

城轨交通乘务任务配对的集合分割模型及算法

石俊刚, 周峰, 徐瑞华

(同济大学 道路与交通工程教育部重点实验室, 上海 201804)

摘要: 基于城轨乘务值乘规则, 建立了城轨乘务任务配对的集合分割模型(R-CP), 并结合列生成思想和跟随分支策略设计了求解算法(CGBF 算法). 实例验证结果表明: 该模型与算法能够满足乘务任务配对方案的要求, 相对于当前手工编制, 具有编制效率高、目标函数更优(即任务数量少、任务工作效率高)等特点.

关键词: 城市轨道交通; 乘务任务配对; 集合分割问题; 列生成思想; 跟随分支策略

中图分类号: U293.5

文献标志码: A

Set Partition Model and Algorithm for Crew Pairing Problem in Urban Rail Transit

SHI Jungang, ZHOU Feng, XU Ruihua

(Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai 201804, China)

Abstract: A set partition model (R-CP) for the crew pairing problem in urban rail transit was proposed based on practical considerations in rail transit operations. A hybrid algorithm of column generation and branch on follow-ons (CGBF) was designed to solve R-CP. The numerical results show that the proposed model and algorithm can meet requirements of crew pairing and yield better objective values than the existing manual methods.

Key words: urban rail transit; crew pairing problem; set partition problem; column generation; branch on follow-ons

乘务计划编制是城市轨道交通(以下简称城轨)乘务管理的重要工作, 良好的乘务计划既能保证乘务员之间工作的均衡性, 又能保证乘务员的驾车效率和安全性, 其中乘务任务配对是乘务计划编制的核心. 目前, 我国城轨乘务任务配对基本采用人工编制, 且随着列车发车间隔缩短, 运营时间增长, 乘务

任务配对的工作难度越来越大, 且编制质量低. 设计高效的乘务任务配对模型及算法成为亟需解决的问题.

国内外在乘务任务配对问题的研究主要集中在航空和高铁领域, 城轨领域的研究成果相对较少. 航空领域, Anbil 等^[1], Wedelin^[2], AhmadBeygi 等^[3]将乘务任务配对问题归纳为集合分割问题(set partition problem, SPP), 并分别采用分支定界法、分支裁剪法和动态规划法进行求解; Beasley^[4], Aydemir-Karadag 等^[5], Souai 等^[6]则将问题归纳为集合覆盖问题(set covering problem, SCP), 并分别采用拉格朗日次梯度算法、组合优化算法和遗传算法进行求解. 航空乘务任务配对研究成熟, 但与城轨在乘务值乘规则、运营模式等方面差异较大, 难以完全借鉴. 高铁领域, 夏平等^[7]构建了乘务任务配对的多目标规划模型, 并设计回溯搜索策略求解; 程岩岩^[8]构建了乘务任务配对的旅行商(TSP)模型, 并采用列生成思想求解; 赵鹏^[9]、田志强^[10]、王莹等^[11]建立了乘务任务配对问题的 SPP 模型, 并分别采用系统工程(SE)方法、蚁群算法、分支定界法进行求解. 高铁乘务任务配对虽然与城轨相似度较高, 但城轨发车密度大, 轮换点多, 求解复杂度高, 因此其模型与算法难以适用. 城轨领域, 刘德利^[12]对乘务值乘方式进行了探讨; 李献忠等^[13]构建了基于广义费用的乘务排班模型, 并设计采用禁忌搜索算法求解, 但优化程度有限; 石俊刚等^[14]结合乘务劳动作息规则, 建立了相应的 SCP 模型, 并结合列生成思想设计了算法, 优化效果较好, 但不能避免随乘现象(随乘现象指 2 个乘务员同时驾驶一列车, 其中 1 名乘务员担任司机, 另 1 名乘务员以乘客身份随乘前往目的地), 存在缺憾.

本文以城轨乘务任务配对为研究对象, 以提高乘务任务配对的质量、效率及合理性为目标, 建立乘

收稿日期: 2014-08-06

基金项目: 国家自然科学基金(71271153, 51408323); 中国博士后科学基金(2014M551454); 浙江省自然科学基金(LQ13G010010)

第一作者: 石俊刚(1986—), 男, 博士生, 主要研究方向为城轨乘务计划优化编制方法. E-mail: 1020120112@tongji.edu.cn

通讯作者: 徐瑞华(1963—), 男, 教授, 博士生导师, 工学博士, 主要研究方向为运输组织现代化. E-mail: rhxu@tongji.edu.cn

务任务配对的集合分割模型,并设计了相应的算法。

1 问题描述与建模

1.1 问题描述

乘务计划主要包括两个部分工作:乘务任务配对和乘务任务轮转。其中,乘务任务配对指将给定的列车运行计划(即运行图)配对成若干乘务任务,每个乘务任务由 1 个乘务员一天完成;乘务任务轮转指按照特定的轮转制度(四班两转、五班三转等)安排乘务员的阶段任务。乘务任务配对为乘务任务轮转的基础,本文研究重点为乘务任务配对问题。

城轨乘务任务配对主要涉及以下概念:出退勤点、轮换点、用餐点、任务段和乘务任务。出退勤点指乘务员出勤和退勤的车站,轮换点指乘务员中途下车轮换休息的车站,用餐点指乘务员用餐的车站。出退勤点、轮转点、用餐点均为允许乘务员下车停留的地点,本文统一称为乘务驻点。一列列车在两个相邻乘务驻点之间的驾驶段称为任务段。如图 1 所示,双实线箭头代表由线路 1 始发站开往终到站的列车,该列车可由乘务驻点拆分成 3 个任务段。

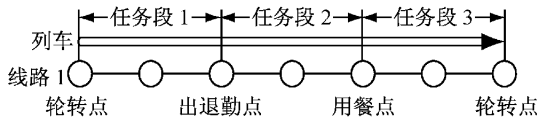


图 1 任务段示意图

Fig.1 Sketch of duty segment

多个任务段在满足特定约束的条件下可相互排列,并组合形成乘务任务,如图 2 所示,实线箭头为任务段,所有任务段组合形成 1 个乘务任务。乘务任务根据上班时间可分为早班、白班、夜班三类。合理的乘务任务必须满足如下乘务员值乘规则及劳动作业时间标准:

(1) 地点约束

乘务任务的出退勤、用餐和轮换必须在相应属性的驻点进行。

(2) 时间约束

乘务任务的出退勤必须满足相应的出退勤时间约束,如早班出勤时间不得晚于上午 8 点、早班退勤时间不得晚于上午 10:30 等。单次连续驾车时间不得超过最大连续驾车时间。单次轮换休息时间不得小于最小休息时间,不得大于最大休息时间。用餐时间必须在允许的用餐时段,并且不得小于最小用餐时间,不得大于最大用餐时间。

(3) 工作量均衡性约束

乘务任务需要保证乘务员工作量均衡,因此需约束最小工作量和最大工作量,一般可采用工作时间和驾驶里程来进行约束。



图 2 乘务任务示意图

Fig.2 Sketch of crew pairing

以上约束大多为范围约束,加之城轨发车间隔短,约束范围内任务段之间组合情况多,因此乘务任务数量巨大。一般任务段数量在 500~2 000,任务数量在 $10^4 \sim 10^8$ 。如何获取所有可能的组合任务,如何获取最优的乘务任务配对方案并提高求解效率是建模和设计算法的目标。

1.2 数学建模

城轨乘务任务配对是指将给定列车运行计划中所有任务段相互排列,并组合成若干乘务任务。最优的任务配对方案为方案中所有乘务任务总费用最小。本文以优化乘务任务配对,避免随乘情况为目标,建立了基于 SPP 模型的城轨乘务任务配对模型 (rail-crew pairing, R-CP),具体形式如下:

式(1)为目标函数,求解目标为所有乘务任务总费用最小。其中, c_j 为任务 j 的费用, m_j 为任务 j 早班对应的变量, d_j 为任务 j 白班对应的变量, n_j 为任务 j 夜班对应的变量。如果任务 j 被采用, m_j, d_j, n_j 为 1,反之为 0。 M 为早班集合, D 为白班集合, N 为夜班集合, S 为任务总集合。

$$\min \sum_{j \in M} c_j m_j + \sum_{j \in D} c_j d_j + \sum_{j \in N} c_j n_j, \quad M, D, N \in S \quad (1)$$

式(2)为乘务任务费用函数。其中, c_{base} 为执勤乘务任务的基本费用,为常数,此部分可尽量减少最优方案中乘务任务的数量; t_j 为任务 j 的总工作时间; α 为时间补贴系数,此部分可尽量减少非必要的休息时间,提高乘务员工作效率; r_j 为任务 j 的合理性变量,任务合理,则为 1,反之为 0; β 为任务不合理的惩罚值,此部分可保证任务的合理性。关于任务的合理性,后文会说明。

$$c_j = c_{base} + t_j \alpha + r_j \beta, \quad \forall j \in S \quad (2)$$

式(3)为乘务任务总工作时间 t_j 的计算方法。其中, t_{drive}^j 为任务 j 的总驾车时间, t_{break}^j 为任务 j 的总轮

转休息时间, $t_{\text{meal}}^{(j)}$ 为任务 j 的用餐时间, $t_{\text{on_duty}}^{(j)}$ 为任务 j 的出勤准备时间, $t_{\text{off_duty}}^{(j)}$ 为任务 j 的退勤时间.

$$t_j = t_{\text{drive}}^{(j)} + t_{\text{break}}^{(j)} + t_{\text{meal}}^{(j)} + t_{\text{on_duty}}^{(j)} + t_{\text{off_duty}}^{(j)}, \quad \forall j \in S \quad (3)$$

式(4)为集合分割约束, 该约束保证所有任务段被包含且仅包含于 1 个乘务任务中, 可避免随乘现象发生(文献[14]中为“ ≥ 1 ”). a_{ij} 为标识任务段 i 是否包含于任务 j 中, 如果是, 则 a_{ij} 为 1, 反之为 0. I 为任务段集合.

$$\sum_{j \in M} a_{ij} m_j + \sum_{j \in D} a_{ij} d_j + \sum_{j \in N} a_{ij} n_j = 1, \quad \forall i \in I, \forall j \in S \quad (4)$$

式(5)~(10)为出退勤时间约束. $t_{\text{start}}^{(j)}$ 为任务 j 的出勤时间, $t_{\text{m_early_start}}$ 为早班最早出勤时间(即运营开始时间), $t_{\text{m_late_start}}$ 为早班最晚出勤时间, $t_{\text{d_early_start}}$ 为白班最早出勤时间, $t_{\text{d_late_start}}$ 为白班最晚出勤时间, $t_{\text{n_early_start}}$ 为夜班最早出勤时间, $t_{\text{n_late_start}}$ 为夜班最晚出勤时间; $t_{\text{end}}^{(j)}$ 为任务 j 的退勤时间, $t_{\text{m_early_end}}$ 为早班最早退勤时间, $t_{\text{m_late_end}}$ 为早班最晚退勤时间, $t_{\text{d_early_end}}$ 为白班最早退勤时间, $t_{\text{d_late_end}}$ 为白班最晚退勤时间; $t_{\text{n_early_end}}$ 为夜班最早退勤时间, $t_{\text{n_late_end}}$ 为夜班最晚退勤时间(即运营结束时间).

$$t_{\text{m_early_start}} \leq t_{\text{start}}^{(j)} \leq t_{\text{m_late_start}}, \quad \forall j \in M, M \in S \quad (5)$$

$$t_{\text{d_early_start}} \leq t_{\text{start}}^{(j)} \leq t_{\text{d_late_start}}, \quad \forall j \in D, D \in S \quad (6)$$

$$t_{\text{n_early_start}} \leq t_{\text{start}}^{(j)} \leq t_{\text{n_late_start}}, \quad \forall j \in N, N \in S \quad (7)$$

$$t_{\text{m_early_end}} \leq t_{\text{end}}^{(j)} \leq t_{\text{m_late_end}}, \quad \forall j \in M, M \in S \quad (8)$$

$$t_{\text{d_early_end}} \leq t_{\text{end}}^{(j)} \leq t_{\text{d_late_end}}, \quad \forall j \in D, D \in S \quad (9)$$

$$t_{\text{n_early_end}} \leq t_{\text{end}}^{(j)} \leq t_{\text{n_late_end}}, \quad \forall j \in N, N \in S \quad (10)$$

式(11)为单次连续驾车时间约束. $t_{\text{condrive}, k}^{(j)}$ 为任务 j 的第 k 次连续驾车时间, Con_j 为任务 j 的连续驾车次数, t_{condrive} 为单次最大连续驾车时间.

$$t_{\text{condrive}, k}^{(j)} \leq t_{\text{condrive}}, \quad \forall j \in S, k = 1, 2, \dots, \text{Con}_j \quad (11)$$

式(12)~(14)为轮换、用餐时间约束. $t_{\text{break}, k}^{(j)}$ 为任务 j 的第 k 次轮换休息时间, R_j 为任务 j 的轮换休息次数, $t_{\text{min_break}}$ 为单次最小休息时间, $t_{\text{max_break}}$ 为单次最大休息时间; $t_{\text{meal_start}}^{(j)}$ 为任务 j 开始用餐时间, $t_{\text{meal_end}}^{(j)}$ 为任务 j 结束用餐时间; $t_{\text{b_min_meal}}$ 为最小用餐时间, $t_{\text{b_max_meal}}$ 为最大用餐时间, $t_{\text{b_meal_start}}$ 为用餐起始时间, $t_{\text{b_meal_end}}$ 为用餐结束时间, b 为班种属性, 包括早班(m)、白班(d)、夜班(n).

$$t_{\text{min_break}} \leq t_{\text{break}, k}^{(j)} \leq t_{\text{max_break}}, \quad \forall j \in S, k = 1, 2, \dots, R_j \quad (12)$$

$$t_{\text{b_min_meal}} \leq t_{\text{meal}}^{(j)} \leq t_{\text{b_max_meal}}, \quad \forall j \in S \quad (13)$$

$$t_{\text{b_meal_start}} \leq t_{\text{meal_start}}^{(j)} < t_{\text{meal_end}}^{(j)} \leq t_{\text{b_meal_end}}, \quad \forall j \in S \quad (14)$$

式(15)~(20)为工作量约束. $t_{\text{m_max}}$ 为早班最长工作时间, $t_{\text{m_min}}$ 为早班最短工作时间, $t_{\text{d_max}}$ 为白班最长工作时间, $t_{\text{d_min}}$ 为白班最短工作时间, $t_{\text{n_max}}$ 为夜班最长工作时间, $t_{\text{n_min}}$ 为夜班最短工作时间; $l_{\text{m_max}}$ 为早班最长驾驶里程, $l_{\text{m_min}}$ 为早班最短驾驶里程, $l_{\text{d_max}}$ 为白班最长驾驶里程, $l_{\text{d_min}}$ 为白班最短驾驶里程, $l_{\text{n_max}}$ 为夜班最长驾驶里程, $l_{\text{n_min}}$ 为夜班最短驾驶里程, l_j 为任务 j 的驾驶里程.

$$t_{\text{m_min}} \leq t_j \leq t_{\text{m_max}}, \quad \forall j \in M, M \in S \quad (15)$$

$$t_{\text{d_min}} \leq t_j \leq t_{\text{d_max}}, \quad \forall j \in D, D \in S \quad (16)$$

$$t_{\text{n_min}} \leq t_j \leq t_{\text{n_max}}, \quad \forall j \in N, N \in S \quad (17)$$

$$l_{\text{m_min}} \leq l_j \leq l_{\text{m_max}}, \quad \forall j \in M, M \in S \quad (18)$$

$$l_{\text{d_min}} \leq l_j \leq l_{\text{d_max}}, \quad \forall j \in D, D \in S \quad (19)$$

$$l_{\text{n_min}} \leq l_j \leq l_{\text{n_max}}, \quad \forall j \in N, N \in S \quad (20)$$

式(21)为地点约束. $p_{\text{start}}^{(j)}$ 为任务 j 的出勤地点, $p_{\text{end}}^{(j)}$ 为任务 j 的退勤地点, O 为出退勤地点集合; $p_{\text{dinner}}^{(j)}$ 为任务 j 的用餐地点, P 为用餐地点集合; $p_{\text{alter}}^{(j)}$ 为任务 j 的轮换地点, Q 为轮换地点集合.

$$p_{\text{start}}^{(j)}, p_{\text{end}}^{(j)} \in O, p_{\text{dinner}}^{(j)} \in P, p_{\text{alter}}^{(j)} \in Q, \quad \forall j \in S \quad (21)$$

式(22)为 0-1 约束.

$$m_j, d_j, n_j \in \{0, 1\}, \quad \forall j \in S \quad (22)$$

R-CP 模型具有以下特点: ① 约束为线性约束, 约束数量较多, 且要求变量为 0-1 变量, 因此 R-CP 模型为 0-1 规划模型. ② R-CP 模型的基础模型为 SPP 模型, 在数量级较大时, SPP 属于 NP-hard 问题[2]. 在求解实际问题时发现, R-CP 问题的变量数量集往往超过 10^4 , 因此 R-CP 同样属于 NP-Hard 问题. ③ R-CP 模型的约束存在等式约束, 采用启发式算法, 难以获取可行解.

2 算法设计

基于 R-CP 模型特点, 本文设计相应算法进行求解, 主要分为以下三步:

第 1 步 获取所有合理乘务任务集合 S .

第 2 步 将 R-CP 问题松弛为线性规划问题, 求解线性规划问题最优解.

第 3 步 分支定界获取全局最优整数解.

2.1 获取合理乘务任务集合 S

合理乘务任务指满足所有乘务值乘规则和劳动作业标准的任务, 即约束(5)~(21). 为搜寻所有合

理乘务任务,本文将任务段视为顶点,将任务段之间的衔接视为有向边,建立 $G=(V,E)$ 的任务段衔接关系有向图,如图 3 所示. V 为任务段顶点集合, E 为衔接边.

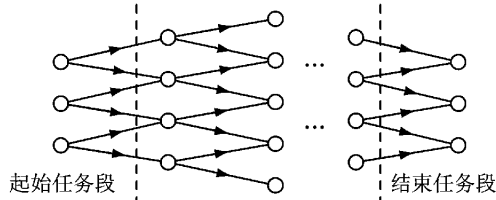


图 3 任务段衔接关系有向图

Fig.3 Directed graph of duty-segment connection

两个任务段顶点 V_1 和 V_2 之间存在衔接边,首先必须保证 V_1 的结束站与 V_2 的开始站相同,其次必须满足以下条件之一:① V_1 与 V_2 为同一列车,即乘务员可在轮换点不轮换,继续驾车;② V_1 与 V_2 为不同列车,在轮换点满足最小和最大休息时间,即约束(12);③ V_1 与 V_2 为不同列车,在用餐点满足用餐时段、最小和最大用餐时间,即约束(13)~(14).

图 3 中,起始任务段开始时间必须根据任务属性,满足特定出勤时间约束,如早班必须满足约束(5). 结束任务段结束时间须满足特定退勤时间约束,如早班必须满足约束(8).

根据以上要求,可依次产生早班、白班、夜班对应的任务段衔接关系有向图,根据有向图,采用基于广度搜索方法,结合劳动量约束(15)~(20),可搜索出满足要求的所有合理的早班任务集合 M 、白班任务集合 D 、夜班任务集合 N ,以及乘务任务总集合 $S=M+D+N$.

2.2 求解线性规划最优解

R-CP 模型为 0-1 规划模型,且变量多,直接求解难度较大. 本文首先求解 R-CP 问题的线性松弛问题(简称 LR-CP 问题),即将约束(22)松弛为约束(23). 在约束(23)中并没有考虑所有变量均 ≤ 1 ,原因在于,当所有变量非负时,如果任意变量 > 1 ,则式(4)中必有一个约束不成立. 因此只要保证所有变量非负,可行解中所有变量必 ≤ 1 .

$$m_j, d_j, n_j \geq 0, \quad \forall j \in S \quad (23)$$

线性规划问题采用单纯形算法可进行求解,然而乘务任务集合 S 的数量级往往超过 10^4 ,直接求解耗时较长,求解效率低.

列生成思想首先由 Gilmore 等^[15]提出,起初用于求解大规模装箱问题,后来被广泛应用于求解大型线性规划问题,该方法已经被证明为求解大型线性规划问题的高效算法.

本文基于列生成思想^[5],设计快速求解 LR-CP 问题的方法 (simplex algorithm based column generation, SABCG),求解步骤如下(求解示意图见图 4):

第 1 步 首先选取一个小集合乘务任务集 X ,形成以 X 为任务集合的新问题,称为主问题. 为保证初始 X 中存在可行解,可将每个任务段视为 1 个任务添加到 X 中(任务段作为任务可能导致部分任务不合理,但费用函数会对不合理任务进行惩罚,因此不会影响全局最优),然后再随机选取部分任务添加到 X 中,直到 X 集合中任务的数量达到要求.

第 2 步 采用单纯形算法求解主问题,获取其原问题最优解 s^* 和对偶问题最优解 u^* ,结合对偶最优解 u^* ,标定乘务任务 $j \in S \setminus X$ 的价格参数 $e_j(u)$. $e_j(u)$ 的计算公式如式(24),根据影子价格理论可知,若 $e_j(u)$ 为负值,将其添加到 X 中,可减小主问题的目标值,并且 $e_j(u)$ 越小,目标值减小越多.

$$e_j(u) = c_j - \sum_{i \in I_j} u_i \quad (24)$$

式中: I_j 为任务 j 包含的任务段 i 集合, u_i 为任务段 i 的最优对偶解.

第 3 步 保留主问题中的最优基变量,其他任务剔除,然后选择乘务任务 $j \in S \setminus X$ 中 $e_j(u)$ 为负值,且价格参数较小的任务添加到 X 中,当 X 集合中任务数量不够时,随机选取一些任务填充 X ,从而形成新的主问题.

第 4 步 如果不存在 $e_j(u)$ 为负的任务,则已达到最优解,终止;否则,回到第 2 步.

为提高主问题的单纯形算法求解效率,每次求解主问题时,可将上一轮迭代获取的主问题最优解作为本次单纯形算法的初始点,而无需从原点开始搜寻.

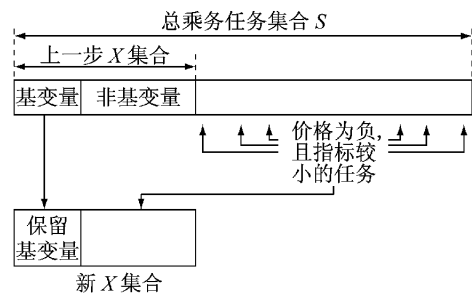


图 4 LR-CP 求解示意图

Fig.4 Solving process of LR-CP

由于 X 集合中的任务数量多少直接影响到迭代次数和一次迭代求解效率,因此其数量的确定,必须根据实际求解问题,找到一次迭代求解时间迅速变

大的拐点,既保证迭代次数不会太多,同时也保证一次迭代的求解速度较快.本文以某任务段数为 240,任务数为 480 的 LR-CP 问题为起点,不断增大任务数量,并统计计算时间,结果如图 5 所示.可以发现,当任务数量达到 600 左右时,其求解时间迅速增大,因此该问题 X 集合中任务的数量应取 600.

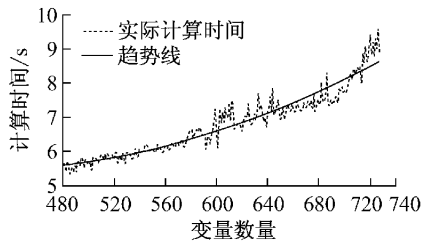


图 5 主问题单纯形求解时间

Fig.5 Time of solving master problem by simplex algorithm

2.3 获取整数解

基于列生成思想求解 LR-CP 问题,若最优解为整数解,则该最优解也是 R-CP 问题的最优解,若最优解不为整数解,则需采用分支定界算法进一步求取最优整数解.

传统的分支定界算法分支策略为:每次选取分数性最强的变量进行分支,针对 LR-CP 问题,则是变量最接近 0.5.如 $x_{11}=0.4932$,则下一步将 $x_{11}=1$ 和 $x_{11}=0$ 作为约束添加到 LR-CP 中,将问题分为两支,然后继续求解各分支最优解,直到最终找到最优整数解为止.

R-CP 问题的变量数量级一般超过 10^4 ,采用传统分支定界策略难以求解最优解. Ryan 等^[16]在求解 SPP 问题时提出了 1 种较好的分支策略,称为跟随分支策略(branch and follow-ons).

在 1 个任务 j 中,如果任务段 w 后紧随着任务段 z ,那么 z 称为 w 的跟随任务段. Ryan 等^[16]已经证明,如果 SPP 的线性松弛问题最优解中存在分数变量,那么总能找到这么 1 对任务段 w 和 z ,使得

$$0 < \sum_{k: a_{wk}=1, a_{zk}=1} y_k < 1, \quad \forall k \in S \quad (25)$$

式中: y_k 为任务变量, $\{k: a_{wk}=1, a_{zk}=1\}$ 指同时包含任务段 w 和 z 的任务. 跟随分支策略指找到满足式(25)的任务段对,然后基于该任务段对进行分支,如下所示:

$$\begin{aligned} \text{Branch0: } & \sum_{k: a_{wk}=1, a_{zk}=1} y_k = 0, \\ \text{Branch1: } & \sum_{k: a_{wk}=1, a_{zk}=1} y_k = 1 \end{aligned} \quad (26)$$

跟随分支策略应用于 R-CP 问题,其优点在于:

- ① 每次分支只需去掉不满足分支条件的任务,无需增加约束,同时也会减小问题复杂度;
- ② 每次分支可针对多个任务变量,分支效率较高.

本文基于跟随分支策略,获取 R-CP 问题的整数解.求解 R-CP 问题的算法(简称 CGBF 算法)总流程如图 6 所示.图 6 中 Upbound 为当前最优整数解的目标值(最优上界),Max 为无穷大实数,Opt 为当前最优解,Nodes 为分支节点集合,最优节点指由目标值最小的父节点所分支的节点. Nodes 的初始集合仅一个节点,即由所有乘务任务集合 S 所形成的 R-CP 问题.

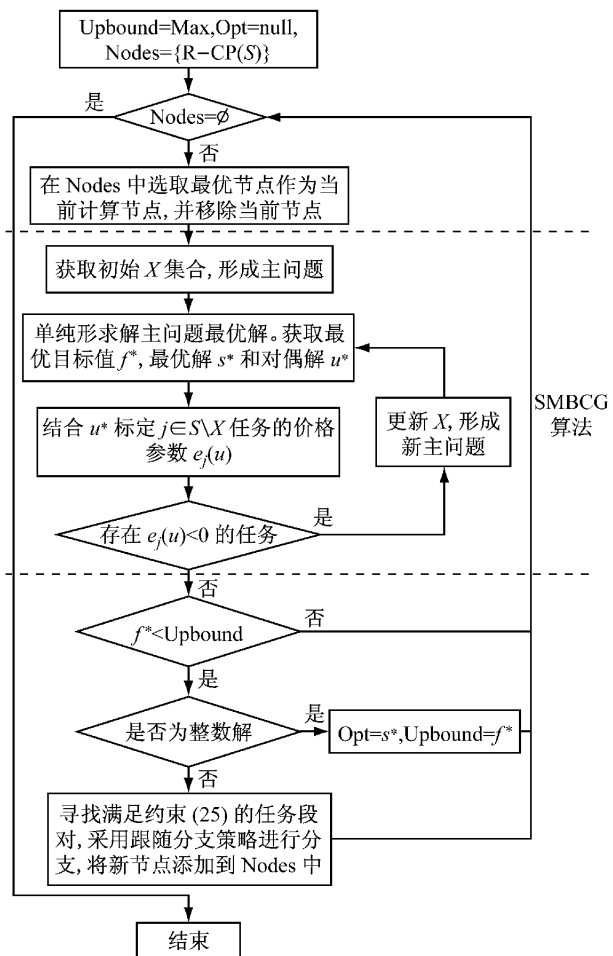


图 6 CGBF 算法总流程

Fig.6 Process of CGBF algorithm

3 算例验证

本文采用 VB.net 编程实现上述模型和算法,本文算例采用 CORE™ i5 2.5 GHz 主频,4G 内存的计算机运行.

以某地铁线路为例,如图 7 所示.该运营线路采用单一交路,高平峰间隔不同的运营模型.线路两端

各有一个车场,列车运行交路为“HDWLJ→CF”. 其中 CF 为折返站,HDWLJ 为出勤点,同时兼轮换点和用餐点.

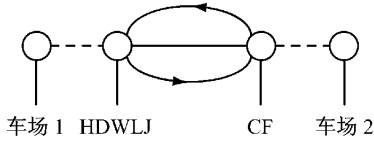


图 7 某线路基本情况示意图

Fig. 7 Sketch of a metro line

为对本文模型及算法进行验证,特选取该线路 PR1205,PR1302,SX1206,SX1301 四个版本列车运行图进行乘务任务配对测试,各版本运行图基本信息见表 1.

表 1 各运行图基本信息

Tab.1 Brief information of train diagrams

运行图版本	上线列车数/列	开行对数/对	列车行驶公里/km
PR1205	29	209	15 504
PR1302	32	212	15 744
SX1206	29	214	15 884
SX1301	24	188	13 880

该线路乘务劳动作息参数见表 2.

根据现场情况,费用函数采用如下形式:

$$c_j = 5 + t_j + 10^3 r_j, \quad \forall j \in S \quad (27)$$

其中,值乘基本费用为 5,任务不合理惩罚为 10^3 . 式(27)中没有考虑驾驶里程因素,原因在于,R-CP 问题可行解中所有任务的驾驶里程总和为定值(即运行图中列车总行驶公里),因此驾驶里程因素不会影响方案的优劣.

分别采用 CGBF 算法对各版本运行图进行乘务任务配对,并将获得的最优方案同传统分支定界算

表 2 乘务劳动作息参数

Tab.2 Crew work schedule standards

参数	数值	
出勤时间	早班最早出勤时间:运营开始	早班最晚出勤时间:8:00
	白班最早出勤时间:7:00	白班最晚出勤时间:15:00
	夜班最早出勤时间:14:00	夜班最晚出勤时间:19:30
退勤时间	早班最早退勤时间:7:00	早班最晚退勤时间:11:30
	白班最早退勤时间:13:30	白班最晚退勤时间:19:30
	夜班最早退勤时间:18:30	夜班最晚退勤时间:运营结束
轮换与用餐时间	轮换休息时间:15 min	最小用餐时间:20 min 最大用餐时间:45 min
	早班最小驾驶里程:62 km 早班最大驾驶里程:124 km 白班最小驾驶里程:62 km 白班最大驾驶里程:186 km(PR1302 为 248 km) 夜班最小驾驶里程:93 km 夜班最大驾驶里程:186 km	早班最小工作时间:无约束 早班最大工作时间:无约束 白班最小工作时间:无约束 白班最大工作时间:10 h 夜班最小工作时间:无约束 夜班最大工作时间:10 h

法(采用单纯形算法求解线性规划,采用传统分支思想求解整数解)以及现场人工编制结果进行对比,具体结果见表 3. 驾驶时间指任务的纯驾车时间,工作时间为任务从出勤到退勤的时间.

根据表 3 可以看出,CGBF 算法具有以下优点:

(1) CGBF 方案在任务数量上较少. 任务数量少,可节省乘务轮转人数,降低运营成本,提高经济效益.

表 3 3 种算法结果对比

Tab.3 Results contrast of three algorithms

版本	CGBF			现场人工编制结果			传统分支定界算法					
	任务数量/个	计算时间/min	驾驶时间/h	工作时间/h	任务数量/个	计算时间/d	驾驶时间/h	工作时间/h	任务数量/个	计算时间/min	驾驶时间/h	工作时间/h
PR1205	早班:29 白班:30 夜班:29	36	362.4	420.4	早班:31 白班:30 夜班:31	2	362.4	430.5	早班:29 白班:30 夜班:29	86	362.4	420.4
PR1302	早班:28 白班:24 夜班:28	52	368.1	420.8	早班:30 白班:25 夜班:30	2	368.1	468.5	早班:28 白班:24 夜班:28	121	368.1	420.8
SX1206	早班:29 白班:28 夜班:29	48	371.3	440.8	早班:31 白班:29 夜班:31	2	371.3	478.1	早班:29 白班:28 夜班:29	108	371.3	440.8
SX1301	早班:26 白班:24 夜班:26	29	324.5	393.0	早班:30 白班:25 夜班:30	2	324.5	421.9	早班:26 白班:24 夜班:26	79	324.5	393.0

(2) CGBF 方案总工作时间较少,工作效率较高.此指标说明 CGBF 方案在轮换地点衔接较好,任务中无效时间少,提高了乘务员运用效率.

(3) CGBF 算法计算效率高. CGBF 算法相对于传统分支定界算法,在取得相同最优解的同时,计算时间更短,计算效率更高.

(4) CGBF 方案避免了随乘情况. R-CP 模型的基础模型为 SPP 模型,要求每个任务段仅被覆盖一次,因此 CGBF 方案均没有随乘现象出现.

4 结语

本文以城轨乘务任务配对问题为研究对象,结合乘务员值乘规则和劳动作业时间标准(出退勤时间约束、用餐轮换时间约束、工作量均衡性约束等),并为避免随乘情况,建立了基于 SPP 模型的乘务任务配对模型(R-CP). R-CP 为 NP-Hard 问题,求解难度大.本文采用列生成思想,缩短了 R-CP 问题线性松弛问题的求解时间,设计了跟随分支策略,提高了分支效率,能够较为快速地搜寻最优方案.利用本模型和算法,开发了计算机程序,并进行了算例验证.结果表明:本文提出的模型与算法能够满足乘务任务配对方案的优化编制,具有编制效率高、目标函数更优(即任务数量少,任务效率更高)等特点.

我国城际高速铁路在乘务员排班及乘务值乘规则方面与城市轨道交通具有相似性,因此本文所提出的模型和算法对城际高速铁路的乘务计划编制同样具有参考价值.

参考文献:

- [1] Anbil R, Forrest J J, Pulleyblank W R. Column generation and the airline crew pairing problem [J]. *Documenta Mathematica*, 1998, 3(1): 677.
- [2] Wedelin D. An algorithm for large scale 0-1 integer programming with application to airline crew scheduling[J]. *Annals of Operations Research*, 1995, 57(1): 283.
- [3] AhmadBeygi S, Cohn A, Weir M. An integer programming approach to generating airline crew pairings[J]. *Computers & Operations Research*, 2009, 36(4): 1284.
- [4] Beasley J E. An algorithm for set covering problems [J]. *European Journal of Operational Research*, 1987, 31(1): 85.
- [5] Aydemir-Karadag A, Dengiz B, Bolat A. Crew pairing optimization based on hybrid approaches [J]. *Computers & Industrial Engineering*, 2013, 65(1): 87.
- [6] Souai N, Teghem J. Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem [J]. *European Journal of Operational Research*, 2009, 199(3): 674.
- [7] 夏平,马建军. 乘务员运用计划编制模型与计算方法的研究[J]. *铁路运输与经济*, 2008, 29(11): 27.
XIA Ping, MA Jianjun. Study on model and algorithm of crew working plan[J]. *Railway Transport and Economy*, 2008, 29(11): 27.
- [8] 程岩岩. 我国铁路乘务调度计划编制方法的研究与设计[D]. 北京: 北京交通大学, 2007.
CHENG Yanyan. Research and design of domestic railway crew scheduling method[D]. Beijing: Beijing Jiaotong University, 2007.
- [9] 赵鹏. 高速铁路动车组和乘务员运用的研究[D]. 北京: 北京交通大学, 1998.
ZHAO Peng. Study on passenger trains operation and crew scheduling in high speed railway[D]. Beijing: Beijing Jiaotong University, 1998.
- [10] 田志强. 高速铁路乘务计划编制优化理论与方法研究[D]. 成都: 西南交通大学, 2007.
TIAN Zhiqiang. Study on theory and methods of crew planning problem of high-speed railway [D]. Chengdu: Southwest Jiaotong University, 2007.
- [11] 王莹,刘军,苗建瑞. 客运专线乘务交路计划编制的优化模型与算法[J]. *铁道学报*, 2009, 31(1): 15.
WANG Ying, LIU Jun, MIAO Jianrui. Modeling and solving the crew scheduling problem of passenger dedicated line[J]. *Journal of the China Railway Society*, 2009, 31(1): 15.
- [12] 刘德利. 乘务派班及值乘方式的探讨[J]. *现代城市轨道交通*, 2009(1): 36.
LIU Deli. Discussion on crew dispatching and duty modes[J]. *Modern Urban Transit*, 2009(1): 36.
- [13] 李献忠,徐瑞华. 基于乘务广义费用的城市轨道交通排班[J]. *同济大学学报:自然科学版*, 2007, 35(6): 750.
LI Xianzhong, XU Ruihua. An optimal wide crew-related costs-based scheduling for crew of urban rail transportation[J]. *Journal of Tongji University: Natural Science*, 2007, 35(6): 750.
- [14] 石俊刚,史宏杰,徐瑞华. 城市轨道交通乘务任务划分模型及算法研究[J]. *铁道学报*, 2014, 36(5): 1.
SHI Jungang, SHI Hongjie, XU Ruihua. Modeling and solving urban rail transit crew pairing problems[J]. *Journal of the China Railway Society*, 2014, 36(5): 1.
- [15] Gilmore P C, Gomory R E. A linear programming approach to the cutting-stock problem[J]. *Operations Research*, 1961, 9(6): 849.
- [16] Ryan D M, Falkner J C. A bus crew scheduling system using a set partition model[J]. *Annals of Operations Research*, 1987, 4(1): 39.