人力不足与人力冗余等问题，我们在建模时引进加权惩罚的思想；不同于现有方法，为有效求解所建立的概率约束规划问题，我们采用随机规划领域中新近提出的样本平均近似方法来将概率约束规划问题转化为确定性整数规划问题，并利用“大M方法”的思想将其线性化；为快速求解所导出的大规模整数规划问题，我们综合运用了Lagrange松弛法、分解算法与信赖域算法等的思想，设计出了一种有效的嵌套分解算法。最后，通过数值算例说明本文模型的合理性与所设计算法的有效性。

全文共分4章：第二章讨论了一种新型混合整数规划模型及其有效求解方法。2.1节就问题展开分析并进行预处理；2.2节详细描述模型；2.3节给出了一套嵌套分解算法；2.4节给出2.3节算法的一种改进方法；2.5节给出实验结果与分析；最后一节为本章小结。第三章讨论了一种新型概率约束模型及其有效求解方法。3.1节就问题展开分析并详细描述模型；3.2节给出求解算法；3.3节给出实验结果与分析；最后一节为本章小结。最后，在第四章中对全文工作进行总结并指出今后的研究方向。

## 2 新型混合整数规划模型及其有效求解

本章分别从问题预处理、建模与模型转化、求解算法设计与实际应用等方面详细论述如何基于确定性规划模型有效处理现实中的复杂综合排班问题。

### 2.1 问题分析与预处理

在确定性模型框架下，我们希望将工休排班和班次排班结合起来，一次排班不但考虑一个较长周期（比如一月内）天与天之间上班的联系，同时，对每一天的上班情况进行细致的描述。在这个过程中，我们以一个较长的周期（比如一月）为一次排班覆盖的长度，在这些天中，每天为每个员工安排适当的班次（这里所指的班次是指符合上班规则的任意班次，而非简单的班次集合）以符合每天各个时段上的人力需求，同时满足天与天之间关系。

因为我们要综合周期内的上班日和休息与每天的上班细节这两方面的信息，在变量定义时，我们需要采用0-1变量的方式，即每天每个员工是否上某个班次为变量。因为以上班人数为变量的方式很难刻画员工天与天之间联系，同时无法描述员工之间公平性的约束。也就是说，我们的模型要构建一个统一的模型，能够一次考虑所有的影响排班的因素，将其容纳于一个模型之中，通过一次求解，排出一个较长周期内精确到每个小时的上班安排。

在建模过程中，我们选择混合整数线性规划来建模，这样做的主要原因有两个：一方面可强化模型的普适性，另一方面是建立如此规模的模型，求解起来就不能通过简单的算法，而一个大规模的数学规划则在数学领域有这广泛的研究，一些规模巨大的问题，通过数学规划中的各种分解算法，也能在有限时间内解决。

对综合排班问题，我们不但要考虑到每时段的细节同时又考虑较长的周期，从而也要考虑上述两方面的约束，同时，两方面的约束再交叉会使得模型更为复杂，这也是就是以往模型刻画综合排班难以刻画大量约束，且较少应用于大规模实例的原因。

本文的建模理念是先不考虑规模带来的求解难度，使用通用的数学模型描述能够广泛应用的普适模型，然后针对建立的数学规划模型，分析其特点，构造分解算法来在可接受的时间内求解。之所以这样做，一方面原因是随着管理的深化，企业的要求越来越精细，而更普适的通用模型能能适应这种不断进步的潮流。另一方面的好处在于，随着数学规划算法的改进也能持续不断的改进模型的求解效果。

在确定模型的框架后，首要的问题就是班次的定义方式。如综述中所述，其有两种定义方式。直观的去看来，以上班的细节作为变量，如开始上班时间和下班时间，休息开

始时间，休息结束时间等就可以描述一天上班的情形了，就如同文献[17,20]的做法。其中分别对每天吃饭和休息的次数有着一到三次不同的假设。这就给我们试图构造泛化的模型制造了困难，因为应用到不同的企业，或许就有不同的企业环境，对休息次数的要求也就不尽相同了。同时，另一个问题为在某些银行或者呼叫中心中的特殊班次，比如说天地班，这是一种一天上两次班，两次之间间隔数个小时的特殊班次。或者是孕妇班，就是为孕妇特别安排的上班时间很短的特殊班次。这处理这一类班次的时候，就很难和用一般班次的那些信息去描述。

为了克服上述问题，我们在模型中将班次予以泛化，将其定义为一个员工一天中任意不同的上班情况。一个班次可以包括但不仅限于上班时间，下班时间，工作时长，中途休息的次数及每次休息的时间和长度等信息。在描述班次的时候，以半小时或15分钟这样的小时间段为单位，将一天分为若干小区间的集合，用一天中每个时间段上是否上班来表示一个班次的信息。

当然，班次并非仅仅通过上述所有时段的上班信息的组合就可生成。由前述的第二类约束可知，由劳动法或者客户的特殊要求，班次有着某些直接的限制。比如所有班次每日上班时间不得超过八小时，中途休息时间不得超过两小时等。所以，所有上班时间的排列组合仅仅是班次的最大可选范围，我们在此范围内，综合考虑与一天内班次相关的所有约束，删除不满足要求的备选班次，挑选出所有满足条件的可行班次。

这里班次生成的过程可以使用一个数学模型来实现，其将一天分为若干小时间段，以每个时段是否工作的0-1变量为自变量，考虑限制班次性质的约束，即1.2.2节中所述的所有第二类约束，其所有可行解即我们要得到的班次库。这里我们可使用数学规划，约束规划等方法来求解。因为这与我们的工作相对独立，故本文不再赘述，可参考文献[44]等。通过如上的预先处理，可生成如表2-1所示的班次库。其中，横坐标代表不同的班次，通过班次的工作时间和休息时间，可以唯一的确定一个班次。纵坐标为时间，表2-1中每项表示某个班次在某个时段是否上班：1表示此时段上班，0表示此时段不上班。通过此班次库，我们可以获得任一班次工作时段的集合（某行取值为1的时段的集合）。同时，也可以确定出某时段上班的班次的集合（某列取值为1的班次的集合）。为了表示方便，以下我们将一天24个小时划分为48个等时长的时段，分别赋予序号1至48。我们使用如表2-1所示的时间与序号间的对应关系。

与现有模型<sup>[14,45]</sup>相比，我们泛化了班次的概念，即不再在排班模型中对班次进行限制，由预处理阶段来确定班次，而生成班次库之后可交由用户维护，根据其具体需求可随时任意添加或删除班次，这样模型就显得更灵活。

从模型角度看，将模型泛化，使得不用细究班次的细节，比如开始时间，下班时间等固定数目的参数，能够扩展模型的适用性，能针对各种应用环境均可套用，尤其是能处理一些现有模型不能描述的具有复杂班次的情况。

从规模上来看，我们将现有模型中描述班次的上、下班时间等<sup>[14]</sup>均用一维变量去描述，规模上并不会增加，反而一些刻画班次内在要求的约束可被省去，从而模型规模亦

更小。

表2-1: 班次库

班次	1	2	3	...
上班时间	07:00-13:00	07:00-14:00	07:00-15:00	...
休息时间	无	11:00-12:00	11:00-12:00	...
工作时长	6	6	7	...
时段1(7:00-7:30)	1	1	1	...
时段2(7:30-8:00)	1	1	1	...
时段3(8:00-8:30)	1	1	1	...
时段4(8:30-9:00)	1	1	1	...
时段5(9:00-9:30)	1	1	1	...
时段6(9:30-10:00)	1	1	1	...
时段7(10:00-10:30)	1	1	1	...
时段8(10:30-11:00)	1	1	1	...
时段9(11:00-11:30)	1	0	0	...
时段10(11:30-12:00)	1	0	0	...
时段11(12:00-12:30)	1	1	1	...
时段12(12:30-13:00)	1	1	1	...
时段13(13:00-13:30)	0	1	1	...
时段14(13:30-14:00)	0	1	1	...
时段15(14:00-14:30)	0	0	1	...
时段16(14:30-15:00)	0	0	1	...
时段17(15:00-15:30)	0	0	0	...
...	...	...	...	...

基于班次构建数学规划模型，其核心思想是员工对班次的选择，故而我们选择每天员工上何班次作为变量。因为有针对员工个人的诸多约束，使用每个员工上第几个班次作为变量是很难用线性的方式描述此类约束。而将变量考虑为每个员工每天是否上某班次这样一组0-1变量就能很轻易的描述了。令

$$y_{ijd} = \begin{cases} 1, & \text{第}d\text{天员工}j\text{上第}i\text{个班次,} \\ 0, & \text{否则,} \end{cases} \quad (2-1)$$

这样的变量定义方式类似于一个指派问题，然而不同于指派问题仅有简单人力的约束，我们希望使用线性等式或不等式来描述1.2.2节中所述的所有约束。下面我们具体分析这些约束的具体表达方法。



因为构建班次库的时候，所有班次已自然地满足了那些针对班次的约束（即第二类约束）。所以我们在建立数学模型的时候就仅需考虑剩下的几类约束。

对于连班限制和连续上班等约束，以要求第一天晚上还在上班的员工第二天上午不能上班这个约束为例，是一个很自然且常见的限制。这种约束是针对一类班次而言的，比如晚上还在上班的班次和上午上班的班次。为了描述的方便，我们将班次依据所受约束的不同分为若干类，也就是划分成若干的子集合，通过子集合来描述约束。我们称这些子集合为班次集合（shift-set）。比如，将此约束中上午上班的班次抽象成早班集合。每个班次集合都是一些具有共同特点（需满足某类约束）的班次的集合。当然，值得注意的是这里的班次之间可能存在相交，比如早班集合和带早茶时间的班次集合。

这些约束都是针对某两个班次的集合而言的。当然，在如上变量定义的基础上，从班次的角度亦能刻画这种限制，那么就是对这两个集合中任意的两个班次都加入一条约束。但是，这样的方式虽然便于构造和理解，但生成出的约束众多且复杂。我们可做以下简化。由员工每天只上一个班次这条性质出发，将员工某天所上班次集合中的班次的情况予以加和，即对所有属于某个班次的集合的 $y_{ijd}$ 求和，得到一个0-1变量（这一点性质由每个员工每天只上一个班组可简单推出），此变量就可以表示员工是否上班次集合中的班了。如此，我们通过对班次不同的约束，将班次划分成类，以班次集合为一个整体来描述约束。而班次集合和班次之间的对应关系我们可通过某班次是否属于某班次集合的一个0-1矩阵来表示，如表2-2所示：

表2-2：班次与班次集合对应表

班次	1	2	3	...
上班时间	07:00-13:00	07:00-14:00	07:00-15:00	...
休息时间	无	11:00-12:00	11:00-12:00	...
工作时长	6	6	7	...
班次集合1	1	1	1	...
班次集合2	1	0	0	...
班次集合3	1	1	0	...
班次集合4	0	0	1	...
班次集合5	0	0	0	...
...	...	...	...	...

表2-2中，横坐标代表不同的班次，纵坐标代表不同的班次集合，表2-2中每项表示某个班次是否属于某个班次集合：1表示属于，0表示不属于。通过此确定的班次表，我们可以获得任一班次所属于的班次集合的集合，和某班次集合所包含的班次的集合。

对于人力需求这一类约束，我们的文章与已有文献有着很大的不同，已有的文献

中,对人力需求一般是通过一个约束来表达的。上述的绝大多数文献,如文献[1],均通过每时刻安排人数不少于预测人力需求的这样一个单边不等式来描述。但这样虽然在每时刻都有充足的人力,但难免会带来人力冗余的问题。这一类模型为规避人力冗余,则将极小化总员工数作为目标。

但是,实际的企业进行排班时,员工的数目很难调整,也就是说,在更多时候,我们将总员工的人数作为确定的常数,只要为每个员工合理的分配班次即可。这么做是由实际的背景所致,员工的招聘需要相当长的周期,而我们排班时每个月所需的员工人数都不相同,而因为劳动法的限制,公司很难因为一个月需求大就大量招聘员工,而因为第二个月需求少而大量辞退员工。

所以,正是由于这种员工人数调整的不及时性,我们将员工总数设为已知常数,在已有员工内部进行一个调整,通过已有员工上班和休息的分配,来平衡不同时间和日期上人力需求。但是,这样做,常常会有人力需求和员工数目相差过多的情况。本文中,我们将人力需求做相应处理,在整个周期内,按照员工人数和平均工作时长,将人力需求等比例放缩,以能保证在需求大的时段安排人数多,在需求少的时段安排人数少,调整时要求所有员工的总工时需求和所有员工的总上班时长相等。具体的调整内容在本章第五节中有进一步的讨论。

至于如何对修正后的人力需求拟合加以约束,我们考虑了等式或不等式的多种方式。但是,实际中的人力需求是很难完全拟合的,因为员工要求连续上班而需求却经常有较大的跳跃,而如果仅用一个单边的不等式,因为总人数已固定,所以会导致无解。所以,我们将对人力需求的拟合程度作为规划的目标,以达到使拟合尽量好的目的。

基于本节的预处理工作,我们就可以用一个混合整数线性规划来描述我们的模型。

## 2.2 模型描述

### 2.2.1 参数和数据的定义

在介绍模型之前,我们先引入如下参数和数据。

- $t: 1 \dots N_t$ , 时段编号。我们将每天分为  $N_t$  个时段,每个时段给予连续标号,全集记做  $T$ 。
- $d: 1 \dots N_d$ , 为天数编号,一个排班周期共有  $N_d$  天,全集记做  $D$ 。
- $j: 1 \dots N_j$ , 为员工编号,全集记做  $J$ 。
- $i: 1 \dots N_i$ , 为班次编号,全集记做  $I$ 。
- $r: 1 \dots N_r$ , 为班次集合编号,全集记做  $R$ 。
- $G_r$ : 第  $r$  类班次集合。

- $J_d$ : 第 $d$ 天上班的员工数。
- $I_t$ : 时段 $t$ 上班的班次的集合。
- $AH_d$ : 所有员工每天上班的平均工作时间。
- $HR_{td}$ : 第 $d$ 天时段 $t$ 的人力需求。
- $AD$ : 一个排班周期内, 一个员工的平均工作天数。
- $l_i$ : 班次 $i$ 的工作时长。
- $WD_j^{\min}(WD_j^{\max})$ : 一个周期内, 员工工作天数的最小(大)值。
- $SR_{r,d}^{\min}(SR_{r,d}^{\max})$ : 第 $d$ 天, 上班次集合 $G_r$ 的员工人数的最小(大)值。
- $TR_{j,r}^{\min}(TR_{j,r}^{\max})$ : 第 $j$ 个员工一个周期内上 $G_r$ 的班次的最小(大)次数。
- $CD_{d,j}^{\min}(CD_{d,j}^{\max})$ : 第 $j$ 个员工若从第 $d$ 天开始上班, 则之后连续上班天数的最小(大)值。
- $RD_{d,j}^{\min}(RD_{d,j}^{\max})$ : 第 $j$ 个员工若从第 $d$ 天开始休息, 则之后连续休息天数的最小(大)值。
- $SD_{r,d,j}^{\min}(SD_{r,d,j}^{\max})$ : 第 $j$ 个员工若从第 $d$ 天开始上班次集合 $G_r$ 中的班, 则之后连续上 $G_r$ 中的班次的天数的最小(大)值。
- $BM_{td}$ : 第 $d$ 天, 第 $t$ 个时段, 人力需求的可容许最大偏差。

对于上述参数与数据, 它们之间需满足以下关系:

1) 每日工作员工数的总和应等于员工平均工作天数乘以总员工数:

$$\sum_{d \in D} J_d = N_j \times AD.$$

2) 总员工数与人力需求量之间应满足以下关系式:

$$2AH_d \times N_j \times AD = \sum_{t \in T} \sum_{d \in D} HR_{td}.$$

若不满足, 我们应当根据实际情况调整总员工数或人力需求量, 调整的细节可参考本章第五节。

## 2.2.2 各类约束的描述方法详述

基于(2-1)中变量的定义和上节中的参数和数据，我们可逐类地用线性等式或不等式来描述各种约束。

### 1) 统计类约束

这一类约束从所有员工总体的角度来加以约束。我们模型中包括对每天上班员工数，上各个班次集合的员工数和所有员工的平均工时等的约束。

- 要求第 $d$ 天上班员工数为 $J_d$ ，我们用 $\Delta_1$ 表示所有此类型约束所约束的天数的集合：

$$\sum_{i \in I} \sum_{j \in J} y_{ijd} = J_d, \quad \forall d \in \Delta_1.$$

- 要求第 $d$ 天上班次集合 $G_r$ 的员工人数不小于 $SR_{r,d}^{\min}$ ，同时不大于 $SR_{r,d}^{\max}$ ，我们用 $\Delta_2$ 表示所有此类型约束所约束的班次集合和天数的二元集合：

$$\sum_{i \in G_r} \sum_{j \in J} y_{ijd} \geq SR_{r,d}^{\min}, \quad \forall (r, d) \in \Delta_2,$$

$$\sum_{i \in G_r} \sum_{j \in J} y_{ijd} \leq SR_{r,d}^{\max}, \quad \forall (r, d) \in \Delta_2.$$

- 要求第 $d$ 天所有上班员工的平均工作时长为 $AH_d$ ，我们用 $\Delta_3$ 表示所有此类型约束所约束的天数的集合：

$$\sum_{i \in I} l_i \left( \sum_{j \in J} y_{ijd} \right) = \lceil I_d * AH_d \rceil, \quad \forall d \in \Delta_3.$$

上式中，因为平均工作时长为法律所规定的固定值，其乘以总人数并不一定为整数，而所需等于的所有员工的总工作时长为整数，故对平均工时与人数的乘积取整。式中用符号 $\lceil \cdot \rceil$ 表示取整。

### 2) 公平性约束

这一类约束限定员工之间工作差异不能过大，我们通过每个员工周期内工作或上某个班次的次数的界限来予以限制。

- 要求任一员工每天最多上一个班次：

$$\sum_{i \in I} y_{ijd} \leq 1, \quad \forall d \in D, \forall j \in J.$$

- 要求员工 $j$ 在一个周期内工作天数不小于 $WD_j^{\min}$ ，同时不大于 $WD_j^{\max}$ ，我们用 $\Delta_4$ 表示所有此类型约束所约束的员工的集合：

$$\sum_{d \in D} \sum_{j \in J} y_{ijd} \geq WD_j^{\min}, \quad \forall j \in \Delta_4,$$

$$\sum_{d \in D} \sum_{j \in J} y_{ijd} \leq WD_j^{max}, \quad \forall j \in \Delta_4.$$

- 要求员工 $j$ ，在周期内上 $G_r$ 的次数不小于 $TR_{r,j}^{min}$ ，同时不大于 $TR_{r,j}^{max}$ ，我们用 $\Delta_5$ 表示所有此类型约束所约束的班次集合和员工的二元集合：

$$\sum_{d \in D} \sum_{i \in G_r} y_{ijd} \geq TR_{r,j}^{min}, \quad \forall (r, j) \in \Delta_5,$$

$$\sum_{d \in D} \sum_{i \in G_r} y_{ijd} \leq TR_{r,j}^{max}, \quad \forall (r, j) \in \Delta_5.$$

### 3) 特殊员工的约束

这一类约束对于某些特殊员工，限制其只能上某些班次，或不能上某些班次。

- 要求第 $d$ 天员工 $j$ 只能安排 $G_r$ 中的班次，我们用 $\Delta_6$ 表示所有此类型约束所约束的班次集合、天数和员工的三元集合：

$$\sum_{i \notin G_r} y_{ijd} = 0, \quad \forall (r, d, j) \in \Delta_6.$$

- 要求第 $d$ 天员工 $j$ 不能安排 $G_r$ 中的班次，我们用 $\Delta_7$ 表示所有此类型约束所约束的班次集合、天数和员工的三元集合：

$$\sum_{i \in G_r} y_{ijd} = 0, \quad \forall (r, d, j) \in \Delta_7.$$

### 4) 连班约束

此为天与天之间约束中的一种，限制两天间班次的相连情况。

- 若第 $d$ 天员工 $j$ 上 $G_{r_1}$ 中的班，则要求其第 $\tilde{d}$ 天不能上 $G_{r_2}$ 中的班，我们用 $\Delta_8$ 表示所有此类型约束所约束的班次集合 $G_{r_1}, G_{r_2}$ ，天数 $d, \tilde{d}$ 和员工 $j$ 的五元集合：

$$\sum_{i \in G_{r_1}} y_{ijd} + \sum_{i \in G_{r_2}} y_{ij\tilde{d}} \leq 1, \quad \forall (r_1, r_2, d, \tilde{d}, j) \in \Delta_8.$$

- 若第 $d$ 天员工 $j$ 上 $G_{r_1}$ 中的班，则要求其第 $\tilde{d}$ 天必须能上 $G_{r_2}$ 中的班，我们用 $\Delta_9$ 表示所有此类型约束所约束的班次集合 $G_{r_1}, G_{r_2}$ ，天数 $d, \tilde{d}$ 和员工 $j$ 的五元集合：

$$\sum_{i \in G_{r_1}} y_{ijd} + \sum_{i \notin G_{r_2}} y_{ij\tilde{d}} \leq 1, \quad \forall (r_1, r_2, d, \tilde{d}, j) \in \Delta_9.$$

- 若第 $d$ 天员工 $j$ 休息，则要求其第 $\tilde{d}$ 天不能上 $G_r$ 中的班，我们用 $\Delta_{10}$ 表示所有此类型约束所约束的班次集合 $G_r$ ，天数 $d, \tilde{d}$ 和员工 $j$ 的四元集合：

$$\sum_{i \in G_r} y_{ij\tilde{d}} \leq \sum_{i \in I} y_{ijd}, \quad \forall (r, d, \tilde{d}, j) \in \Delta_{10}.$$

- 若第 $d$ 天员工 $j$ 上 $G_r$ 中的班，则要求其第 $\tilde{d}$ 天必须休息，我们用 $\Delta_{11}$ 表示所有此类型约束所约束的班次集合 $G_r$ ，天数 $d, \tilde{d}$ 和员工 $j$ 的四元集合：

$$\sum_{i \in G_r} y_{ijd} + \sum_{i \in I} y_{ij\tilde{d}} \leq 1, \quad \forall (r, d, \tilde{d}, j) \in \Delta_{11}.$$

### 5) 连续上班或休息约束

此为天与天之间约束的另一种，限制连续工作或休息的长度。

- 若员工 $j$ 从第 $d$ 天开始上班，则要求其连续上班的天数不小于 $CD_{d,j}^{\min}$ ，且不大于 $CD_{d,j}^{\max}$ ，我们用 $\Delta_{12}$ 表示所有此类型约束所约束的天数和员工的二元集合：

$$\sum_{d'=d}^{d+CD_{d,j}^{\min}} \sum_{i \in I} y_{ijd'} \geq CD_{d,j}^{\min} \sum_{i \in I} y_{ijd}, \quad \forall (d, j) \in \Delta_{12},$$

$$\sum_{d'=d}^{d+CD_{d,j}^{\max}} \sum_{i \in I} y_{ijd'} \leq CD_{d,j}^{\max} \sum_{i \in I} y_{ijd}, \quad \forall (d, j) \in \Delta_{12}.$$

- 若员工 $j$ 从第 $d$ 天开始休息，则要求其连续休息的天数不小于 $RD_{d,j}^{\min}$ ，且不大于 $RD_{d,j}^{\max}$ ，我们用 $\Delta_{13}$ 表示所有此类型约束所约束的天数和员工的二元集合：

$$\sum_{d'=d}^{d+RD_{d,j}^{\min}} (1 - \sum_{i \in I} y_{ijd'}) \geq RD_{d,j}^{\min} (1 - \sum_{i \in I} y_{ijd}), \quad \forall (d, j) \in \Delta_{13},$$

$$\sum_{d'=d}^{d+RD_{d,j}^{\max}} (1 - \sum_{i \in I} y_{ijd'}) \leq RD_{d,j}^{\max} (1 - \sum_{i \in I} y_{ijd}), \quad \forall (d, j) \in \Delta_{13}.$$

- 若员工 $j$ 从第 $d$ 天开始上 $G_r$ 的班，则要求其连续上 $G_r$ 班的天数不小于 $SD_{r,d,j}^{\min}$ ，且不大于 $SD_{r,d,j}^{\max}$ ，我们用 $\Delta_{14}$ 表示所有此类型约束所约束的班次集合、天数和员工的三元集合：

$$\sum_{d'=d}^{d+SD_{r,d,j}^{\min}} \sum_{i \in G_r} y_{ijd'} \geq SD_{r,d,j}^{\min} \sum_{i \in G_r} y_{ijd}, \quad \forall (r, d, j) \in \Delta_{14},$$

$$\sum_{d'=d}^{d+SD_{r,d,j}^{\max}} \sum_{i \in G_r} y_{ijd'} \leq SD_{r,d,j}^{\max} \sum_{i \in G_r} y_{ijd}, \quad \forall (r, d, j) \in \Delta_{14}.$$

### 6) 人力需求约束

这类约束描述排班的目的，即最佳地覆盖人力的需求。

- 对于一些用户要求更苛刻的时段，如第 $d$ 天第 $t$ 个时段，根据其需求，要求安排人力与预测人力需求的绝对偏差最大不得超过 $BM_{td}$ ，我们用 $\Delta_{15}$ 表示所有此类型约束所约束的时段和天数的二元集合：

$$\left| \sum_{i \in I_t} \sum_{j \in J} y_{ijd} - HR_{td} \right| \leq BM_{td}, \quad \forall (t, d) \in \Delta_{15}.$$

### 7) 目标函数

我们使用在每天每时段上，安排工作人数与需求的人数之间绝对偏差的总和

$$\sum_{d \in D} \sum_{t \in T} \omega_{td} \left| \sum_{i \in I_t} \sum_{j \in J} y_{ijd} - HR_{td} \right|$$

作为优化目标，寻求其最小值，即对预测人力需求的最优拟合。这里，用户可通过调节 $\omega_{td}$ 来对不同关心程度的时段加以不同的权重。部分时段的较大权重体现出用户更关心这些时段上实际安排人力与期望人力之间的接近程度。

### 2.2.3 完整的混合整数规划模型

将上述各个约束条件的数学描述与目标函数集中到一起，就可得到一个混合整数规划问题。但是，因目标函数和人力需求约束中均含有不可微的绝对值项，不易求解。为此，我们先通过引入辅助变量 $a_{td}^1, a_{td}^2$ 对这些约束进行等价变形，满足如下条件：

$$\sum_{i \in I_t} \sum_{j \in J} y_{ijd} - HR_{td} = a_{td}^1 - a_{td}^2, \quad \forall t \in T, \forall d \in D,$$

$$a_{td}^1, a_{td}^2 \geq 0, \quad \forall t \in T, d \in D.$$

于是，目标函数可转化为如下的线性表达式：

$$\sum_{d \in D} \sum_{t \in T} \omega_{td} (a_{td}^1 + a_{td}^2),$$

并可将人力需求约束转化为如下的线性表达式：

$$a_{td}^1 + a_{td}^2 \leq BM_{td}, \quad \forall (t, d) \in \Delta_{15}.$$

综上所述，我们的模型可被描述为以下的混合整数规划问题：

$$\min \sum_{d \in D} \sum_{t \in T} \omega_{td} (a_{td}^1 + a_{td}^2) \quad (2-2)$$

$$\text{s.t.} \quad \sum_{i \in I} \sum_{j \in J} y_{ijd} = J_d, \quad \forall d \in \Delta_1, \quad (2-3)$$

$$\sum_{i \in I} l_k \left( \sum_{j \in J} y_{ijd} \right) = [I_d * AH_D], \quad \forall d \in \Delta_3, \quad (2-4)$$

$$\sum_{i \in I} y_{ijd} \leq 1, \quad \forall d \in D, \forall j \in J, \quad (2-5)$$

$$\sum_{d \in D} \sum_{j \in J} y_{ijd} \geq WD_j^{min}, \quad \forall j \in \Delta_4, \quad (2-6)$$

$$\sum_{d \in D} \sum_{j \in J} y_{ijd} \leq WD_j^{max}, \quad \forall j \in \Delta_4, \quad (2-7)$$

$$\sum_{i \in G_r} \sum_{j \in J} y_{ijd} \geq SR_{r,d}^{min}, \quad \forall (r, d) \in \Delta_2, \quad (2-8)$$

$$\sum_{i \in G_r} \sum_{j \in J} y_{ijd} \leq SR_{r,d}^{max}, \quad \forall (r, d) \in \Delta_2, \quad (2-9)$$

$$\sum_{d \in D} \sum_{i \in G_r} y_{ijd} \geq TR_{j,r}^{min}, \quad \forall (r, j) \in \Delta_5, \quad (2-10)$$

$$\sum_{d \in D} \sum_{i \in G_r} y_{ijd} \leq TR_{j,r}^{max}, \quad \forall (r, j) \in \Delta_5, \quad (2-11)$$

$$\sum_{i \notin G_r} y_{ijd} = 0, \quad \forall (r, d, j) \in \Delta_6, \quad (2-12)$$

$$\sum_{i \in G_r} y_{ijd} = 0, \quad \forall (r, d, j) \in \Delta_7, \quad (2-13)$$

$$\sum_{i \in G_{r_1}} y_{ijd} + \sum_{i \in G_{r_2}} y_{Ij\tilde{d}} \leq 1, \quad \forall (r_1, r_2, d, \tilde{d}, j) \in \Delta_8, \quad (2-14)$$

$$\sum_{i \in G_{r_1}} y_{ijd} + \sum_{i \notin G_{r_2}} y_{ij\tilde{d}} \leq 1, \quad \forall (r_1, r_2, d, \tilde{d}, j) \in \Delta_9, \quad (2-15)$$

$$\sum_{i \in G_r} y_{ijd} + \sum_{i \in I} y_{Ij\tilde{d}} \leq 1, \quad \forall (r, d, \tilde{d}, j) \in \Delta_{10}, \quad (2-16)$$

$$\sum_{i \in G_r} y_{ijd} \leq \sum_{i \in I} y_{ij\tilde{d}}, \quad \forall (r, d, \tilde{d}, j) \in \Delta_{11}, \quad (2-17)$$

$$\sum_{d'=d}^{d+CD_{d,j}^{min}} \sum_{i \in I} y_{ijd'} \geq CD_{d,j}^{min} \sum_{i \in I} y_{ijd}, \quad \forall (d, j) \in \Delta_{12}, \quad (2-18)$$

$$\sum_{d'=d}^{d+CD_{d,j}^{max}} \sum_{i \in I} y_{ijd'} \leq CD_{d,j}^{max} \sum_{i \in I} y_{ijd}, \quad \forall (d, j) \in \Delta_{12}, \quad (2-19)$$

$$\sum_{d'=d}^{d+CD_{r,d,j}^{min}} \sum_{i \in G_r} y_{ijd'} \geq CD_{r,d,j}^{min} \sum_{i \in G_r} y_{ijd}, \quad \forall (r, d, j) \in \Delta_{13}, \quad (2-20)$$

$$\sum_{d'=d}^{d+CD_{r,d,j}^{max}} \sum_{i \in G_r} y_{ijd'} \leq CD_{r,d,j}^{max} \sum_{i \in G_r} y_{ijd}, \quad \forall (r, d, j) \in \Delta_{13}, \quad (2-21)$$

$$\sum_{d'=d}^{d+CD_{d,j}^{min}} (1 - \sum_{i \in I} y_{ijd'}) \geq RD_{d,j}^{min} (1 - \sum_{i \in I} y_{ijd}), \quad \forall (d, j) \in \Delta_{14}, \quad (2-22)$$



$$\sum_{d'=d}^{d+CD_{d,j}^{max}} (1 - \sum_{i \in I} y_{ijd'}) \leq RD_{d,j}^{max} (1 - \sum_{i \in I} y_{ijd}), \quad \forall (d, j) \in \Delta_{14}, \quad (2-23)$$

$$\sum_{i \in I_t} \sum_{j \in J} y_{ijd} - HR_{td} = a_{td}^1 - a_{td}^2, \quad \forall d \in D, \forall t \in T, \quad (2-24)$$

$$a_{td}^1 + a_{td}^2 \leq BM_{td}, \quad \forall (t, d) \in \Delta_{15}, \quad (2-25)$$

$$a_{td}^1, a_{td}^2 \geq 0, \quad \forall t \in T, d \in D, \quad (2-26)$$

$$y_{ijd} \in \{0, 1\}. \quad (2-27)$$

以下我们将上述模型记作原始问题P1。

与文献[22,24]相比，我们对员工每天直接分配班次，而不是笼统地分配一个班次集合，这样我们对班次的描述更精细，对每各时段上的人力需求满足更好。这些文献对每天班次的划分很粗糙，绝大多数仅从早中晚班这样大的框架来定义，来安排满足每个班次所需的人数，这很难在安排班次人数的时候准确把握每时刻的人力需求。一般需要在其后再加入一个班次排班的阶段来具体确定班次的上班时间。但往往在第二阶段，因为每天每个班次的人数业已固定，调整的空间就很小了，未必能找到最优解。这种分两个阶段的方法，可以看做后述我们建立的通用模型的一种简化后的特殊形式。而我们的模型将这两个阶段融为一体，能更好的契合人力需求，避免以往模型到第二阶段无法排班或效果很差的情况。

另一方面，与文献[27,28]相比较，我们并不将天与天之间独立考虑，而在两天之间加入限定条件。相较于连续化考虑，至少找到了一种有效刻画天与天之间关系的方法，在描述工作日与休息日上就有着更人性化的考虑，更明显的一点好处为这样可以避免员工接连上同种班次，或者第一天下班很晚而第二天上班很早的情况。

## 2.3 新模型的有效求解

对于如上建立的混合整数线性规划，自然可以通过现有的分枝定界法或割平面法去求解。但是，所应用的实例多为一些较大规模的公司或者运营系统，对于大规模的问题，简单的算法就未必能有效的在可接受的时间内求解了。假设一个排班系统有100个员工，一天内的班次数目为500个，并以一个月为周期，这种情况下，模型约有 $O(N_d \cdot N_j \cdot N_i)$ ，即 $1.5 \times 10^6$ 个变量。这么大规模的MILP，分枝定界或者割平面算法就无法在有效的在可接受的时间内求解了。鉴于此，为有效求解问题P1，我们将充分利用各种现代优化技术，导出可有效求解规划问题P1的分解算法。

### 2.3.1 基于Benders思想的分层方法

从问题的结构出发，我们针对其自身特性构造新型算法。不难看到，问题P1中线性约束对应的矩阵是极稀疏的，而进一步观察可发现，所有的变量在模型中均

以 $\sum_{i \in I} y_{ijd}$ ,  $\sum_{j \in J} y_{ijd}$ ,  $\sum_{i \in G_r} y_{ijd}$ ,  $a_{td}^1$ ,  $a_{td}^2$ 的形式出现：约束(2-3),(2-4),(2-6)-(2-9),(2-24)衡量有多少员工，所以使用的变量均为 $\sum_{j \in J} y_{ijd}$ ,  $a_{td}^1$ ,  $a_{td}^2$ ；表示人力需求的约束(2-25,2-26)仅用 $a_{td}^1$ ,  $a_{td}^2$ 就可表示；连班或连续上班约束(2-5),(2-18)-(2-21)仅到某天某员工是否上班的信息，用变量 $\sum_{i \in I} y_{ijd}$ 就可以描述；而用到某天某员工是否上某个班次集合信息的约束(2-10)-(2-15),(2-22),(2-23)用变量 $\sum_{i \in G_r} y_{ijd}$ 就可以描述；这两者的综合(2-16,2-17)可用 $\sum_{i \in I} y_{ijd}$ ,  $\sum_{i \in G_r} y_{ijd}$ 的线性组合表示。

各种约束中， $\sum_{j \in J} y_{ijd}$ 表示的约束可以对天分解，而 $\sum_{i \in I} y_{ijd}$ 和 $\sum_{i \in G_r} y_{ijd}$ 表示的约束可以对员工分解。那么，如果我们将模型分为针对这两类变量的两层，那么每层都是一个具有良好块结构的子规划，就可以用分解算法来加速求解。这就让我们联想到了Benders分解的方法，但是，经典的Benders分解的方法并不能够直接套用，其原因在于如果将模型分为用 $\sum_{j \in J} y_{ijd}$ 描述与用 $\sum_{i \in I} y_{ijd}$ ,  $\sum_{i \in G_r} y_{ijd}$ 描述两层，两层间变量的关系无法线性表示。虽然其中 $\sum_{j \in J} y_{ijd}$ 与 $\sum_{i \in G_r} y_{ijd}$ 之间有线性等式

$$\sum_{j \in J} \left( \sum_{i \in G_r} y_{ijd} \right) = \sum_{i \in G_r} \left( \sum_{j \in J} y_{ijd} \right)$$

可以完整的表示两者之间的联系，但是 $\sum_{i \in I} y_{ijd}$ 与 $\sum_{j \in J} y_{ijd}$ ,  $\sum_{i \in G_r} y_{ijd}$ 之间的关系不能用线性的等式来描述。从实际背景上来看，班次集合可能相交，比如说，早晨10点前上班的班次集合和中午12点吃饭的班次集合，之间相交但不相同。这样，对班次集合的简单加和不一定等于每天上班的全集，这就是问题的所在。如此，我们必须重新定义班次集合，提取所有班次集合相交的部分，将之构造为互不相交的划分，并称划分后互不相交的班次的集合为班次类。具体地，我们可以通过如下的算法来生成班次类：

**算法2-1:** 对给定的班次集合 $G_1, G_2, \dots, G_{N_r}$ 。

步骤0: 令 $n = 1, I = 1, B_1 = G_1, G(I) = \{B_1\}$ 。

步骤1: 对每个 $B_i, i = 1, \dots, I$ ，生成班次类如下：

步骤1.1: 若 $B_i \cap G_{n+1} = \emptyset$ ，则转至步骤1.3。

步骤1.2: 若 $B_i \cap G_{n+1} \neq \emptyset$ ，则令 $B_{I+1} = B_i \cap G_{n+1}, B_i = B_i - B_i \cap G_{n+1}, I = I + 1$ , 和 $G(I) = G(I - 1) \cup \{B_I\}$ 。

步骤1.3: 令 $i = i + 1$ 。若 $i = I$ ，则转至步骤2；否则，转至步骤1.1。

步骤2: 若 $B_i \neq \emptyset$ ，则令 $B_{I+1} = B_i, I = I + 1$ ，以及 $G(I) = G(I - 1) \cup \{B_I\}$ 。

步骤3: 令 $n = n + 1$ ，若 $n < N_r$ ，则转至步骤1；否则，终止并输出 $G(I)$ 。

与班次集合类似的，我们重新生成班次类与班次的对应关系表。我们用某班次是否属于某班次类的一个0-1矩阵来表示班次类和班次之间的对应关系，如表2-3所示。其中，横坐标代表不同的班次，纵坐标为班次类，表2-3中每项表示某个班次在是否属于某个班次类：1表示属于，0表示不属于。通过此确定的班次表，我们可以获得任一班次所属于的班次类的集合，以及某班次类所包含的班次的集合。

表2-3:班次类和班次之间的对应关系表

班次	1	2	3	...
上班时间	07:00-13:00	07:00-14:00	07:00-15:00	...
休息时间	无	11:00-12:00	11:00-12:00	...
工作时长	6	6	7	...
班次类1	1	0	0	...
班次类2	0	0	1	...
班次类3	0	1	0	...
班次类4	0	0	0	...
班次类5	0	0	0	...
...	...	...	...	...

如上算法输出的 $G(I)$ 即所有班次类的集合,  $I$ 为班次类个数。为了记号统一, 我们令 $N_s$ 等于算法2-1输出的 $I$ , 表示班次类总数。将所有班次类重新排列, 按照标号 $s$ , 从1到 $N_s$ , 分别记做 $B_s, s = 1, \dots, N_s$ 。那么, 任一班次集合可以表示成一系列互不相交的班次类 $B_{r_1}, \dots, B_{r_{I_r}}$ 的并。有了上述处理,  $\sum_{i \in G_r} y_{ijd}$ 就可通过 $\sum_{i \in B_s} y_{ijd}$ 的线性组合来表示如下:

$$\sum_{i \in G_r} y_{ijd} = \sum_{B_s \subseteq G_r} \sum_{i \in B_s} y_{ijd}.$$

关于 $\sum_{i \in G_r} y_{ijd}$ 的线性约束(2-10)-(2-15),(2-22),(2-23), 它们也均可通过 $\sum_{i \in B_s} y_{ijd}$ 来线性描述。

同时,  $\sum_{i \in I} y_{ijd}$ 也可表示为 $\sum_{i \in B_s} y_{ijd}$ 的如下线性组合:

$$\sum_{i \in I} y_{ijd} = \sum_{s \in S} \sum_{i \in B_s} y_{ijd}.$$

关于 $\sum_{i \in G_r} y_{ijd}$ 的线性约束(2-5),(2-18)-(2-21), 也就可以通过 $\sum_{i \in B_s} y_{ijd}$ 来线性描述。而约束(2-16),(2-17)由 $\sum_{i \in G_r} y_{ijd}$ 和 $\sum_{i \in G_r} y_{ijd}$ 共同组成, 从而亦可通过 $\sum_{i \in B_s} y_{ijd}$ 来线性描述。

我们将上述这些能用 $\sum_{i \in B_s} y_{ijd}$ 表示的约束(2-5),(2-10)-(2-23)归做一类, 将用 $\sum_{j \in J} y_{ijd}, a_{td}^1, a_{td}^2$ 表示的约束(2-3),(2-4),(2-6)-(2-9),(2-24)-(2-26)归为另一类, 此时可看到, 因为 $B_s$ 之间互不相交, 所以,  $\sum_{j \in B_s} y_{ijd}$ 与 $\sum_{i \in G_r} y_{ijd}$ 之间就可用线性等式

$$\sum_{j \in J} \left( \sum_{i \in B_s} y_{ijd} \right) = \sum_{i \in B_s} \left( \sum_{j \in J} y_{ijd} \right)$$

完整地表示两者之间的联系。为后文表达简便, 我们将变量分别重新定义为 $x$ 和 $z$ :

$$z_{id} = \sum_{j \in J} y_{ijd}, \quad \forall d \in D, \quad \forall i \in I,$$

$$\begin{aligned}
 z_{(N_i+t)d} &= a_{td}^1, \quad \forall d \in D, \quad \forall t \in T, \\
 z_{(N_i+N_t+t)d} &= a_{td}^2, \quad \forall d \in D, \quad \forall t \in T, \\
 x_{sjd} &= \sum_{i \in B_s} y_{ijd}, \quad \forall s \in S, \quad \forall d \in D \quad \forall j \in J.
 \end{aligned}$$

同时，我们定义 $x$ 和 $z$ 的向量形式：

$$\bar{z}_d = (z_{1d}, z_{2d}, \dots, z_{(N_i+2N_t)d})^T \in R^{N_i+2N_t},$$

$$\bar{x}_j = (x_{1j1}, x_{1j2}, \dots, x_{1jN_d}, \dots, x_{ijd}, \dots, x_{N_sj1}, x_{N_sj2}, \dots, x_{N_sjN_d})^T \in R^{N_d \times N_s}.$$

则模型(2-2)-(2-27)可用向量与矩阵的形式重新描述为：

$$\min \sum_{d \in D} \sum_{t \in T} \omega_{td} (z_{(N_i+t)d} + z_{(N_i+N_t+t)d}) \quad (2-28)$$

$$\text{s.t.} \quad \sum_{j \in J} x_{sjd} = \sum_{i \in B_s} z_{id}, \quad \forall d \in D, \quad \forall s \in S, \quad (2-29)$$

$$P2 \quad \Phi'_d \bar{z}_d = \phi'_d, \quad \forall d \in D, \quad (2-30)$$

$$\Upsilon'_j \bar{x}_j = v'_j, \quad \forall j \in J, \quad (2-31)$$

$$x_{sjd} \in \{0, 1\}, \quad z_{id} \in Z. \quad (2-32)$$

其中 $\Phi'_d \bar{z}_d = \phi'_d$ 表示约束(2-3),(2-4),(2-6)-(2-9),(2-24)-(2-26)，而 $\Upsilon'_j \bar{x}_j = v'_j$ 表示约束(2-5),(2-10)-(2-23)，我们将此问题记为 $P2$ 。

至此，我们通过重新划分的班次类，将模型的变量分为两大部分。从实际问题看，这两部分变量 $x$ 和 $z$ 是站在两个不同的维度来描述的，这使得我们考虑是否能够将模型分为内外两层，在内层先满足求解一部分变量，满足其自身所对应的约束，然后再将解作为常量带入原问题，若求的的另一部分变量的解并不能使原问题可行或最优，则在重新去求解开始先固定的那部分变量，当然此时要利用另一部分变量的解作为已知，如此循环交替，直到求出问题最优解。这样的分层方法与经典的Benders分解思想类似，即将原问题分离成两组决策变量 $z$ 和 $x$ 分别对应的两层。

考虑所有含 $z$ 的约束和目标的 $z$ 层，我们可导出下列以 $z$ 为变量的规划问题为：

$$\min \sum_{d \in D} \sum_{t \in T} \omega_{td} (z_{(N_i+t)d} + z_{(N_i+N_t+t)d}) \quad (2-33)$$

$$P(z) \quad \text{s.t.} \quad \sum_{j \in J} x_{sjd} = \sum_{i \in B_s} z_{id}, \quad \forall d \in D, \quad \forall s \in S, \quad (2-34)$$

$$\Phi'_d \bar{z}_d = \phi'_d, \quad \forall d \in D. \quad (2-35)$$

$$z_{id} \in Z. \quad (2-36)$$

我们称 $P(z)$ 为“主问题”，其目标为每一天子目标的线性和，而每项约束仅包括 $d$ 天的变量，所以主问题可被分解成如下 $N_d$ 个相互独立的子规划快速而并行地求解：

$$\begin{aligned}
 & \min \sum_{t \in T} \omega_{td} (z_{(N_i+t)d} + z_{(N_i+N_t+t)d}) \\
 P(z_d) \quad & \text{s.t.} \quad \sum_{j \in J} x_{sjd} = \sum_{i \in B_s} z_{id}, \quad \forall s \in S, \\
 & \Phi'_d \bar{z}_{id} = \phi'_d. \\
 & z_{id} \in Z.
 \end{aligned} \tag{2-37}$$

同理，考虑所有含 $x$ 的约束，我们可得到以 $x$ 为变量的如下方程组：

$$P(x) \begin{cases} \sum_{j \in J} x_{sjd} = \sum_{i \in B_s} z_{id}, \quad \forall d \in D, \forall s \in S, & (2-38) \\ \Upsilon'_j \bar{x}_j = v'_j, \quad \forall j \in J, & (2-39) \\ x \in \{0, 1\}. & (2-40) \end{cases}$$

我们称之为“从属问题”。

如上，我们将原问题分解成更易于处理的两层，通过交替求解这两层，逐步逼近原问题的最优解。具体过程如下：

首先，我们求解不含表示 $z$ 和 $x$ 关系的约束的主问题 $P(z^0)$ ，

$$\begin{aligned}
 P(z^{(0)}) \quad & \min \sum_{d \in D} \sum_{t \in T} \omega_{td} (z_{(N_i+t)d} + z_{(N_i+N_t+t)d}) \\
 & \text{s.t.} \quad \Phi'_d \bar{z}_{id} = \phi'_d, \quad \forall d \in D
 \end{aligned}$$

此规划可以分解成 $N_d$ 个相互独立的子规划来并行求解。我们将其解记做 $z$ 的初值 $z^{(1)}$ 。

然后，从此初值出发，将其代入“从属问题” $P(x)$ 中求解，并得到一可行解 $x$ ，但是，注意到在 $x$ 层的求解时，因为原始问题的变量 $z$ 的取值业已固定，说明约束(2-37)要比原本限制在 $z$ 上的约束(2-3),(2-4),(2-6)-(2-9),(2-24)-(2-26)要更严格，这相较原问题等于加强了约束，则 $x$ 层未必一定存在可行解。所以，我们在循环迭代时，当 $x$ 层不能求得可行解的情况下，在求得近似可行解时亦终止，并称为近似解。

由于 $x$ 层有藕合约束存在，不能像 $P(z)$ 般分解为 $N_d$ 个子规划并行求解。为此，我们特提出一套有效的分解算法，将其转化为一系列的小子问题来求解，这部分的内容详见2.3.2节。

在得到“从属问题”的解 $x$ 后，再反代回 $P(z)$ 形成具有新约束的“主问题”，从而更新 $z$ 值，之后再如此交替地求解两阶段问题，直到得出最优解或达到可接受范围内的解为止。在交替求解的过程中，我们用 $n$ 来指代循环的次数，用 $z^{(n)}$ 指代第 $n$ 次迭代 $z$ 层的解，用 $x^{(n)}$ 指代第 $n$ 次迭代 $x$ 层的解。具体算法流程如下：

**算法2-2:**

步骤0: 求解 $P(z^{(0)})$ , 记其最优解为 $z^{(1)}$ , 并令 $n = 1$ 。

步骤1: 求解 $P(x)$ , 其中 $z = z^{(n)}$ , 得到其可行解或近似可行解 $x^{(n)}$ 。

步骤2: 若步骤1中得到的为可行解, 且满足终止条件, 则终止并输出 $x^{(n)}, z^{(n)}$ ; 否则, 转入步骤3。

步骤3: 令 $n = n + 1$ , 求解 $P(z)$ , 其中 $x = x^{(n)}$ , 得到其最优解 $z^{(n)}$ , 并转入步骤1。

分析以上算法, 关于 $x$ 层是否存在可行解, 我们有如下结论:

**定理2-1:** 分层算法2-2中,  $P(x^{(n)})$ 存在可行解的必要条件是 $V_{P2} < V_{P(z^{(n-1)})}$ , 其中 $V_{P2}$ 为原始问题的最优值,  $V_{P(z^{(n-1)})}$ 是 $P(z^{(n-1)})$ 的最优值。

**证明:** 用反证法, 假设 $P(x^{(n)})$ 在第 $n$ 步迭代中求得一个可行解, 且满足 $V_{P2} > V_{P(z^{(n-1)})}$ 。因为 $z^{(n-1)}$ 满足限制(2-30),  $P(x^{(n)})$ 的可行解 $x^{(n)}$ 满足(2-29,2-31), 故而 $x^{(n)}$ 和 $z^{(n-1)}$ 必定是 $P2$ 的可行解。

然而 $x^{(n)}$ 和 $z^{(n-1)}$ 对应的最优值等于 $V_{P(z^{(n-1)})}$ , 且由假设知 $V_{P2} \leq V_{P(z^{(n-1)})}$ , 这就表明 $V_{P2}$ 不是 $P2$ 的最优解。这和定理假设就相矛盾, 所以我们可以肯定 $V_{P2} \leq V_{P(z^{(n-1)})}$ 是 $P(x^{(n)})$ 存在可行解的必要条件。  $\square$

对于第一次迭代时,  $x$ 层的可行性与原问题的最优性之间有如下结论:

**定理2-2:** 在算法2-2中, 如果 $P(x^{(1)})$ 有可行解, 则 $x^{(1)}$ 和 $z^{(0)}$ 就是 $P2$ 整个问题的最优解。

**证明:** 我们比较 $P2$ 和 $P(z^{(0)})$ 。两者有相同的目标函数, 但 $P(z^{(0)})$ 中的约束只是在 $P2$ 中约束的一部分(2-30)。所以 $P(z^{(0)})$ 的最优值必定不大于 $P2$ 的最优值, 即

$$V_{P2} \geq V_{P(z^{(0)})}.$$

同时, 如果 $P(x^{(1)})$ 存在可行解, 由定理2-1我们可以知如下的不等式成立:

$$V_{P2} \leq V_{P(z^{(0)})}.$$

从上面两个不等式中, 我们可以得出:

$$V_{P2} = V_{P(z^{(0)})}. \quad \square$$

我们通过这两个定理可以得出算法2-2的一些特性。首先, 定理2-2给出了一种很快(仅仅迭代了一次)就寻找到总体最优解的情形, 而且从实际中看(可参考第2-5节的实证), 这种迅速求的最优解的情形在简单情况, 即宽松的约束环境和较宽广的解域下, 很经常出现。一方面的原因是约束的放松使得最优解的个数增加, 另一方面是使得在从属问题中最优解的寻取对约束(2-37)更加不敏感, 从而很容易求出可行解。

这种能在第一步求得解的情况, 可被看做已有文献<sup>[14]</sup>中一大类通用建模方式的等价。其第一步沿用文献[1]提出的集合覆盖模型确定每个班次所需的数目, 第二步再将员

工分配到这些已经确定好数目的班次中去<sup>[14]</sup>。这时，其第一步的集合覆盖模型仅考虑和各个班次需要人数相关的约束，而这些约束都是能被包括于 $P(z^{(0)})$ 中的，那么其第一阶段就能被看做我们模型的 $P(z^{(0)})$ ，其第二部分再分配班次到员工的过程相当于一个减缩版的 $P(x^{(1)})$ ，这样子看这一类文献的算法框架可被等价的看做我们分层迭代算法的第一次循环的过程，实际很多情况下第一步就可找到可行解，从定理2-2就可知，算法第一步就直接找到总体的最优解。

在一些复杂情况下， $P(x^{(1)})$ 找不到可行解时，我们的算法会继续下一次迭代来修正约束(2-37)，而已有文献中的算法<sup>[14]</sup>如果找不到可行解就已经结束了。有些文献也注意到了会存在无解的情况，但并没有给出新的方法来修正。这也就是我们的算法较已有文献算法更优的原因。

另外，我们的分层算法在处理某些特殊环境，比如强约束环境下的问题时，也会遇到很困难的问题，尤其是第一次循环中 $P(x^{(1)})$ 找不到可行解时，因为 $P(z^{(1)})$ 相较 $P(z^{(0)})$ ，约束(2-34)的加入导致目标(2-33)骤然上升，而这个上升的幅度在某些环境下很剧烈，这就导致了算法不鲁棒。我们将在本章第4节中专门讨论，并提出一种针对这种情况的改进算法，并在本章第五节中将之与本节的算法加以比较。

### 2.3.2 内层的增广Lagrange松弛分解方法

对于 $x$ 层的求解，因为此线性方程组定义在离散空间上，导致其为一NP难的问题，从而不能在多项式时间内求解，常见的方法有转化为数学规划和约束规划求解两类。应用约束规划(constraint programming)建模并用分枝的策略求解类似排班问题的做法可参考文献[46,47]。

对于我们待求解的问题 $P(x)$ ，其具有良好的特性，我们若将此线性方程组写成矩阵形式，我们可观察出其具有很明显的块状结构。如图2-1所示。

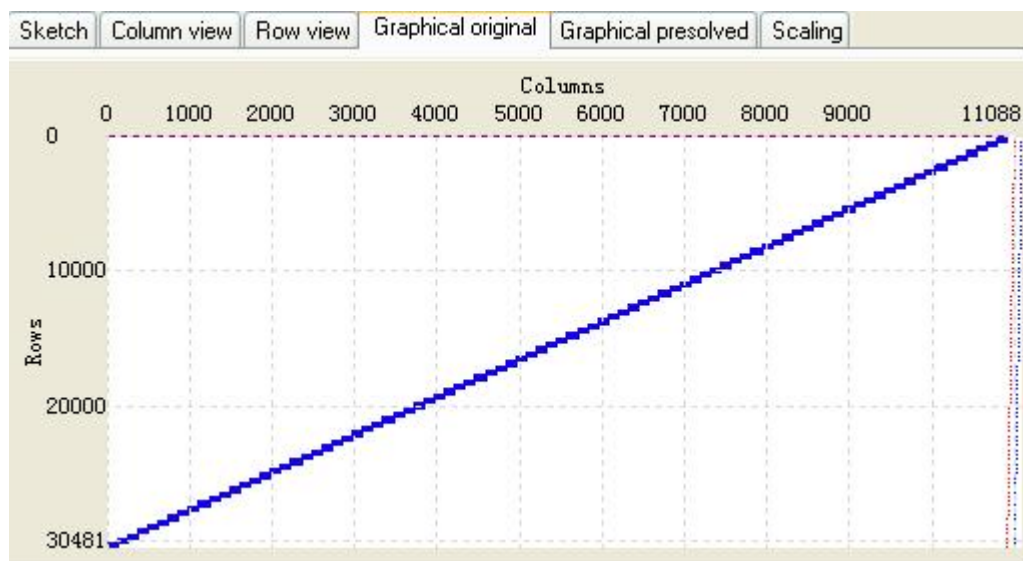


图2-1: 约束矩阵结构示意图

从图2-1中可看出，其除了最上层的一行约束（实际对应我们模型中的约束（2-37））外，均可对 $j = 1, \dots, N_j$ 分成 $N_j$ 块。进而观察每个约束矩阵块，其结构都是相似的，如图2-2所示。尤其是在 $j$ 较大及约束较强时，耦合约束占总约束的比例较小，在这时如我们将耦合约束松弛为目标，则问题的约束可分解为 $J$ 个独立的部分，这样再套用分解算法就能将大规模的原问题化为一系列的小规模子问题，从而加速求解。

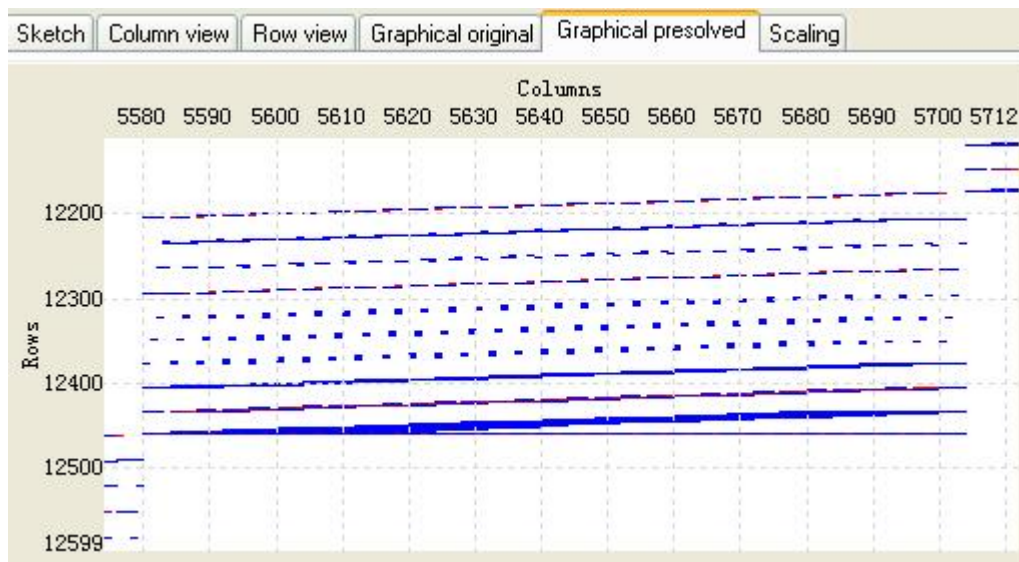


图2-2: 约束矩阵块状结构的内部结构

关于松弛法，我们首先联想到最常用的Lagrange松弛法，但经过如下的分析后，转而使用增广Lagrange松弛方法。

Lagrange松弛法通过引入Lagrange乘子将耦合（不可分）约束转化为目标函数，因为松弛因子的引入，将原问题转化为一个两层优化问题。内层根据当前步乘子确定其对应的最优值：

$$V_{LR}(\lambda) = \min \sum_{d \in D} \sum_{i \in I} \left[ \lambda_{sd} \left( \sum_{j \in J} x_{sjd} - \sum_{i \in B_s} z_{id} \right) \right] \quad (2-41)$$

$$\text{s.t. } \Upsilon'_j \bar{x}_j = v'_j, \quad \forall j \in J. \quad (2-42)$$

外层对乘子寻求最优：

$$\max_{\lambda \geq 0} V_{LR}(\lambda). \quad (2-43)$$

松弛法的优点在于通过一系列较容易求解的规划来逼近原本很难求解的问题，能大幅度提高求解效率，较快时间内就能得到偏差较小的解。但其缺点也很明显：

首先，从理论上讲，拉格朗日松弛法仅能在内层规划连续的情况下，有收敛性保障<sup>[48]</sup>。而离散情况对应的规划本身就是NP难的，所以应用松弛法就没有相类似的收敛性保证。



其次，在某些情况下，经典的松弛法会出现同解震荡的问题<sup>[49]</sup>，导致松弛法不收敛。比如本例中，若所有 $x$ 层规划对每个员工 $j$ 所对应的约束均相同，那么每个员工所对应的解也相应的均相同，这就导致 $\sum_{j \in J} x_{sjd}$ 中的 $x_{sjd}$ 会同时取0或1，则 $\sum_{j \in J} x_{sjd}$ 就只能等于0或 $N_j$ ，那么这就将与约束 $\sum_{j \in J} x_{sjd} = I_d$ 产生冲突。从实际背景的角度去看，对所有员工分解后，每个员工上班所要求满足的约束均相同，那么其解亦相同，这就导致所有员工一同上班一同下班，自然不能满足将人力分散开的约束了。

再次，拉格朗日松弛法中方向和步长一旦选取的不合理或会导致解波动很大，而对离散问题，因其不可微，其自身就不存在梯度或次梯度方向，而用LP松弛问题（是指仅去掉变量为整数的约束后的规划问题）的次梯度方向作为近似，虽然能在初始较快的下降，但是当靠最优解很近时，反而因为方向或步长的些许偏差，就导致下降速度很慢甚至震荡不前。在不引入启发式的算法的情况下，仅凭松弛法很难在短时间内寻找到全局最优。

针对经典的拉格朗日乘子法的上述缺陷，特别是考虑到子问题同构震荡的问题，我们将罚函数的思想引入到拉格朗日函数中来，在(2-41)中松弛项后加入惩罚项，即采用增广拉格朗日松弛法来加速其收敛<sup>[66,67]</sup>。

我们在耦合约束 $\sum_{j \in J} x_{sjd} = \sum_{i \in B_s} z_{id}$ 不能得到满足时加以惩罚，使用偏离程度的绝对和作为惩罚的度量，即惩罚项。如此，乘子法的内层(2-41)-(2-42)转化为如下的规划问题：

$$V_{LR}(\lambda) = \min \sum_{d \in D} \sum_{i \in I} \left[ \lambda_{sd} \left( \sum_{j \in J} x_{sjd} - \sum_{i \in B_s} z_{id} \right) + \beta_{sd} \left| \sum_{j \in J} x_{sjd} - \sum_{i \in B_s} z_{id} \right| \right] \quad (2-44)$$

$$\text{s.t. } \Upsilon'_j \bar{x}_j = v'_j, \quad \forall j \in J. \quad (2-45)$$

我们将其记作 $P'(x)$ 。

我们可以看到增广Lagrange松弛实为经典的Lagrange松弛法和罚函数法的一个结合，它既可以看做是为了避免经典Lagrange松弛法带来的子问题同构而加入惩罚项作为修正，也可以看做是为了改进罚函数法收敛速度慢且罚因子无限增加的缺陷而引入Lagrange松弛项。

值得注意的是，我们的方法与通常文献<sup>[48]</sup>中所使用的增广Lagrange松弛方法不同，我们采用绝对偏差和<sup>[65]</sup>，而非二次函数来刻画惩罚项。虽然二次惩罚项所具有的光滑特性极具吸引力，使得能应用研究较成熟的梯度算法或一些二次规划的特有算法，但出于以下四个方面的考虑，本文还是选择绝对值罚函数：

1) 对于大规模问题，二次罚函数作为惩罚项增长迅速，而当固定乘子的调整策略确定后，对于数值计算中的小扰动（即变量出现较小的偏差），二次项会放大这个变化，导致解变差，这种现象尤其是在乘子的调整范围较小时更容易发生。而且因为持续的放大效应，甚至会导致惩罚项溢出或过大以至于淹没原优化目标。

2) 二次惩罚项不是精确惩罚函数，而绝对偏差和作为一种精确罚函数在乘子较大时

更容易逼近最优值。

3) 绝对值惩罚项虽然为一种非光滑函数，但是求解时可以通过引入辅助变量转化为线性形式，虽然引入了少量变量，但在同级别规模目标为线性的情况较非线性的情况会容易求解许多。

4) 从引入罚函数的实际效果来看，在有一定偏差存在的情况下，绝对值罚函数可代表用户更希望满足绝大多数时段，或总体的绝对偏离最小，即便在某些时段有颇大的偏差；而二次罚函数会导致有较多的时段不能被满足，但是偏差会比较均匀，是一种对较大偏差表示出更厌恶的选择。

下边我们分别就乘子法的内、外层分别讨论。

对于外层定义在 $\lambda$ 连续空间上的极大化问题(2-43)，我们可采用一般的线性搜索方法。其最核心的内容即在于如何选取迭代的方向和步长。我们将目标函数对 $\lambda_{sd}$ 的导数作为方向，即：

$$d^k = \left( \sum_{j \in J} x_{1j1}^k - \sum_{i \in B_1} z_{i1}, \sum_{j \in J} x_{1j2}^k - \sum_{i \in B_1} z_{i2}, \dots, \sum_{j \in J} x_{1jN_d}^k - \sum_{i \in B_1} z_{iN_d}, \right. \\ \left. \dots, \sum_{j \in J} x_{Sj1}^k - \sum_{i \in B_S} z_{i1}, \sum_{j \in J} x_{Sj2}^k - \sum_{i \in B_S} z_{i2}, \dots, \sum_{j \in J} x_{SjN_d}^k - \sum_{i \in B_S} z_{iN_d} \right)$$

实际上，对连续线性规划的原始Lagrange松弛方法，若将约束全部松弛到目标，则此方向为精确的梯度方向，若部分松弛则为次梯度方向<sup>[50]</sup>。但对增广Lagrange问题，此方向就不一定为次梯度方向。而在本文的模型中，因原始问题为离散问题，所以也就无所谓梯度或次梯度一说，所以我们近似使用此方向，在文献中这种方法也被称为替代次梯度（Surrogate subgradient）<sup>[51]</sup>。文献[54]中对于一般线性整数规划问题，肯定了替代次梯度方向在Lagrange松弛框架的有效性，认为其能够降低对偶问题求解过程的计算量。

因为我们选取的方向仅为次梯度方向的近似，那么满足经典次梯度算法步长规则的步长并不一定适用我们的问题。这里我们采用在非光滑优化中较常用的信赖域方法。信赖域方法的思想是依照每次迭代改进的效果来确定下一步的步长，若上一步迭代效果好，说明此方向很好，则加大步长以求更快下降；若上一步迭代效果差，则缩小步长，以免在不好的方向上前进过多<sup>[50,52]</sup>。应用到本文模型中，我们每一步实时观察此次迭代和上次迭代之间改进的程度，并选用总的目标值的下降作为改进程度的度量：当本次迭代目标值减去上次迭代目标值后小于 $\theta_L$ ，则增大步长；大于 $\theta_U$ ，则减小步长；介于之间时，保持步长不变。

至此，我们可构建如下乘子法的外层循环算法：

### 算法2-3:

步骤1: 初始化乘子 $\lambda_{sd}^1$ 为零，令 $k = 1$ ， $s_k = 1$ ，并选取合适的 $\theta_L$ ， $\theta_U$ ， $\alpha_L$ 和 $\alpha_U$ 。

步骤2: 对给定的 $\lambda_{sd}^k$ ，求解 $P'(x)$ ，并记其最优值为 $V_x^k$ ，其最优解为 $x^{k+1}$ 。

步骤3: 若 $V_x^k - V_x^{k-1} < \theta_L$ ，则令 $s^k = \alpha_L \cdot s^{k-1}$ ；若 $V_x^k - V_x^{k-1} > \theta_U$ ，则令 $s^k = \alpha_U \cdot s^{k-1}$ ；否则，令 $s^k = s^{k-1}$ 。

步骤4: 更新 $\lambda_{sd}^k$ 为 $\lambda^k + s^k \cdot d^k$ , 其中

$$d^k = \left( \sum_{j \in J} x_{1j1}^k - \sum_{i \in B_1} z_{i1}, \sum_{j \in J} x_{1j2}^k - \sum_{i \in B_1} z_{i2}, \dots, \sum_{j \in J} x_{1jN_d}^k - \sum_{i \in B_1} z_{iN_d}, \right. \\ \left. \dots, \sum_{j \in J} x_{Sj1}^k - \sum_{i \in B_S} z_{i1}, \sum_{j \in J} x_{Sj2}^k - \sum_{i \in B_S} z_{i2}, \dots, \sum_{j \in J} x_{SjN_d}^k - \sum_{i \in B_S} z_{iN_d} \right).$$

步骤5: 若 $\forall d \in D, \forall s \in S$ , 都有 $\sum_{j \in J} x_{sjd} = \sum_{i \in B_s} z_{id}$ 成立, 则终止, 并输出 $x^{k+1}$ 为 $P(x)$ 的可行解; 若 $k \geq K$ 亦终止, 并输出 $x^{k+1}$ 为 $P(x)$ 的近似解; 否则, 令 $k = k + 1$ , 并转入步骤2。

对于乘子法内层需要求解的规划 $P'(x)$ , 由于目标函数中存在绝对值惩罚项, 从而不能等价地分解成若干独立子问题。然而其规模对于已有的分支定界或割平面等求解整数规划的算法依然过大, 为了更有效的求解这一类大规模问题, 常常人为构造一系列较易求解的子问题来逼近求解原问题。一种常用的方法为交错方向法 (alternating direction method), 可参考文献[53]; 另一种为我们使用的分解算法为对角二次逼近法, 这种方法用一系列小的子规划的循环求解逼近原问题的解, 而子规划维度降低的幅度很大, 从而能够有效的加速求解, 可参考早期引入思想的文献[54], 在随机规划中加以应用的文献[55], 系统分析在LP松弛问题中此方法收敛特性的文献[48], 以及在一些工程领域加以应用的文献[49,56]。

事实上, 上述序贯求解的思想源于数值求解方程组的Jacobi迭代法和Guass-Seidel迭代法。沿着这两种迭代框架, 我们的可分逼近也可分为Guass-Seidel模式和Jacobi模式。其中沿用Jacobi模式的算法也被称为非线性Jacobi算法 (nonlinear Jacobi Algorithm) [57]。

我们将这种可分逼近的方法应用于我们模型的 $x$ 层, 并选择了收敛较快的Guass-Seidel模式。这些序贯的子规划均是在 $P'(x)$ 的基础上, 固定一部分变量, 从而降低规模来序列求解。具体的方法为, 我们顺序求解从第1个子问题到第 $N_j$ 个问题, 在求解第 $j$ 个子问题 $P'(x)$ 时, 假定第 $j$ 个员工的排班策略为变量, 其余员工的均为常数, 即求解如下规划:

$$\min \sum_{d \in D} \sum_{i \in I} [\lambda_{sd} (\sum_{j' \neq j} x_{sj'd}^* - x_{sjd} - \sum_{i \in B_s} z_{id}^*) + \beta_{sd} |\sum_{j' \neq j} x_{sj'd}^* - x_{sjd} - \sum_{i \in B_s} z_{id}^*|] \\ \text{s.t. } \Upsilon_j \bar{x}_j = v_j'.$$

我们将第 $j$ 个子规划记为 $P'(x_j)$ 。假设我们在求解第 $m$ 次迭代的第 $j$ 个子问题, 则上式中,  $x_{sj'd}^*$ 代表第 $m-1$ 次迭代的第 $j+1$ 至 $J$ 个子问题和第 $m$ 次迭代的第1个至第 $j-1$ 个子问题的解,  $z_{id}^*$ 代表第 $m$ 次迭代 $z$ 层的解。如此顺序的求解, 并依次循环, 直到在 $M^L$ 步内无改进则终止。

注意到, 在求解每个子规划时, 约束均为线性等式或不等式, 目标均为绝对偏差的和。这时我们可通过类似第2-3节中处理规划目标中绝对值项的方法, 引入辅助变量和约

束将非光滑的绝对值转化为线性表达式，则此非光滑问题可表示为如下形式的混合整数线性规划：

$$\begin{aligned}
 & \min \sum_{d \in D} \sum_{i \in I} [\lambda_{sd} (\sum_{j' \neq j} x_{sj'd}^* - \sum_{i \in B_s} z_{id}^* - x_{sjd}) + \beta_{sd} (\kappa_{sjd}^1 + \kappa_{sjd}^2)] \\
 P'(x_j) \quad & \text{s.t.} \quad \Upsilon_j \bar{x}_j = v_j' \\
 & \sum_{j' \neq j} x_{sj'd}^* - \sum_{i \in B_s} z_{id}^* - x_{sjd} = \kappa_{sjd}^1 - \kappa_{sjd}^2, \\
 & \kappa_{sjd}^1, \kappa_{sjd}^2 \geq 0.
 \end{aligned}$$

我们记 $V_{P'(x_j)}$ 为 $P'(x_j)$ 的最优值， $x_j$ 为其最优解。 $m^L$ 为连续不改进时的最大尝试次数，当连续 $m^L$ 次不改进时终止。 $\epsilon$ 为判断不改进时目标值变化的上界。 $m^{\max}$ 为 $m$ 最大的取值，当 $m$ 超过 $m^{\max}$ 时终止。我们可用以下算法求解 $P(x^k)$ ：

#### 算法2-4：

步骤0: 令 $x^* = x^{k-1}$ ， $m = 0$ ；

步骤1: 令 $j = 1$ 。

步骤1.1: 求解 $P'(x_j)$ ，并用其最优解 $x_j$ 更新 $x^*$ 。

步骤1.2: 若 $j = N_j$ ，则转至步骤2；否则，令 $j = j + 1$ ，并转入步骤1.1。

步骤2: 若 $m \geq m^{\max}$ ，且最优值在 $m^L$ 步内无改进（亦即 $\max\{V_{P'(x_j)}^{k,m}, V_{P'(x_j)}^{k,m-1}, \dots, V_{P'(x_j)}^{k,m-m^L}\} - \min\{V_{P'(x_j)}^{k,m}, V_{P'(x_j)}^{k,m-1}, \dots, V_{P'(x_j)}^{k,m-m^L}\} < \epsilon$ ），则终止并输出 $x^k$ 为 $x^*$ ；否则，令 $m = m + 1$ ，并转入步骤1。

### 2.3.3 算法描述

本节给出了一整套求解 $P(y)$ 的方法，其中算法2-1为预处理、引入初值的算法，算法2-2为主循环，亦即最外层循环，其内嵌套算法2-3以求解其中的 $P(x)$ ，而在算法2-3的内部，为了更快的求解 $P'(x)$ ，可以使用算法2-4。

总结以上各个子算法，我们给出如下整体的算法框架：

#### 算法2-5：

步骤0: 初始化所以参数，并导入所以数据。调用算法2-1生成所以班次类。

步骤1: 求解 $P(z^{(0)})$ ，记起最优解为 $z^{(1)}$ ，并令 $n = 1$ 。

步骤2: 将 $z = z^{(n)}$ 带入 $P(x)$ ，并用如下算法求解 $P(x)$ ：

步骤2.1: 初始化乘子 $\lambda_{sd}^1$ 为零，令 $k = 1$ ， $s_k = 1$ ，并选取合适的 $\theta_L$ ， $\theta_U$ ， $\alpha_L$ 和 $\alpha_U$ 。

步骤2.2: 对给定的 $\lambda_{sd}^n$ ，调用算法2-4求解 $P'(x)$ ，并记其最优值为 $V_x^k$ ，其最优解为 $x^{k+1}$ 。

步骤2.3: 若 $V_x^k - V_x^{k-1} < \theta_L$ ，则令 $s^k = \alpha_L \cdot s^{k-1}$ ；若 $V_x^k - V_x^{k-1} > \theta_U$ ，则令 $s^k = \alpha_U \cdot s^{k-1}$ ；否则，令 $s^k = s^{k-1}$ 。

步骤2.4: 更新 $\lambda_{sd}^k$ 为 $\lambda^k + s^k \cdot d^k$ , 其中

$$d^k = \left( \sum_{j \in J} x_{1j1}^k - \sum_{i \in B_1} z_{i1}, \sum_{j \in J} x_{1j2}^k - \sum_{i \in B_1} z_{i2}, \dots, \sum_{j \in J} x_{1jN_d}^k - \sum_{i \in B_1} z_{iN_d}, \right. \\ \left. \dots, \sum_{j \in J} x_{Sj1}^k - \sum_{i \in B_S} z_{i1}, \sum_{j \in J} x_{Sj2}^k - \sum_{i \in B_S} z_{i2}, \dots, \sum_{j \in J} x_{SjN_d}^k - \sum_{i \in B_S} z_{iN_d} \right).$$

步骤2.5: 若 $k \geq K$ , 或 $\forall d \in D, \forall s \in S$ , 都有 $\sum_{j \in J} x_{sjd} = \sum_{i \in B_s} z_{id}$ 成立, 则令 $x^{(n)} = x^{(n,k)}$ , 并转入步骤3; 否则, 令 $k = k + 1$ , 并转入步骤2.2.

步骤3: 若步骤2.5得到可行解, 且满足终止条件, 则终止程序并输出 $x^{(n)}$ ,  $z^{(n)}$ ; 否则转入步骤4.

步骤4: 令 $n = n + 1$ , 将 $x = x^{(n)}$ 带入 $P(z)$ , 并求解得最优解 $z^{(n)}$ , 然后转入步骤2.

## 2.4 分层方法的改进

上节建立了针对 $P(y)$ 这一类大规模问题的一整套求解算法, 在实践中发现, 其运用在约束条件非常强的情况时效果并不令人满意, 随着约束的变严格, 算法的稳定性会变差。以我们实际应用中所遇到的一个问题为例, 此例中在某天下午19:00至24:00之间, 人力需求有一个突然的上升。但是, 假定公司同时规定, 每天19:00至24:00之间不能由新的员工上班, 且无员工休息, 这时, 19:00-24:00之间, 如论如何安排, 后一时段上班的人数都必然比前一时段人数少, 这就使得人力需求曲线和安排的人力之间必然存在差异, 人力需求就不可能被完全拟合, 即 $V_{P1} > 0$ 。同时我们假定, 仅有此约束导致人力需求不能被完全拟合, 那么不考虑这个约束的初始问题 $P(z^{(0)})$ 甚则可是人力需求的偏差为0, 即 $V_{P(z^{(0)})} = 0$ , 由定理2-1就可知 $P(x^{(1)})$ 必不可行。

上例只是给出较为明显的导致 $V_{P1} > 0$ 情况的一例, 当我们的模型应用在一些严格约束的环境下, 比如说由大量晚班和天地班的医院或者呼叫中心时,  $V_{P1}$ 会较大, 因为约束的限制, 使得人力需求偏差的下界大于0。这个时候, 如果仍然使用如上的循环方式, 则会在开始的很多步迭代内, 因为约束并没有被完全考虑, 会有 $V_{P(z^{(n-1)})} < V_{P1}$ 的最优值均会小于 $V_{P1}$ 的情况, 这样每次都会在 $x$ 层循环中达到循环上限或循环到解几乎没有改动为止。这样的缓慢升高最优值且不能终止(因为 $x$ 层始终不会有可行解)的过程, 会耗费大量的时间。

对于上述例子, 如果仍然使用如上的循环方式, 在迭代的第一步, 因为约束并没有被完全考虑, 会有 $V_{P(z^{(n-1)})} < V_{P1}$ 的情况。这会在第一次求解 $x$ 层时, 一旦引入约束(2-37)后, 因为 $z$ 值的不合理, 使得 $x$ 不一定朝着我们期待的方向前进, 从而无可行解, 但如此的 $x$ 却在带入第二次循环时, 因为约束(2-34)的引入使得 $z$ 层目标上升, 而且上升的幅度很大, 这就导致算法不稳定。

从另一个角度考虑, 即第一次迭代( $P(z^{(0)})$ )不加强约束(2-34), 而从第二次迭代( $P(z)$ )开始又加入了强约束(2-34), 这就使得可行域骤然缩小, 而这个约束是由上一步

迭代的近似 $x$ 值生成的，并不一定恰当，所以有可能使得约束加的严格了，导致将最优解排除在外。虽然这通过主问题和从属问题的不断循环能逐渐调整，但目标值的突然上升会导致收敛的过程剧烈震荡，而这不但会用大量时间来调整，也降低了算法的稳定性。

我们改进的思路就是不直接加入硬性约束，即将 $P(z)$ 中的约束(2-34)，而是将它作为惩罚项放到目标里，用对这个约束的不满足程度来作为惩罚，使得这个约束逐渐的被满足，通过与原有目标之间的平衡，使目标值逐渐上升，直到达到 $V_{P1}$ 。具体的改进方法如下：

在开始阶段， $P(x^{(1)})$ 找不到可行解时，松弛约束(2-37)不能满足且偏差很大时，较缓慢地加速问题 $P(z)$ 目标值(2-33)的上升，即不一次加入严格的约束(2-34)使其一部满足而牺牲目标的效果，也要考虑算法效率，在离解域较远时有很好的办法来加速其靠近。我们注意，越接近可行域，约束(2-34或2-37)对应的绝对偏差 $|\sum_{j \in J} x_{sjd} - \sum_{i \in B_s} z_{id}|$ 也相应越小，所以此偏差就能很好的表示当前解离可行域的远近，所以我们选取此偏差来表示当前解与解域的距离。在用函数表示时，我们对各时间段上的偏差予以加权求和。为了与表示对不同时段的重视程度，每项的权重均可取不同，我们使用式(2-44)中的 $\beta$ 作为惩罚项中每项的权值，记为：

$$\delta(x, w) = \sum_{d \in D} \sum_{i \in I} \beta_{sd} \left| \sum_{j \in J} x_{sjd} - \sum_{i \in B_s} z_{id} \right|$$

实际在使用的时候，此权重一般取为整值。即我们在算法中做如下假定：

$$\beta_{sd} \in Z, \quad \forall s, d \quad (2-46)$$

所以我们在 $z$ 层引入 $\delta(x, w)$ 作为惩罚项，使得在当前解离解域较远时，通过目标中引入代表离解域距离的 $\delta(x, w)$ 惩罚项来强迫 $P(z^{(n)})$ 的目标强制上升，以尽快接近解域。而在很靠近解域时，其所占目标的值又较小，不会使原有目标(2-33)变差的很快，而进入可行域后这部分惩罚项为0，就完成其功能了。做了如此修改后，我们 $z$ 层即修正为：

$$\begin{aligned} P'(z) \quad \min \quad & \sum_{d \in D} \sum_{t \in T} \omega_{td} (z_{(N_i+t)d} + z_{(N_i+N_t+t)d}) + \eta^{(n)} \sum_{d \in D} \sum_{i \in I} \beta_{sd} \left| \sum_{j \in J} x_{sjd} - \sum_{i \in B_s} z_{id} \right| \\ \text{s.t.} \quad & \Phi_d \bar{z}_d = \Phi_d, \quad \forall d \in D. \end{aligned}$$

相对于算法2-5，迭代的框架不变，仅仅将其中的求解子规划 $P(z)$ 修正为求解子规划 $P'(z)$ 。为求简练，这里我们不赘述算法流程，记作算法2-6。

至于加入惩罚项后对求解的改进，有如下结论。

**定理2-3:** 我们记 $\xi_L^{(n)} = \sum_{d \in D} \sum_{t \in T} \omega_{td} (z_{(N_i+t)d}^n + z_{(N_i+N_t+t)d}^n)$ ， $\xi_L = \sum_{d \in D} \sum_{t \in T} \omega_{td} (z_{(N_i+t)d}^0 + z_{(N_i+N_t+t)d}^0)$ 。当第 $n$ 次迭代，我们选取惩罚项权重满足 $\eta^{(n)} > \xi_L^{(n-1)} - \xi_L$ 时，第 $n$ 次迭代结果惩罚项的值必小于或等于第 $n-1$ 次迭代惩罚项的值，即 $\delta^{(n)} \leq \delta^{(n-1)}$ 。

**证明：**用反证法，我们假设第 $n$ 步相较前一步惩罚项的值增大：

$$\delta(x^{(n)}, w^{(n-1)}) > \delta(x^{(n)}, w^{(n)}).$$

由 $\delta(x, w)$ 的定义我们可知，对任意 $n$ ， $\delta^{(n-1)}$ 均为整数。故而，惩罚项增大最少1，即：

$$\delta^{(n)} - \delta^{(n-1)} \geq 1.$$

当第 $n$ 步规划求解 $P(z^n)$ 时， $x^{(n)}$ 和 $\eta^{(n)}$ 均为常数，故而 $P(z^n)$ 的最优解 $z^n$ 满足：

$$L(x^n, z^{(n)}, \eta^{(n)}) \leq L(x^n, z^{(n-1)}, \eta^{(n)}). \quad (2-47)$$

当 $n > 1$ 时，由定理2-1我们可知， $\xi_L$ 是 $\xi_L^{(n)}$ 的下界，继而：

$$\begin{aligned} & L(x^n, z^{(n)}, \eta^{(n)}) - L(x^n, z^{(n-1)}, \eta^{(n)}) \\ = & \sum_{d \in D} \sum_{t \in T} \omega_{td} (z_{(N_i+t)d}^{(n)} + z_{(N_i+N_t+t)d}^{(n)}) + \eta^{(n)} \sum_{d \in D} \sum_{i \in I} \beta_{sd} \left| \sum_{j \in J} x_{sjd}^{(n)} - \sum_{i \in B_s} z_{id}^{(n)} \right| \\ & - \sum_{d \in D} \sum_{t \in T} \omega_{td} (z_{(N_i+t)d}^{(n-1)} + z_{(N_i+N_t+t)d}^{(n-1)}) + \eta^{(n)} \sum_{d \in D} \sum_{i \in I} \beta_{sd} \left| \sum_{j \in J} x_{sjd}^{(n)} - \sum_{i \in B_s} z_{id}^{(n-1)} \right| \\ = & \xi_L^{(n)} + \eta^{(n)} \delta(x^{(n)}, w^{(n)}) - \xi_L^{(n-1)} - \eta^{(n)} \delta(x^{(n)}, w^{(n-1)}) \\ \geq & \xi_L - \xi_L^{(n-1)} + \eta^{(n)} (\delta(x^{(n)}, w^{(n)}) - \delta(x^{(n)}, w^{(n-1)})) \\ \geq & \xi_L - \xi_L^{(n-1)} + \eta^{(n)} > 0, \end{aligned}$$

这与(2-47)式矛盾。

当 $n = 1$ 时， $\xi_L^{(n-1)} = \xi_L$ ，且 $\eta^{(1)} > \xi_L^{(0)} - \xi_L = 0$ ，所以我们有

$$\begin{aligned} & L(x^1, z^{(1)}, \eta^{(1)}) - L(x^1, z^{(0)}, \eta^{(1)}) \\ = & \xi_L^{(1)} - \xi_L + \eta^{(1)} (\delta(x^{(1)}, w^{(1)}) - \delta(x^{(1)}, w^{(0)})) \\ \geq & \xi_L - \xi_L + \eta^{(1)} > 0, \end{aligned}$$

这也与(2-47)式矛盾。

在两种情况下均有矛盾出现，说明我们的假设条件为谬，反之就可得到定理中给出的结论。  $\square$

## 2.5 数值实验

我们的实验环境选取为大型现代呼叫中心。假定有88个员工，即 $N_j = 88$ ，一个排班周期有31天，即 $N_d = 31$ 。我们将每天24小时分为48个等长的半小时时段，即 $N_t = 48$ ，每天可安排的班次有472个，即 $N_i = 472$ 。那么综合整个排班周期来看，相当于总共

有14,632个班次，应用于我们的模型中共有1,287,616个变量。为了更好的对比，我们本章的所有实验均在此环境和背景下实现。

关于数据，我们通过历史数据分别预测31天每天48个时段上的人力需求值。一天内最长用的模板为双峰的模式，而不同天的人力需求各不相同，其中每个月初始数天和最后的数天的需求较其他天明显较多，且一天内的需求分布更不均衡，这很好的切合了实际。

而如我们在2.2.1节中所述，员工人数与人力需求量之间应满足如下等式关系：

$$2AH_d \times \sum_{d \in D} J_d = \sum_{t \in T} \sum_{d \in D} HR_{td}.$$

如果不满足，我们则根据应用环境相应调整员工人数或人力需求量。

在第一种情形下，预测出的人力需求明显多于或少于员工人数，而在短时期内员工又很难大量增加或减少，那么我们只能固定员工人数，然后用当前仅有的员工尽量去拟合需求曲线，故而我们根据平均工时，总工作时长，员工人数等比例放缩人力需求量，如下式所示：

$$HR_{td} = \frac{2AH_d \times N_j \times AD}{\sum_{t \in T} \sum_{d \in D} HR_{td}} HR'_{td}, \quad \forall t \in T, \forall d \in D.$$

在第二种情形下，员工人数可以灵活调整，以求最佳的拟合人力需求，那么我们固定人力需求量，根据平均工时，总工作时长，员工人数等参数等比例放缩员工人数，如下式所示：

$$J_d = \frac{\sum_{t \in T} \sum_{d \in D} HR_{td}}{2AH_d \times N_j \times AD} J'_d, \quad \forall d \in D.$$

$$N_j = \frac{\sum_{t \in T} \sum_{d \in D} HR_{td}}{2AH_d \times AD}, \quad \forall d \in D.$$

### 2.5.1 模型规模比较

为了显示本文新构造算法的优越性，我们在其所能处理的规模上与已有文献进行比较，如表2-4所示。

此表说明了本文算法所能处理的模型规模比以往的要大得多。我们能一次安排31天的班表，每天都有472个班次，如果我们将整个一个排班周期内的排班看做是一个班次，那么我们总共就有14,632个班次。而从下一节的实验结果中看，我们的分解算法能够在不到10分钟的时间内求解带有1,287,616个变量和36,593个约束的此类整数规划问题。



表2-4: 与已有文献的模型规模比较

模型出处	[58]	[30]	[59]	[60]	[61]	[62]	[27]	[22]	[21]	本文
模型类型	LP	LP	TS	IP	IP	IP	IP	IP	IP	IP
每天工作时长	12,16	19	24	12,16	24	24	$\leq 24$	24	24	24
班次时长	4,9	9	7-9	4,6	9,9.5	8,10	7,9	8	-	6-10
每周工作天数	5	5	$\leq 7$	5	-	4,5	-	$\leq 5$	-	3-7
班次开始时间	F	V	F	F	V	F	V	F	V	V
休息时间	F	V	X	V	V	X	F	F	V	V
最小时间段间隔	1	0.25	0.25	1	0.5	1	0.25	8	0.25	0.5
最大变量个数	1,239	9,410	-	203	-	25,344	-	2,956	23,040	1,287,616
最大约束个数	-	-	-	-	-	-	4,752	3,120	15,648	36,593
应用范围	N	C	N	N	N	C	N	N	N	N

表2-4中, IP表示整数规划方法, LP表示线性规划方法, TS表示禁忌搜索方法, -表示参考文献中未提及此项, F表示为固定值, V表示为可变参数, C表示应用于呼叫中心的模型, N表示通用模型。

## 2.5.2 算法效率比较

本节, 我们对第二章提出的算法2-5与算法2-6进行比较, 来说明其针对不同问题的有效性。我们设计三组对比实验, 都来源于大规模呼叫中心的实例且具有相同的参数与数据, 只有在约束条件上有所不同, 一组为简单约束环境, 其实这里的简单仅是相对于后两种简单, 2.2.2节所述的所有约束类型均被涵括, 只是每类中约束的数目不如复杂环境多; 第二组为复杂约束环境, 为我们所做实例中最复杂的情形, 各种约束类型中均包含大量约束; 第三组为宽松环境, 约束的复杂程度介于前两组之间。

我们在一台2394 Mhz的X86计算机上做数值模拟, 操作系统为"windows XP SP3", 程序语言为C++, 编译环境为"MS Vc6.0"。在求解分解后的子规划 $P(x_j), P(z_d)$ 时, 使用商用优化软件"Xpress-MP2008A"。在C++环境中, 通过"Xpress.BCL.4.0.0"调用"Xpress.Optimizer.19.00.00"求解。

我们设 $\lambda$ 初值为0, 内层迭代最大次数为4, 即算法2-3中限制 $k \leq 4$ , 其它一些参数如:  $a_L = 0.8$ ,  $a_U = 1.2$ ,  $\theta_L = 1$ ,  $\theta_U = 10$ 。

首先是简单约束条件, 总计有1,287,616个变量, 11,132个约束。我们分别用算法2-5与算法2-6求解, 我们用图2-4来展示两个算法的收敛过程。

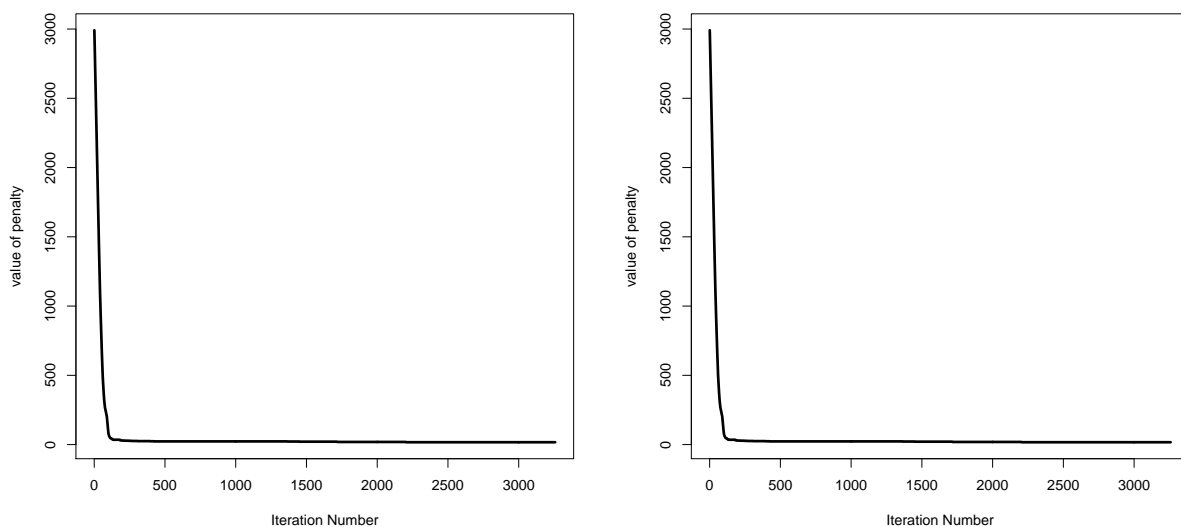


图2-4: 算法2-5与2-6在简单约束条件下的收敛过程比较

图2-4展示了改进算法2-6与原有算法2-5在简单约束条件下的收敛过程，左图为算法2-5收敛过程中目标函数值的变化情况，相应的右图为算法2-6的目标函数值的变化情况。

其次是一般宽松约束条件，总计有1,287,616个变量，22,264个约束。我们分别用算法2-5与算法2-6求解，我们用图2-5来展示两个算法的收敛过程。

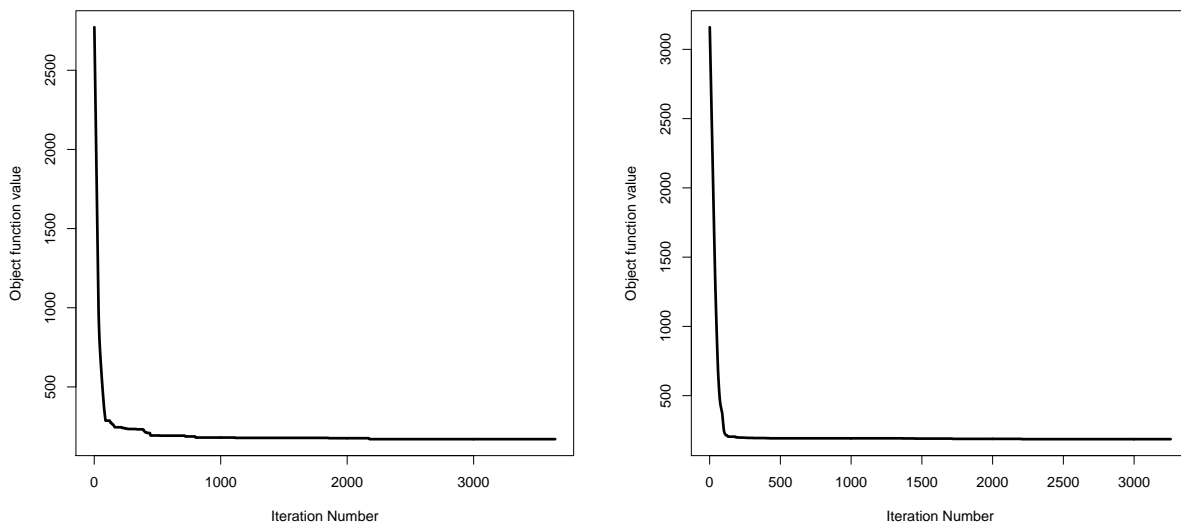


图2-5: 算法2-5与2-6在宽松约束条件下的收敛过程比较

图2-5展示了改进算法2-6与原有算法2-5在宽松约束条件下的收敛过程，左图为算法2-5收敛过程中目标函数值的变化情况，相应的右图为算法2-6的目标函数值的变化情况。

最后是严格约束条件，总计有1,287,616个变量，36,593个约束。我们分别用算法2-5与算法2-6求解，并用图2-6来展示两个算法的收敛过程。

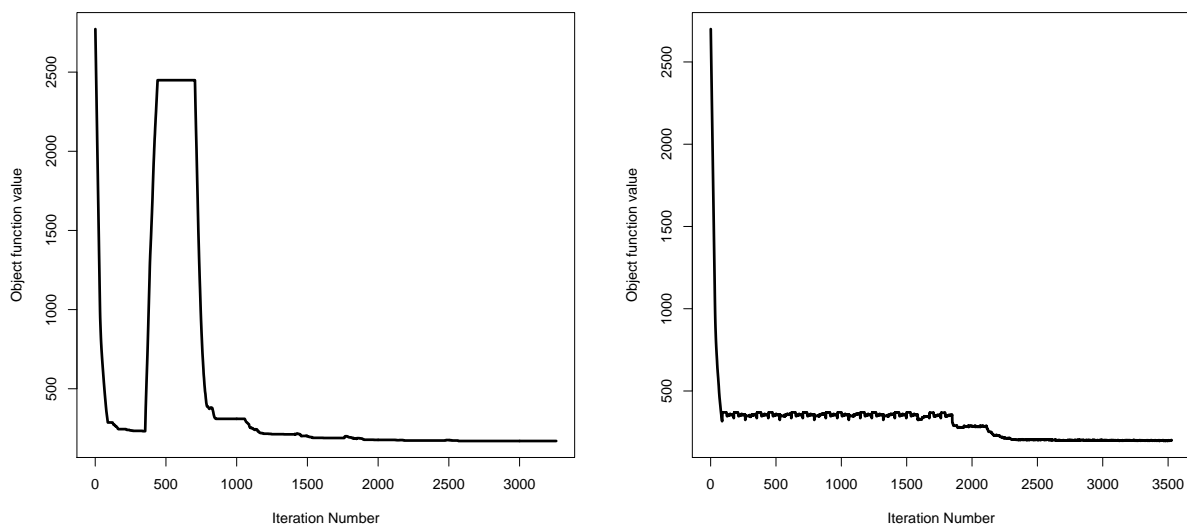


图2-6：算法2-5与2-6在严格约束条件下的收敛过程比较

图2-6展示了改进算法2-6与原有算法2-5在严格约束条件下的收敛过程，左图为算法2-5收敛过程中目标函数值的变化情况，相应的右图为算法2-6的目标函数值的变化情况。

我们将上述三组实例的信息汇总于表2-5。

表2-5：算法2-5与算法2-6在不同环境下的实证结果

环境	宽松	宽松	严格	严格	非常简单	非常简单
算法	2-5	2-6	2-5	2-6	2-5	2-6
班次个数	472	472	472	472	472	472
员工个数	88	88	88	88	88	88
变量个数	1,287,616	1,287,616	1,287,616	1,287,616	1,287,616	1,287,616
约束个数	22,264	22,264	36,593	36,593	11,132	11,132
约束违背程度	0	0	0	0	0	0
最优值	170	186	170	186	168	168
迭代次数	2176	2205	2567	3520	392	392
算法运行时间 (s)	227s	230s	523s	714s	42s	42s

我们在3种条件下比较这两种算法。首先，在简易的约束条件下，当 $n = 1$ 时，这两种算法都收敛于最优解，而且其表现基本相同。该两种算法的最优值均等于 $V_{P(z^{(0)})}$ ，此为 $n = 1$ 时算法终止的先决条件。在此情况下，两种算法只有很小的区别，因为算法2-6

中的惩罚项是为了克服在定理2-2中必要条件不能被满足的情况下的缺点而增加的，既然此时定理2-2中必要条件已被满足了，那么这两种算法的结果自然就相同。实际上，在简化的约束条件下， $n = 1$ 时收敛情况会频繁出现。

然后，我们可以看出在宽松的约束条件下，算法2-5相对于算法2-6有明显优势，不但体现在收敛速度更快上，同时其所求得解也更优，这里的原因就如同2.3.4节中所述，算法2-6其强迫原有目标上升，这可能在宽松的约束条件下并不值得，因为定理2-1中的不等式 $V_{P1} < V_{P(z^{(n-1)})}$ 很容易被满足。但是，除非是在迭代过程中，不等式 $V_{P1} < V_{P(z^{(n-1)})}$ 是否成立是很难检验的。

在严格的约束条件下，算法2-5在收敛过程中剧烈震荡。这表明在初始阶段，其很快落入了局部最优，跳出局部最优的过程即导致了这种剧烈震荡，但随着算法的进行，其又会逐渐逼近最优解。而算法2-6则因为惩罚项的引入，由定理2-3就可保证约束的偏差不会增大，这一点可以从图2-5中看出。

总的来说，这几组实验结果都在我们的预期之中，验证了2.3.4节中分析的正确性。

## 2.6 小结

相较已有文献，本章所建立的0-1混合整数规划模型采用更泛化的班次概念，更精确地描述了各类约束，从而具有更广泛的应用范围。更泛化与更精细地描述模型虽然能更好的切合实际，但自然会使得问题规模过于庞大，再加之本文基于的混合整数规划本身就是一NP-hard问题，大规模的此类问题用现有算法很难有效求解。为了克服这些因素带来的影响，我们从问题的约束矩阵结构出发，采用Benders分解，增广Lagrange松弛，对角二次逼近等嵌套分解算法有效的求解了这样一类大规模0-1混合整数规划问题。从实证结果来看收敛速度快，运行结果好，且人力需求拟合效果很好，合作方与客户对我们的结果均很满意。

### 3 两种新型概率约束规划模型与求解

为了更恰当、灵活地反应现实中的人力需求约束，本章提出两种新的概率约束规划模型，并探讨其求解与应用。

#### 3.1 模型介绍

为灵活反映实际中对人力需求约束的不同要求，我们将依据该类约束所限定的范围，采用两种概率约束形式进行建模：一种是分别对每个时段加以限制，另一种则是对整个周期引入统一的一个约束。

考虑有 $T$ 个时段， $J$ 个班次的排班问题，我们首先探讨对各个时段分别加以限制的模型。为精细刻画不同时段上人力需求的满足程度，我们将每个时段上所安排人数与该时段实际需求人数 $r_t, t = 1, \dots, T$ ，的差定义为人力不足量，同时给每个人力不足量一个下限 $-\delta_t, t = 1, \dots, T$ 。在每个时段上，根据问题的具体要求，保证人力不足量不低于 $-\delta_t$ 的概率不小于某个置信度 $\alpha_t$ 。那么，依极小化总的人力需求为目标，我们可建立下列概率约束规划模型：

$$\min \sum_{j=1}^J x_j \quad (3-1)$$

$$\text{s.t. } P\left(\sum_{j=1}^J a_{tj}x_j - r_t \geq -\delta_t\right) \geq \alpha_t, \quad t = 1, 2, \dots, T, \quad (3-2)$$

$$x \in X. \quad (3-3)$$

其中 $j = 1, \dots, J$ 表示不同的班次，要求任意两个班次均不相同。 $a_{tj} \in \{0, 1\}$ 表示班次 $j$ 在时段 $t$ 是否为上班状态，1表示上班，0表示不上班。 $x_j$ 为决策变量，意为上班次 $j$ 的员工数。 $X$ 表示除人力需求约束(3-2)外的其它所有约束。例如， $X$ 可包含如下三种典型的限制。

一是变量取整及其大于等于零的约束：

$$x_j \geq 0, x_j \in Z, \quad j = 1, 2, \dots, J.$$

二是公平性约束，即要求周期内所有上班员工的平均工作时长为预先给定的值 $AH$ ：

$$\left(\sum_{j=1}^J l_j * x_j\right) = AH * \sum_{j=1}^J x_j,$$

其中 $l_j$ 表示班次 $j$ 在排班周期内的上班时长。

三是特殊班次限制，即要求不同班次集合 $r$ 中安排的员工人数不小于某预先给定的下界 $SR_r^{\min}$ ，同时不大于某预先给定的上界 $SR_r^{\max}$ ：

$$\sum_{j \in G_r} x_j \geq SR_r^{\min}, \quad \forall r \in \Delta,$$

$$\sum_{j \in G_r} x_j \leq SR_r^{\max}, \quad \forall r \in \Delta,$$

其中 $G_r$ 表示班次集合 $r$ 中包含的班次。 $\Delta$ 表示所有需满足相应类型约束的班次集合的集合。

当然，我们还可根据实际需要，增加更多更详细的约束，因这部分约束并不牵扯随机因素，故而可参考如文献[20,28]等中确定性排班问题的建模方法。因非本文考虑的重点，以下不再对其进行详细论述。

若不对不同时段赋予特定的满足概率值，则可采用统一的置信水平，并给出另一种形式的概率约束表述：设要求各个时段上人力不足量均不低于最低限度的概率不小于某个置信度 $\alpha$ ，则可构建如下的联合概率约束模型：

$$\min \sum_{j=1}^J x_j \quad (3-4)$$

$$\text{s.t. } P\left(\sum_{j=1}^J a_{tj}x_j - r_t + \delta_t \geq 0, t = 1, \dots, T\right) \geq \alpha, \quad (3-5)$$

$$x \in X. \quad (3-6)$$

上述两种模型(3-1)-(3-3)和(3-4)-(3-6)与已有文献中的模型<sup>[38,39]</sup>相比更加灵活，可根据对不同天、不同时段关注程度加以不同的偏差容忍程度和概率约束满足程度。同时，在 $X$ 中也综合考虑了确定性排班文献中较新的成果<sup>[20,28]</sup>。

## 3.2 模型求解

相较概率约束规划模型(3-1)-(3-3)，联合概率约束(3-5)中的概率事件并非由单一线性不等式所表示，而是由 $T$ 个线性不等式联合表示。对此，可通过无穷范数将其表示为由一个不等式表示的概率事件的形式。为书写方便并用统一方式表达这两种概率约束模型，我们引入以下记号。首先，令 $x = (x_1, x_2, \dots, x_J)^T \in R^J$ ， $r = (r_1, r_2, \dots, r_T)^T \in R^T$ ， $f(x) = \sum_{j=1}^J x_j$ 。进而，记

$$G_t(x, r_t) = \sum_{j=1}^J a_{tj}x_j - r_t + \delta_t, \quad t = 1, 2, \dots, T,$$

$$q_t(x) = P(G_t(x, r_t) \geq 0) = \mathbb{E}(\mathbb{1}_{(0, \infty)} G_t(x, r_t)), \quad t = 1, 2, \dots, T.$$

这里 $\mathbb{1}_{(0, \infty)} G_t(x, r_t)$ 表示指示函数, 即当 $(G_t(x, r_t)) \geq 0$ 时,  $\mathbb{1}_{(0, \infty)} G_t(x, r_t) = 1$ , 当 $(G_t(x, r_t)) < 0$ 时,  $\mathbb{1}_{(0, \infty)} G_t(x, r_t) = 0$ 。类似地, 记

$$G(x, r) = \min \left\{ \sum_{j=1}^J a_{tj} x_j - r_t + \delta_t, \quad t = 1, \dots, T \right\},$$

$$q(x) = P(G(x, r) \geq 0).$$

则概率约束模型(3-1)-(3-3)就可转化为

$$\min \quad f(x) \tag{3-7}$$

$$\text{s.t.} \quad q_t(x) \geq \alpha_t, \quad t = 1, 2, \dots, T, \tag{3-8}$$

$$x \in X. \tag{3-9}$$

而概率约束模型(3-4)-(3-6)则可转化为

$$\min \quad f(x) \tag{3-10}$$

$$\text{s.t.} \quad q(x) \geq \alpha, \tag{3-11}$$

$$x \in X. \tag{3-12}$$

因转化后的两个模型具有相同的形式, 以下仅就概率约束模型(3-7)-(3-9)的求解展开论述。

为克服已有算法对分布类型的依赖性、难于实现的不足, 我们采用随机规划领域中新近提出的样本平均近似(SAA, sample average approximation, [40,63])策略来转化概率约束, 其最大的优点在于不需要假定原始的分布类型, 仅仅通过抽取一组样本来近似原随机分布, 且可保证收敛性。

对随机变量 $r_t$ , 设 $r_t^i, i = 1, \dots, I$ , 为 $r_t$ 的 $I$ 个独立同分布的样本。我们用这 $I$ 个样本来近似其本身的分布, 并用这些样本的平均来近似其原本分布的期望, 即用

$$\hat{q}_t(x) = \sum_{i=1}^I \frac{1}{I} \mathbb{1}_{(0, \infty)}(G_t(x, r_t^i))$$

来近似代替 $q_t(x)$ 。那么, 对于给定的概率或置信水平 $\gamma_t \in [0, 1], t = 1, \dots, T$ , 问题(3-7)-(3-9)则可转化为下列最优化问题:

$$\min \quad f(x) \tag{3-13}$$

$$\text{s.t.} \quad \hat{q}_t(x) \geq \gamma_t, \quad t = 1, 2, \dots, T, \tag{3-14}$$

$$x \in X. \tag{3-15}$$

我们将此问题称之为原问题(3-7)-(3-9)在置信水平 $\gamma_t$ ,  $t = 1, \dots, T$ ,下的样本平均近似问题。关于上述样本平均近似方法的收敛性, 由文献[64]中的分析, 我们有如下结论:

**定理3-1:** 假设原始问题与样本平均近似问题的置信度相同即,  $\gamma_t = \alpha_t, t = 1, \dots, T$ , 且除概率约束外的可行域为紧集,  $G(x, r)$ 为Caratheodory函数, 且有如下条件成立: 设原始问题的最优解为 $\bar{x}$ , 对任给的 $\epsilon > 0$ , 存在 $x \in X$ , 使得 $\|x - \bar{x}\| \leq \epsilon$ 和 $q_t(x) > \alpha$ 。则 $\hat{v}_I \rightarrow v^*$ , 且当 $I \rightarrow +\infty$ 时, 以概率1有 $D(\hat{S}_I, S) \rightarrow 0$ , 这里 $\hat{v}_I$ ,  $\hat{S}_I$ 分别表示样本平均近似问题(3-13)-(3-15)的最优解和可行集, 而 $v^*$ ,  $S$ 分别表示原始问题(3-1)-(3-4)的最优解和可行集,  $D(\hat{S}_I, S)$ 表示集合 $\hat{S}_I$ 与 $S$ 之间的距离。

对比样本平均近似问题与通常离散分布下所得问题可以看出, 样本平均近似也可被看做一种具有 $I$ 个相等概率样本的特殊分布。在置信度不变的情况下, 当 $I$ 趋于无穷大的时候, 样本平均近似问题的解收敛于原问题的解。在 $I$ 足够大的情况下, 若 $\gamma_t \geq \alpha_t, t = 1, \dots, T$ , 则样本平均近似问题的可行解亦是真实问题的可行解, 而当 $\gamma_t < \alpha_t, t = 1, \dots, T$ 时, 样本平均近似问题的最优解是真实问题的下界。所以样本平均近似方法也可用于构造可行解或是估计最优解的下界。

因 $\hat{q}_t(x)$ 中含有非光滑的示性函数, 导致问题(3-13)-(3-15)难于处理。为此, 我们使用类似文献[63]中的“大M方法”, 通过引入辅助变量来将示性函数转化为线性形式。具体地, 令

$$z_t^i = \begin{cases} 1, & \text{若 } G_t(x, r_t^i) \geq 0, \\ 0, & \text{其他.} \end{cases} \quad (3-16a)$$

$$(3-16b)$$

则 $\hat{q}_t(x)$ 就可以用 $z_t^i$ 的线性函数表示如下:

$$\hat{q}_t(x) = \sum_{i=1}^I \frac{1}{I} z_t^i,$$

这里 $z_t^i$ 与 $x, r_t^i$ 之间应满足下列线性不等式:  $G_t(x, r_t^i) + M(1 - z_t^i) \geq 0, G_t(x, r_t^i) \leq Mz_t^i, z_t^i \in \{0, 1\}$ , 其中 $M$ 是一个取值比 $G_t(x, r_t^i)$ 的最大值还要大的数。因为在概率约束中我们已对 $z_t^i$ 加入了更严格的下界约束, 不等式 $G_t(x, r_t^i) \leq Mz_t^i$ 实际上已不起作用, 所以在模型中可不加入此不等式。

通过(3-16)式中 $z_t^i$ 的引入和上述的转换, 我们就可将原概率约束模型(3-1)-(3-4)转化为如下形式的整数线性规划问题:

$$\min f(x) \quad (3-17)$$

$$\text{s.t. } \sum_{i=1}^I \frac{1}{I} z_t^i \geq \alpha_t, \quad t = 1, 2, \dots, T, \quad (3-18)$$

$$\sum_{j=1}^J a_{tj} x_j + \delta_t - r_t^i + M(1 - z_t^i) \geq 0, \quad t = 1, 2, \dots, T, \quad i = 1, 2, \dots, I, \quad (3-19)$$

$$x \in X, \quad z \in \{0, 1\}^{I \times T}. \quad (3-20)$$



由于实际中 $I$ 、 $J$ 、 $T$ 等的值均比较大，上述问题往往是大规模的线性整数规划，直接用经典的分枝定界法或者割平面法很难在短时间内找到满意的解。为有效求解问题(3-17)-(3-20)，我们将借鉴处理大规模线性规划的分解算法的思想，通过对偶的方法将问题的求解分解为一系列小规模问题的处理。

观察整数线性规划(3-17)-(3-20)，除了概率约束外，其余约束都因为关联所有决策变量而不具可分性。如果强行将其关于各个样本分成 $I$ 个子规划问题，则可能会导致在不同情形下的最优决策不一致，为克服这一困难，我们引入类似随机规划中非预期性约束(nonanticipativity constraints)的如下一组约束，

$$x^{i_1} = x^{i_2}, i_1, i_2 = 1, \dots, I,$$

这里 $x^{i_1}$ 表示情景(样本) $i_1$ 下决策向量的值，其余类同。此约束要求任意两个不同情景下的决策相同。为减少此类约束的个数，我们可要求对每一个情景，都有一个约束来保证其下的决策与另一任意情景下的决策相同，这样仅需要 $I$ 个约束。值得注意的是，这里引入约束的顺序是多样的，如令任意情景下的决策均等于第一个情景下的决策：

$$x^i = x^1, i = 2, \dots, I,$$

抑或令任意两个相邻情景下的解均相等：

$$x^1 = x^I, x^i = x^{i-1}, i = 2, \dots, I.$$

以下我们采用第二种，即每两个相邻情景间进行约束的方法。

对应于上述引入的情景决策，原本的目标函数 $f(x)$ 就可以表示为 $f(x^1), f(x^2), \dots, f(x^I)$ 的某种线性组合，本文根据各个情景发生的概率将它们进行加权求和，即令，

$$\hat{f}(x) = \sum_{i=1}^I \frac{1}{I} f(x^i).$$

有了上述的非预期性约束，我们就可利用Lagrange松弛法，将所有非预期约束和概率约束通过引入Lagrange乘子放缩至目标函数中，从而保证剩余约束均相对于不同的情景 $i$ 可分。该方法的优点在于可通过一系列较容易求解的小规模问题的处理来逼近原本大规模、很难求解的问题，可大幅度提高求解效率，在较短时间内得到满足精度要求的解。

Lagrange松弛因子的引入将原规划问题转化为一个两层的优化问题。具体地，令 $\lambda^1 = [\lambda_1^1, \lambda_2^1, \dots, \lambda_T^1]^T$ ， $\lambda^2 = [\lambda_{1,1}^2, \lambda_{1,2}^2, \dots, \lambda_{1,J}^2, \dots, \lambda_{I,J}^2]^T$ ， $\lambda = [\lambda^{1T}, \lambda^{2T}]^T$ ，内层需求解的最优化问题可表述为

$$V_{LR}(\lambda) = \min \hat{f}(x) + \sum_{t=1}^T \lambda_t^1 \left( \sum_{i=1}^I \frac{1}{I} z_t^i - \alpha_t \right) + \sum_{j=1}^J (\lambda_{1,j}^2 (x_j^1 - x_j^I)) \quad (3-21)$$

$$+ \sum_{i=2}^I \lambda_{i,j}^2 (x_j^i - x_j^{i-1}) \quad (3-22)$$

$$\text{s.t.} \quad \sum_{j=1}^J a_{tj} x_j^i + \delta_t - r_t^i + M(1 - z_t^i) \geq 0, \quad t = 1, 2, \dots, T, \quad i = 1, 2, \dots, I \quad (3-23)$$

$$z \in \{0, 1\}^{I \times T}, \quad x^i \in X, \quad i = 1, \dots, I. \quad (3-24)$$

同时，外层的优化问题需对Lagrange乘子寻优，即求解下列优化问题：

$$\max_{\lambda_1 \geq 0} V_{LR}(\lambda). \quad (3-25)$$

对于整数线性规划问题(3-21)-(3-23)，因为目标函数和约束均可对 $i$ 分解，故而其可等价于 $I$ 个子规划。对于每个子规划，我们均可运用分枝定界法与割平面法相结合的方法来快速求解，如可直接利用Xpress等优化软件。以下不再赘述。

对于要求部分变量非负连续优化问题(3-24)，我们可采用基于线性搜索的优化算法。那么最核心的内容即在于如何确定搜索方向和选取合适的搜索步长。在本文的模型中，因原问题为离散问题，其本身不具有梯度，而精确的次梯度又很难得到，故我们使用以下方向

$$d = \left[ \sum_{i=1}^I \frac{1}{I} z_1^i - \alpha_1, \sum_{i=1}^I \frac{1}{I} z_2^i - \alpha_2, \dots, \sum_{i=1}^I \frac{1}{I} z_T^i - \alpha_T, x_1^1 - x_1^I, x_2^1 - x_2^I, \dots, \right. \\ \left. x_J^1 - x_J^I, x_1^2 - x_1^1, x_2^2 - x_2^1, \dots, x_J^2 - x_J^1, \dots, x_1^I - x_1^{I-1}, x_2^I - x_2^{I-1}, \dots, x_J^I - x_J^{I-1} \right]^T. \quad (3-26)$$

这种方向对应于文献中所谓的替代次梯度。对于一般线性整数规划问题，文献[51]肯定了替代次梯度方向在Lagrange松弛类方法中的有效性，认为它能减少对偶问题求解过程的计算量，这正是我们按上述方法确定搜索方向 $d$ 的原因。

虽然我们选取的方向可看作是对次梯度方向的近似，但因原问题的离散性，经典次梯度算法中的步长规则并不适用我们的问题。为有效求解问题(3-24)，我们采用在非光滑优化中较常用的信赖域技术<sup>[50,52]</sup>来确定搜索步长。信赖域方法的思想是依照每次迭代对目标函数值改进的效果来确定下一步的步长，若上一步迭代改进明显，说明此方向很好，则加大步长以求更快下降；若上一步迭代改进不明显，则缩小步长，以免在不好的方向上前进过多。应用到我们的问题，我们将通过实时观察此次迭代和上次迭代之间改进的程度，这里选用目标函数值总的下降量作为改进程度的度量。当本次迭代目标值减去上次迭代目标值小于 $\theta_L$ 时，则增大步长；当相应的差值大于 $\theta_U$ 时，则减小步长；当介于前述两种情形之间时，步长保持不变。

至此，我们可构建如下确定最优Lagrange乘子的迭代算法：

### 算法3-1：

步骤1：初始化Lagrange乘子 $\lambda_t^{1(0)} = 1, t = 1, \dots, T, \lambda_{i,j}^{2(0)} = 0, i = 1, \dots, I, j = 1, \dots, J$ ，记相应的乘子向量为 $\lambda^{(0)}$ ，并选取参数 $\theta_L > 1, 0 < \theta_U < 1, \eta_L$ 和 $\eta_U$ 的值。给定迭代步数上限 $K$ ，令 $k = 1, s_k = 1$ 。

步骤2: 求解对应于当前乘子 $\lambda^{(k)}$ 的最优化问题(3-21)-(3-23), 得最优解 $x^{(k)}$ 及最优值 $V_{LR}(\lambda^{(k)})$ 。

步骤3: 若 $V_{LR}(\lambda^{(k)}) - V_{LR}(\lambda^{(k-1)}) < \theta_L$ , 令 $s^{(k)} = \eta_L \cdot s^{(k-1)}$ ; 若 $V_{LR}(\lambda^{(k)}) - V_{LR}(\lambda^{(k-1)}) > \theta_U$ , 令 $s^{(k)} = \eta_U \cdot s^{(k-1)}$ ; 否则, 令 $s^{(k)} = s^{(k-1)}$ 。

步骤4: 通过(3-25)式计算当前步的搜索方向 $d^{(k)}$ , 并令 $\lambda^{(k+1)} = \lambda^{(k)} + s^{(k)} \cdot d^{(k)}$ 。若对某个 $t$ ,  $\lambda_t^{1(k+1)} < 0$ , 则令 $\lambda_t^{1(k+1)} = 0$ 。

步骤5: 若对任意的 $i_1, i_2$ 和 $j$ , 都有 $x_j^{i_1(k)} = x_j^{i_2(k)}$ , 则终止, 当前的解向量 $x^{(k)}$ 即为问题(3-17)-(3-20)的最优解; 若 $k > K$ , 终止, 取 $x^{(k)}$ 为问题(3-17)-(3-20)的近似解; 否则, 令 $k = k + 1$ , 转入步骤2。

对于联合概率约束模型(3-10)-(3-12), 亦可通过与对问题(3-7)-(3-9)类似的方法将其转化为下列整数线性规划问题:

$$\begin{aligned} \min \quad & \hat{f}(x) \\ \text{s.t.} \quad & \sum_{i=1}^I \frac{1}{I} z^i \geq \alpha, \\ & \sum_{j=1}^J a_{tj} x_j + \delta_t - r_t^i + M(1 - z^i) \geq 0, \quad t = 1, 2, \dots, T, \quad i = 1, 2, \dots, I, \\ & x \in X, \quad z \in \{0, 1\}^I. \end{aligned}$$

对于该问题, 可完全同上采用Lagrange松弛法进一步分解。因为求解算法的导出完全类似, 这里不再赘述。

### 3.3 数值例子

为说明本文所建模型的合理性, 所导出样本平均近似问题的可解性与所设计松弛分解算法的有效性与收敛性, 我们在本节进行两组数值实验, 均以某个实际话务中心的员工安排为背景,  $X$ 包括我们在模型(3-1)-(3-3)后简单介绍的公平性、特殊班次等约束。

首先考虑第一组实验, 我们利用历史话务量数据, 预测出未来一月的基本人力需求量。考虑到未来需求的随机波动性与预测的不精确性, 我们假定预测值存在一服从期望值为0的正态分布的随机扰动。也就是说, 人力需求量可被看做服从均值为预测人力需求量的正态分布。基于正态分布的累加分布函数, 我们很容易将模型(3-1)-(3-3)转化为整数线性规划问题, 并找到其精确最优解。

代替上述通常的预测方法, 我们采用样本平均法来构建形如(3-17)-(3-20)的规划问题。具体地, 使用不同数量的样本来近似人力需求量的分布, 然后在不同样本集下求解样本平均近似问题(3-17)-(3-20), 并将所得解与原问题的精确最优解进行比较, 检验近似问题的最优解是否随着样本数量的增大收敛于原问题的精确最优解。

为克服简单随机抽样工作量大、方差大的不足，我们使用等距抽样的方法来生成样本数据集。这种方法的好处在于抽样过程简单。相较于抽取 $I$ 个样本的简单随机抽样，等距抽样所需随机抽取的次数仅为1，这节约了很多工作量。我们分别选取20个样本与50个样本进行实验，同时与原问题进行对比。三组实验采用相同的置信度与偏差容忍度，并对任意 $t = 1, \dots, T$ ，取 $\alpha_t = 0.98$ ， $\delta_t = -2 + 0.05 * r_t$ 。为便于比较，我们在表3-1中列出实际中各天所需的最小人力：

表3-1:样本近似问题最优值与原始问题最优值的比较

天	1	2	3	4	5	6	7	8	9-31	合计
原始问题	84	84	77	70	68	66	68	68	...	2209
20个样本平均近似	87	88	81	73	71	69	71	71	...	2309
50个样本平均近似	85	85	78	71	68	66	68	68	...	2222

为对比说明不同结果，我们将不同天原问题精确的最优值、20和50个样本下样本平均问题的最优值，即每天所需最小人力绘制于图3-1中：

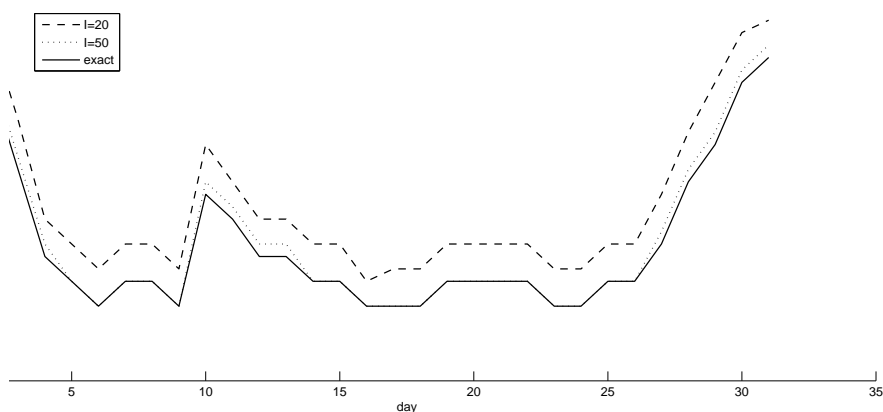


图3-1: 样本平均近似问题最优值与原问题最优值的比较.

从图3-1中不难看出，随着样本量的增大，样本平均近似问题的最优值逐渐逼近原始问题的最优值，尤其是在50个样本时，差别就已经很小了。

作为用户可调节的选项，偏差容忍度决定了概率约束在限制可行解上所起作用的大小，从而影响到问题最优值的大小。为此，我们考虑另一组实验，研究不同偏差容忍程度对问题最优解与最优值的影响。我们考虑两种形式的偏差：绝对偏差和相对偏差。不难理解，绝对(相对)偏差越少，对应问题的最优值越小，而相对偏差方式较绝对偏差方式，对应的最优值一般较大。

为验证上述结论，我们设计了四组实验，它们分别是：较大的绝对偏差 $\text{bias}_t = 1, t = 1, \dots, T$ ；较小的绝对偏差 $\text{bias}_t = 0.5, t = 1, \dots, T$ ；较大的相对偏差 $\text{bias}_t = 0.05 * r_t$

$r_t, t = 1, \dots, T$ 和较小的相对偏差 $\text{bias}_t = 0.03 * r_t, t = 1, \dots, T$ 。我们再统一令 $\delta_t = -2 + \text{bias}_t, t = 1, \dots, T, \alpha_t = 0.98, t = 1, \dots, T$ ，样本量 $I = 20$ 。为考虑概率约束不起作用时问题最优值的变化，我们特增加一次对照实验，它对应于 $\text{bias}_t = -\infty$ ，此时所得最优值应为上述情形下最优值的下界。上述各种情况下所得最优值见表3-2。

表3-2:不同偏差容忍程度下问题最优值的比较

天	1	2	3	4	5	6	7	8	9-31	合计
$\text{bias}_t = -\infty$ (无约束问题)	78	78	72	65	63	61	63	63	...	2055
$\text{bias}_t = 0.1$	82	81	75	68	66	64	66	66	...	2215
$\text{bias}_t = 0.5$	80	81	75	68	66	64	66	66	...	2140
$\text{bias}_t = 0.05 * r_t$	87	88	81	73	71	69	71	71	...	2309
$\text{bias}_t = 0.03 * r_t$	82	84	78	70	68	67	69	69	...	2218

为进一步了解整个月内最优值的变化，我们将上述五组解的结果在图3-2中予以展示。

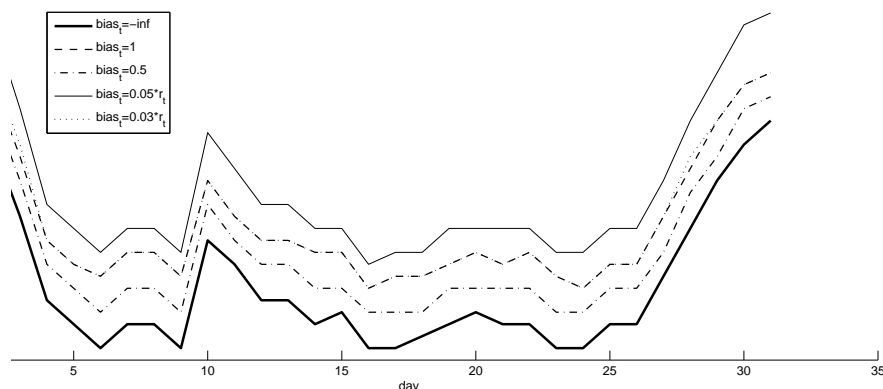


图3-2: 不同偏差容忍程度下问题最优值的比较

从表3-2与图3-2中可以看出，无论是绝对偏差还是相对偏差，偏差值较大时问题的最优值均较大，较小时问题的最优值较小，这与前述的理论结论是一致的。而绝对偏差情形与相对偏差情形相比较，绝对偏差下对应的最优值要小于相对偏差下的对应的最优值。

### 3.4 小结

相较已有文献，本文所建立的带有概率约束条件的智能排班模型更精确地描述了人力需求限制的要求，从大概率角度有效地限制了人力不足，更符合客户期望。而我们基于样本平均近似方法、Lagrange松弛法、分解算法等思想所设计的嵌套分解算法也

较已有算法更为有效。然而，在样本平均近似方法中，样本的选取对结果有着较大的影响，故而如何寻找一种更稳定、有效的抽样方法仍待更深入的研究。

## 4 结论与展望

智能排班问题是具有巨大潜在经济效益的一类优化问题。它要求在给定的一系列要求，比如每天的用工人员需求、工作强度要求、劳动保障要求等之下，对给定数量的工作班组进行合理的工休安排。由于问题本身涉及的因素非常复杂多样，所以尽管已有众多学者近半世纪来的深入研究，且取得相当多的成果，但仍存在很多问题未被解决。本文即对其中的部分问题展开了系统性研究，取得了一些成果。

### 4.1 结论

首先，本文对现有研究成果做了一个详实的文献综述，将近50年来的对排班问题的发展产生重大影响的文献予以归类，整理了几类子问题的发展脉络与相互联系，之后分别评述其各自的优缺点与应用范围。

之后，基于已有研究成果，本文提出了两种角度的建模思路：传统的确定性规划和新型的随机规划方法。

在确定性规划方面，我们在模型中引入了班次和班次类的概念。从而克服了现有模型难以用线性约束精确描述一日内多次休息，多次上下班的难题，使得整体模型得以线性化。该思想拓宽了模型的应用范围，可解决更加复杂的实际问题。但随之的问题是在处理大规模问题时，我们的0-1混合整数规划规模过于巨大，用已有文献的方法不能在有限时间内解决，故而我们提出一套针对此问题的嵌套分解方法，将原有的大规模问题分解为一系列的小问题序贯求解。

因为综合运用了多种处理超大规模复杂优化问题的近代优化算法技巧，我们的算法整体求解速度快，可应用于大规模的企事业单位。我们已经处理了有将近4万个变量、3万个约束的大规模问题，算法很快就可安排出满足所有约束的最优班表。更为重要的是，我们所提供的排班方案对人力需求拟合效果好。在增加大量且严格的约束的情况下，人力需求拟合效果依然很好，合作方与客户对我们的结果均很满意。

其次，针对确定性排班问题固定人力需求量的不足，本文后半部分将其考虑为随机变量，通过引入人力不足量的概念，使用概率约束条件，来刻画人力需求约束；为了确定化随机变量，我们又使用了随机规划领域中新近提出的样本平均近似方法来将概率约束规划问题转化为确定性整数规划问题，并利用“大M方法”的思想将其线性化；之后，为求解这样一个大规模混合整数规划问题，我们参考了文章前半部分的成熟技巧，利用有效的嵌套分解算法来予以求解。最后，通过实验我们发现在概率约束模型框架下，随着样本量的逐渐增大样本逐渐可逼近原始分布，故而说明，我们采用的样本平均

近似方法是行而有效的。

基于本文提出的所有方法，课题组与合作方共同开发了一个适用于大型呼叫中心的智能排班软件。其中的排班主程序先由本文作者独立完成（C++源程序近2000行），然后由合作方重构为面向对象的框架，本文作者还参加了部分面向对象版本程序的调试、承担了部分接口设计、重要数据结构的转化工作，剩余的数据库设计、人力需求预测部分的程序由合作方完成。我们共同完成的智能排班软件已作为一个重要子系统嵌入到一个更大的软件系统-呼叫中心调度、预测与仿真系统中。而且其中的智能排班部分已经作为研究成果的形式在国内外部分地区得到了应用，取得了良好效果。

## 4.2 展望

排班问题涉及的因素多种多样，本文中未加考虑或深入研究，而在实际中又比较重要的一些问题包括：

### 1) 人力需求的预测

无论是确定性模型还是随机模型，我们为了能最优地满足未来的需求，自然要面对预测的问题。当我们在努力拟合预测出的需求值时，首先要保证预测值的可信度，否则排班的模型再精细也无法适用于未来。而我们在本文中将历史数据作为总体来表现需求值，虽然说未来就是历史的延续，但单纯使用历史数据无法表现人力需求的某些可预知特性，故而从排队论或者其它角度来预测人力需求，或用历史数据预测时引入时间序列的方法，以展现某种趋势性变化，都是我们进一步可作的工作。

### 2) 多技能或多区域的问题

本文考虑的仅为单技能、单区域的模型，而现实中常常会有多个排班单位综合排班的情况，即所谓多区域；亦或不同员工掌握不同的技能很难交叉的情况，即所谓多技能。在这两种情况下，所需考虑的情形更加复杂，所需构建的模型规模更庞大，虽然我们的模型有过这方面的尝试且效果不错，但均是较为简单的推广或经过简单化的处理之后加以应用；更详尽的约束描述方式，更切合多区域或多技能模型的求解方法还需进一步的研究。

### 3) 混合整数规划更有效的求解算法

虽然本文针对这一类特殊的0-1混合整数规划提出一套行为有效的嵌套分解方法，但本身作为NP难问题，寻求更快更好求解方法的道路是永没有止境的。我们希望更进一步的研究能从数学角度给出更佳更鲁棒且理论收敛特性更加好的新算法。

### 4) 针对不同市场的应用

本文的模型所给出依然是针对智能排班问题的泛化通用模型，可应用于各种新型的排班市场，比如呼叫中心、交通调度、医院学校等不同环境，如何给出更适合所应用的环境的具体模型，如何构建并加入针对具体问题的约束条件，这都是在推广此模型中很实际的问题，也是我们还待研究的领域。



### 5) 新模型与算法在国内的具体应用

总的来说,我国在排班或调度领域的智能化程度还比较低,与发达国家相比,还处于靠手工凭经验的较初级阶段,经常会出现排出的班表不科学,不公平,员工不满意的现象。当然,随着科技的进步这种情况在未来会逐渐改善,用计算机来统筹规划必然会成为发展的趋势。所以,研究符合我国国情的智能排班方法将是一个长期,复杂但却意义重大的课题,需要持续不断的艰苦探索。

所有以上问题,都是我们未来亟待解决的问题。

## 致 谢

本文是在我的导师陈志平教授的悉心指导下完成的。从课题的选择到论文的最终完成，陈老师都始终给予我细心的指导和不懈的支持。他严肃的科学态度，严谨的治学精神，精益求精的工作作风，深深地感染和激励着我。这两年多来，陈老师不仅在学业上给我以精心指导，同时还在思想、生活上给我以无微不至的关怀，在此谨向陈老师致以诚挚的谢意和崇高的敬意。

在此，我要感谢各位师兄弟、同学及舍友，和你们在一起，我度过了三年愉快的研究生生活。特别感谢的是项目合作方的赵荣乾学长以及项目组成员杨海源、宋振霞和胡乾慧等同学，正是由于你们的帮助和支持，我才能克服一个一个的困难和疑惑，将所参加项目顺利完成。此外，翟桥柱老师、段启宏老师、程蓓同学以及数学系的很多朋友在完成论文的过程中给予了我很大帮助，在这里请接受我诚挚的谢意!

最后我还要感谢培养我长大含辛茹苦的父母，是你们的支持让我顺利完成学业，谢谢你们!

参考文献

- [1] Dantzig G. A comment on Edie's traffic delay at toll booths[J]. *Operations Research*, 1954, 2: 339 - 341
- [2] Baker K R. Workforce allocation in cyclic scheduling problems: a survey[J]. *Operational Research Quarterly*, 1976, 27: 155-167
- [3] Baker K R. Magazine M T. Work force scheduling with cyclic demands and days-off constraints[J]. *Management Science*, 1977, 24: 161-167
- [4] Hung R. Multiple-shift workforce scheduling under the 3-4 workweek with different weekday and weekend labor requirements[J]. *Management Science*, 1994, 40(2): 49-57
- [5] Costa M C, Jarray F, Picouleau C. An acyclic days-off scheduling problem[J]. *4OR*, 2006, 4(1): 73-85
- [6] Elshafei M, Alfares H K. A dynamic programming algorithm for days-off scheduling with sequence dependent labor costs[J]. *Journal of Scheduling*, 2008, 11(2): 85-93
- [7] Hung R. Single shift off-day scheduling of a hierarchical workforce with variable demands[J]. *European Journal of Operational Research*, 1994, 78(1): 280-284
- [8] Narasimhan R. An algorithm for multiple shift scheduling of hierarchical workforce on four-day or three-day workweeks[J]. *Information Systems and Operational Research*, 2000, 38(1): 14-32
- [9] Azmat C S, Hurlimann T, Widmer M. Mixed integer programming to schedule a single-shift workforce under annualized hours[J]. *Annals of Operations Research*, 2004, 128: 199-215
- [10] Azmat C S, Widmer M. A case study of single shift planning and scheduling under annualized hours: A simple three-step approach[J]. *European Journal of Operational Research*, 2004, 153(1): 48-175
- [11] Billionnet A. Integer programming to schedule a hierarchical workforce with variable demands[J]. *European Journal of Operational Research*, 1999, 114: 105-114
- [12] Ernst A T, Jiang H, Krishnamoorthy M, Sier D. Staff scheduling and rostering: a review of applications, methods and models[J]. *European Journal of Operational Research*, 2004, 153: 3-27
- [13] Helber S, Henken K. Profit-oriented shift scheduling of inbound contact centers with skills-based routing, impatient customers, and retrials[J]. *OR Spectrum*, 2010, 32(1): 109-134
- [14] Bechtold S E, Jacobs L W. Implicit modeling of flexible break assignments in optimal shift scheduling[J]. *Management Science*, 1990, 36(11): 1339-1351

- 
- [15] Segal M. The operator-scheduling problem: a network flow approach[J]. *Operations Research*, 1974, 22: 808-823
- [16] Keith L, Telliard W. ES&T Special Report: Priority pollutants: I-a perspective view[J]. *Environment Science and Technology*, 1979, 13(4): 416 - 423
- [17] Seckiner, Ulusam S, Kurt M. Scheduling A Hierarchical Workforce with Flexible break assignments and variable demands[C]. *Operational Research Industrial Engineering 24th National Congress*, Gaziantep Turkey, 2004
- [18] Moondra S L. An LP model for work force scheduling for banks[J]. *Journal of Bank Research*, 1976, 6: 299-301
- [19] Gaballa A, Pearce W. Telephone sales manpower planning at Qantas[J]. *Interfaces*, 1979, 9(3): 1-9
- [20] Aykin T. A comparative evaluation of modeling approaches to the labor shift scheduling problem[J]. *European Journal of Operational Research*, 2000, 125: 381-397
- [21] Rekik M, Cordeau J F. Implicit shift scheduling with multiple breaks and work stretch duration restrictions[J]. *Journal of Scheduling*, 2009, 10:
- [22] Yakoob S A, Sherali H. Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations[J]. *Annals of Operations Research*, 2007, 155: 119-142
- [23] Yakoob S A, Sherali H. Multiple shift scheduling of hierarchical workforce with multiple work centers[J]. *INFORMATICA*, 2007, 18(3): 325-342
- [24] Musliu N. Local search strategies for rotating workforce scheduling[C]. In Paper presented at the Workshop on Design and Evaluation of Advanced Hybrid Meta-Heuristics, Nottingham, UK, 2004
- [25] Alfares H K, Lilly M T, Emovon I. Maintenance staff scheduling at afam power station[J]. *IEMS*, 2007, 6(1): 22-27
- [26] Bard J F, Purnomo H W. Cyclic preference scheduling of nurses using a Lagrangian-based heuristic[J]. *Journal of Scheduling*, 2007, 10(1): 5-23
- [27] Addou I, Soumis F. Bechtold-Jacobs generalized model for shift scheduling with extraordinary overlap[J]. *Annals of Operations Research*, 2007, 155: 177-205
- [28] Beaumont N. Scheduling staff using mixed integer programming[J]. *European Journal of Operational Research*, 1997, 98(3): 473-484
- [29] Rekik M, Cordeau J F, Soumis F. Using Benders decomposition to implicitly model tour scheduling[J]. *Annals of Operations Research*, 2004, 128: 111-133
- [30] Cezik T, Gunluk O, Luss H. An integer programming model for the weekly tour scheduling problem[J]. *Naval Research Logistics*, 2001, 48(7): 607 - 624

- 
- [31] Li Y J, Chen J, Cai X Q. An integrated staff-sizing approach considering feasibility of scheduling decision[J]. *Annals of Operations Research*, 2007, 155(1): 361-390
- [32] Eitzen G, Panton D, Mills G. Multi-skilled workforce optimisation[J]. *Annals of Operations Research*, 2004, 127: 359-372
- [33] Alfares H K. Survey, Categorization, and Comparison of Recent Tour Scheduling Literature[J]. *Annals of Operations Research*, 2004, 127: 145-175
- [34] Aksin Z, Armony M, Mehrotra V. The modern call center: a multi-disciplinary perspective on operations management research[J]. *Production and operations management*, 2007, 16(6): 665-688
- [35] Caprara A, Monaci M, Toth P. Models and algorithms for a staff scheduling problem[J]. *Mathematical Programming*, 2003, 98(3), 445-476
- [36] Gan S, et al. Mitotic and postmitotic senescence in plants[M]. *Science's SAGE KE*, 2003
- [37] Easton F F, Rossin D F. A Stochastic goal program for employee scheduling[J]. *Decision Sciences*, 1996, 27(3): 541 - 568
- [38] Mabert V. A case study of encoder shift scheduling under uncertainty[J]. *Management Science*, 1979, 25(7): 623-631
- [39] Bordoloi S K. Agent recruitment planning in knowledge-intensive call centers[J]. *Journal of Service Research*, 2004, 6(4): 309-323
- [40] Ruszczyński A, Shapiro A. *Handbooks in OR&MS: Stochastic Programming*[M]. North-Holl and Amsterdam, 2003
- [41] Easton F F, Rossin D F. Sufficient working subsets for the tour scheduling problem[J]. *Management Science*, 1991, 37(11): 1441 - 1451
- [42] Bechtold S E, Showalter M J. A methodology for labor scheduling in an service operating system[J]. *Decision Sciences*, 1987, 18(1): 89 - 102
- [43] Easton F F, Mansourb N. A distributed genetic algorithm for deterministic and stochastic labor scheduling problems[J]. *European Journal of Operational Research*, 1999, 118(3): 505-523
- [44] Beaumont N B. Using mixed integer programming to design employee rosters[R]. Working Paper, Department of Business Management, Monash University, 1996
- [45] Aykin T. Optimal Shift Scheduling with Multiple Break Windows[J]. *Management Science*, 1996, 42(4): 591-602
- [46] Artigues C, Gendreau M, Rousseau L M. A flexible model and a hybrid exact method for integrated employee timetabling and production scheduling[C]. *Practice and Theory of Automated Timetabling VI*, 2007, 6: 67-84
- [47] Li H T, Womer K. Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm[J]. *Journal of Scheduling*, 2009, 12(3): 281-298

- [48] Ruszczyński A. On convergence of augmented lagrangian decomposition method for sparse convex optimization[J]. *Mathematics of Operations Research*, 1995, 20(3): 634-656
- [49] Zhai Q Z, Guan X H, Cui J. Unit commitment with identical units: successive subproblem solving method based on lagrangian relaxation[J]. *IEEE transactions on power system*, 2002, 17(4): 1250-1256
- [50] Fletcher R. *Practical methods of optimization*, 2nd edition[M]. Chichester: U.K. Wiley, 1987
- [51] Zhao X, Luh P B, Wang J. Surrogate gradient algorithm for lagrangian relaxation[J]. *Journal of Optimization Theory and Applications*, 1999, 100(3): 699-712
- [52] Yuan Y. Conditions for convergence of trust region algorithm for nonsmooth optimization[J]. *Mathematical Programming*, 1985, 31: 220-228
- [53] Gabay D, Mercier B. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations[J]. *Computers & Mathematics with Applications*, 1976, 2: 17-40
- [54] Stephanopoulos G, Westerberg W. The use of Hestenes's method of multipliers to resolve dual gaps in engineering system optimization[J]. *Journal of Optimization Theory and Applications*, 1975, 15: 285-309
- [55] Mulvey J M, Ruszczyński A. A new scenario decomposition method for large-scale stochastic optimization[J]. *Operations Research*, 1995, 43(3): 477-484
- [56] Guan X H, Zhai Q Z, Lai F. New lagrangian relaxation based algorithm for resource scheduling with homogeneous subproblem[J]. *Journal of Optimization and Applications*, 2002, 113(1): 65-82
- [57] Bertsekas DP. *Constrained optimization and Lagrange multiplier methods*[M]. Academic Press, NY, 1982
- [58] Brusco M J, Johns T R. Improving the dispersion of surplus labor in personnel scheduling solutions[J]. *Computers and Industrial Engineering*, 1995, 28(4): 745 - 754
- [59] Gartner J, Musliu N, Slany W. Rota: A Research project on algorithms for workforce scheduling and shift design optimization[J]. *AI communications*, 2001, 14(2): 83 - 92
- [60] Brusco M J. Solving personnel tour scheduling problems using the dual all-integer cutting plane[J]. *IIE Transactions*, 1998, 30(9): 835 - 844
- [61] Lin C K Y, Lai K F, Hung S L. Development of a workforce management system for a customer hotline service[J]. *Computers and Operations Research*, 2000, 27(10): 987 - 1004
- [62] Brusco M J, Jacobs L W. Starting-time decisions in labor tour scheduling: an experimental analysis and case study[J]. *European Journal of Operational Research*, 2001, 131(3): 459 - 475
- [63] Luedtke J, Ahmed S. A sample approximation approach for optimization with probabilistic constraints[J]. *SIAM Journal on Optimization*, 2008, 19: 674-699

- [64] Pagnoncelli B K, Ahmed S, Shapiro A. Sample average approximation method for chance constrained programming: theory and applications[J]. *Journal of Optimization Theory and Application*, 2009, 142(2), 399-416
- [65] 翟桥柱. 电力系统优化调度模型与算法研究[D]. 西安交通大学博士学位论文, 2005
- [66] 陈志平, 徐宗本. 计算机数学—计算复杂性理论与NPC、NP难问题的求解[M]. 北京: 科学出版社, 2001
- [67] 邢文训, 谢金星. 现代优化计算方法[M]. 北京: 清华大学出版社, 1999

附件 1:

## 学位论文独创性声明

本人声明，所呈交的学位论文系在导师指导下本人独立完成的研究成果。文中依法引用他人的成果，均已做出明确标注或得到许可。论文内容未包含法律意义上已属于他人的任何形式的研究成果，也不包含本人已用于其他学位申请的论文或成果。

本人如违反上述声明，愿意承担以下责任后果：

1. 交回学校授予的学位证书；
2. 学校可在相关媒体上对作者本人的行为进行通报；
3. 本人按照学校规定的方式，对因不当取得学位给学校造成的名誉损害，进行公开道歉。
4. 本人负责因论文成果不实产生的法律纠纷。

论文作者签名：\_\_\_\_\_

日期：\_\_\_\_\_年\_\_\_\_月\_\_\_\_日

## 学位论文知识产权权属声明

本人在导师指导下所完成的论文及相关的职务作品，知识产权归属学校。学校享有以任何方式发表、复制、公开阅览、借阅以及申请专利等权利。本人离校后发表或者使用学位论文或与该论文直接相关的学术论文或成果时，署名单位仍然为西安交通大学。

论文作者签名：\_\_\_\_\_

日期：\_\_\_\_\_年\_\_\_\_月\_\_\_\_日

导师签名：\_\_\_\_\_

日期：\_\_\_\_\_年\_\_\_\_月\_\_\_\_日

（本声明的版权归西安交通大学所有，未经许可，任何单位及个人不得擅自使用）