# 含任务选择与并行任务分配的人员任务调度问题的自适应大邻域搜索

Martin Gutjahr、Sophie N. Parragh*和Fabien Tricoire

生产与物流管理研究所/*JKU*商学院，林茨约翰·开普勒大学，奥地利林茨

arXiv:2302.04494v1 [cs.DM] 9 Feb 2023

**摘要**

受一项现实应用的启发，我们对一个复杂的员工排班问题进行建模并求解。任务需要分配给工作人员进行监督。单名工作人员可以并行负责多项任务，且其班次可在可工作时段内灵活安排。每名工作人员拥有不同的技能组合，因此能够负责不同的任务。任务分配须符合优先级和技能要求。目标是在最大化按优先级加权的已分配任务数量的同时，尽量减少分配惩罚。我们设计了一种自适应大邻域搜索（ALNS）算法，并采用专门设计的破坏与修复算子。我们在基于现实数据构建的基准实例上测试该算法，并将其与使用商业混合整数规划求解器获得的最优结果进行比较。此外，我们还分析了采用三种额外备选目标函数的影响。将该算法应用于大规模公司数据时，其表现优于此前采用的求解方法。

## 1 引言

受体育比赛博彩赔率提供商 Sportradar 面临的现实问题启发，我们研究了一个复杂的人员排班问题。工作人员被分配执行监督任务。每名工作人员都具备一组特定技能，使其能够承担难度各异的不同任务。工作人员可以并行处理多项任务，其个人技能决定了可并行处理任务的数量上限。此外，每名工作人员都会提供自己的可工作时段，班次可安排在这些时段内。根据工作人员所在地的法律要求，每名工作人员都必须满足休息和休息时间方面的规定。休息可以分成若干段，但每次休息后恢复工作之前，都需要预留准备时间。任务数量超过了可承担的容量。每项任务都有相应的优先级评分，并对所属任务组提出特定的技能水平要求。目标是在最大化所有已分配任务的总优先级评分的同时，尽量减少因分配技能水平低于任务要求的操作员而产生的惩罚。我们针对所研究的问题提出了一种混合整数线性规划（MIP）模型，并开发了一种自适应大邻域搜索（ALNS）算法，以在较短时间内求解大规模实例。我们将启发式方法得到的解与使用 CPLEX 求解所开发混合整数规划模型得到的最优解进行比较。此外，我们还将所提出的 ALNS 应用于由

*sophie.parragh@jku.at

Sportradar 提供的大规模现实数据。我们指出，本文提出的思路并不局限于所研究的问题，也可用于其他人员排班、人员轮班和时刻表编制问题的求解方法。这些问题主要关注基于优先级分数进行任务选择和分配的要求。航空和铁路交通领域也存在类似问题，例如因普遍员工短缺或罢工而导致的航班取消，以及医疗分诊。

本文其余部分安排如下：第 2 节概述相关研究，第 3 节介绍所建立的数学模型。第 4 节详细描述所提出的自适应大邻域搜索算法。第 5 节和第 6 节给出计算结果，包括所提出的测试实例、与最优解及公司方案的比较，以及通过改变目标函数的权重得出的管理启示。第 7 节总结全文，并提出未来研究的可能方向。

## 2 相关研究

据我们所知，文献中尚未研究过本文提出的问题。不过，该问题具有若干研究较为充分的问题的特征。因此，我们分别回顾这些不同领域的文献，并指出所回顾的研究与本文所探讨问题之间的主要异同。

首先，本文提出的问题属于人员排班问题领域。该问题的主要目标是为可用操作员制定排班表，并为其分配已有开始时间和结束时间的任务，因此与航空业、（公共）交通运输业或医院中出现的员工排班问题有关。本文所讨论的问题与这些员工排班问题之间的一个主要区别

是可以并行执行多项任务。Chu（2007）例如提出了一种用于机场人员排班的目标规划方法，而 Maenhout 和 Van-houcke（2010）则为航空公司机组人员排班问题开发了一种混合散布搜索方法。

Yunes 等人（2005）研究了城市公交运输中的机组排班与人员排班问题，并通过混合列生成方法求解。Solos 等人（2013）例如采用变邻域搜索求解了一个考虑班次分配和休息时长的护士排班问题。与本文研究的问题不同，大多数人员排班研究都要求覆盖所有任务，其目标通常是尽量降低（分配）成本。不过，也有研究探讨其他目标，例如实现工作量目标方面的公平性（Prot 等人，2015）。

从方法论角度来看，Barrena 等人（2013）和 Dong 等人（2020）成功地将自适应大邻域搜索应用于列车排班与时刻表编制问题；Mansini 和 Zanotti（2020）则使用自适应大邻域搜索为医院病房中的医生排班，而 Sørensen 等人（2012）将其应用于高中课程表编制。

与本文所研究的问题相关的人员排班子领域之一是多技能问题。在这类问题中，每名工作人员都具备一组预定义技能，使其能够执行可用任务中的特定子集。例如，Cuevas 等人（2016年）提出了一种混合整数规划，用于解决短期多技能劳动力巡回排班问题，同时规划工作人员的班次和休息日，并将活动分配到各个班次。De Bruecker 等人（2018年）提出了一种三阶段混合整数规划方法，用于优化飞机维修人员的技能组合和培训计划。此外，Kovacs 等人（2012年）使用自适应大邻域搜索解决服务技术人员路径规划与排班问题，以最大限度地降低覆盖成本，同时考虑技术人员的技能水平。

本文所研究的问题还具有任务选择这一特征：需要从大量可用任务中

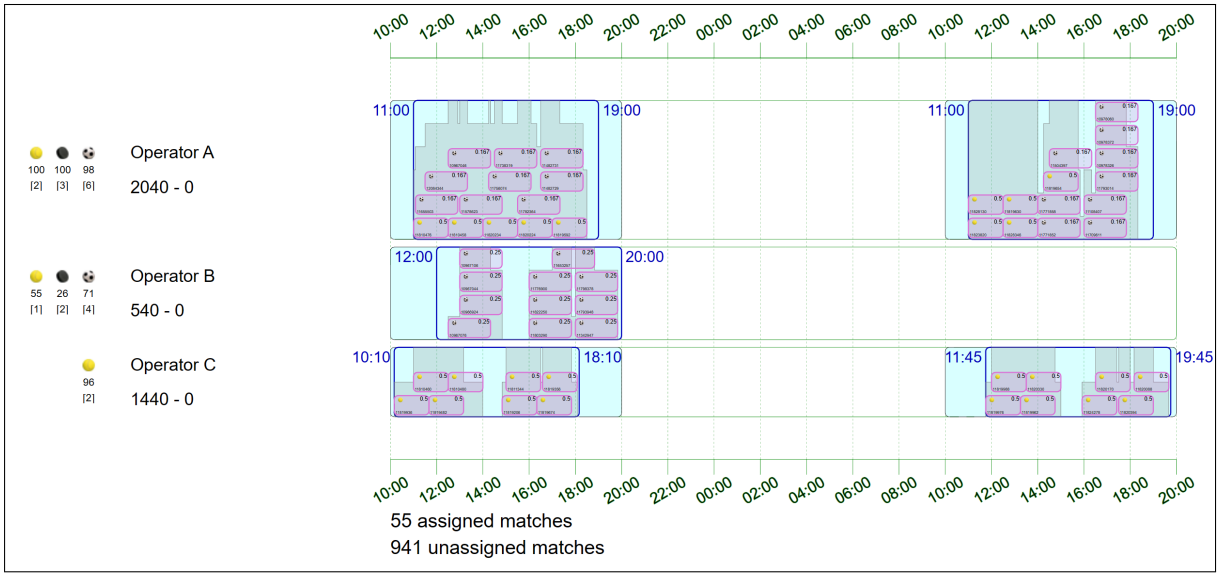
选择任务，以最大化已覆盖任务的总收益。尽管 Vijaykumar 和 Sohi（1998年）早在上个千年就研究了多标量处理器中的任务选择问题，但这一领域直到最近才受到更多关注。例如，Song 等人（2010年）研究了协作式云服务平台中的任务选择与分配问题，并结合自适应滤波器和启发式算法求解该问题。相关问题通常在众包背景下进行研究。Abououf 等人（2019年）使用遗传算法和禁忌搜索算法研究移动众包中的多工人多任务选择问题，而 Deng 等人（2016年）则借助动态规划和分支定界研究空间众包中的任务选择问题。Li 等人（2018b）提出了一种博弈论方法，用于求解食品配送中的任务选择与调度问题；Shaghaghi 和 Adve（2017年）研究了多功能雷达系统中的任务选择与调度问题，并利用分支定界将该问题求解至最优。

由于需要进行任务选择和分配，以构建包含最高价值任务的班次，因此该问题也可以视为装箱问题的一般化形式。早在20世纪70年代，就已有将调度问题重新表述为装箱问题的研究（Garey 等人，1976年；Coffman 等人，1978年）。Leinberger 等人（1999年）研究了多容量装箱在作业调度问题中的应用，Vijayakumar 等人（2013年）提出了一种双重装箱方法，用于安排医院中的手术病例。近年来，Li 和 Zhang（2018年）研究了具有二维装箱约束的单机装箱调度问题，Witteman 等人（2021年）则从装箱问题的角度研究了飞机维修任务分配。Su 等人（2021年）采用混合遗传方法，从装箱问题的角度研究了无关并行工作组调度问题。与本文所考虑的方法相关，自适应大邻域搜索已成功用于求解装箱问题。He 等人（2021年）使用自适应大邻域搜索求解圆形装箱问题；Zeng 和 Zhang（2021a）提出了一种用于具有二维装箱约束的单机批处理的自适应大邻域搜索方法，而 Zeng 和 Zhang（2021b）则将自适应大邻域搜索应用于带有冲突惩罚的二维装箱问题。

本文研究的问题也与资源受限项目调度问题（RCPSPs）有关，因为在将任务分配给工作人员时，必须考虑每位工作人员的容量限制。然而，与一般的 RCPSPs 不同，各项任务彼此独立，且其开始和结束时间固定。因此，目标是通过选择价值最高的任务，尽可能高效地利用可用资源（工人容量）。近年来，RCPSPs 相关研究例如探讨了资源平衡，以尽可能均衡地使用稀缺资源（Li 等人，2018a），以及灵活的资源配置，即资源使用量可随时间变化（Naber 和 Kolisch，2014）。Tao 和 Dong（2017）以及 Hauder 等人（2020）等研究考虑了备选活动链。Lova 等人（2000）研究了目标函数不同于通常的最小化总工期、且包含多个并行项目的 RCPSPs。与本文所提出方法相关的启发式方法已成功应用于 RCPSPs，例如 Palpant 等人（2004）以及 Muller（2009，2011）的研究。

根据 Van den Bergh 等人（2013年）对人员排班文献的综述，我们所研究的问题属于人员任务排程问题，其定义如下（Krishnamoorthy 和 Ernst，2001年）：必须将具有固定开始和结束时间的一组任务分配给具备不同资质的员工队伍，使其能够在预先规定的班次内仅执行部分可用任务。在班次最小化任务排程问题（SMTSP）中，目标是最小化员工总数。SMTSP 与区间图上的列表着色问题有关（Smet 等人，2014年），后者属于 NP 完全问题（Bonomo 等人，2009年）。我们引入了原始定义中未考虑的若干新方面，其中最重要的是所需的任务选择。因此，我们将所提出的问题称为

Fi图 1：玩具示例中任务分配给操作员的图示




带任务选择的人员任务调度问题（PTSP-TS）。PTSP-TS 对给定机器的区间调度问题进行了推广：在该问题中，必须将一组具有给定开始和结束时间的任务安排到给定的机器集合上，目标是最大化已分配任务的（加权）数量（Kolen 等人，2007年）。任务只能由部分机器执行的变体（与本文所考虑的技能要求相关）已知属于 NP 难问题（Kolen 等人，2007年；Arkin 和 Silverberg，1987年）。

## 3 问题定义

在 PTSP-TS 中，需要将任务分配给操作员，以最大化已覆盖任务的收益，同时尽量减少因分配给技能水平较低的操作员而产生的惩罚。图 1 展示了一个玩具示例的解，该示例使用了三名操作员和总计 996 项可分配任务。每名操作员都具备相应技能，可负责不同的任务。在我们的应用中，这些技能涉及对不同类型体育比赛的监督。在所考虑的 34 小时规划时域内，需要在五个可工作时段中安排班次。图中，每个可工作时段都用青绿色矩形表示，可工作时段内规划的班次则用蓝色边框标出。此外，我们还给出了班次的开始和结束时间。每个班次中的已分配任务均用紫色矩形表示，其长度代表任务的实际长度。矩形旁显示的浮点数表示任务所需的操作员注意力，后文称为所需带宽。班次中的灰色条表示操作员的带宽使用量，其高度随当前所需带宽总量而变化。灰色条达到班次边框顶部，表示带宽已全部占用。例如，第二个班次开始时，操作员 A 仅执行两项任务，此时带宽已满；而在班次接近结束时，则可行地安排六项并行任务。在每名操作员下方，我们列出了分配给该操作员的所有任务的总分配得分，以及因分配的任务低于该操作员所具备的技能水平而产生的总惩罚。在此玩具示例中，没有分配任何低于操作员技能水平的任务，因此该值为零。

为了更正式地陈述问题，我们定义一个任务集合 $V$ 和一个操作员集合 $O$。每项任务 $i$ 都有给定的优先级 $p_i$，以及已知的工作能力要求 $b_{is}$

（称为带宽），具体取决于其所分配到的班次 $s$ 。需要指出的是，我们并未对操作员本身进行显式建模，而是对操作员的班次进行建模；带宽要求取决于班次 $s$ 所属的操作员 $o$ 。集合 $S_o$ 包含操作员 $o$的各个班次。每项任务都要求特定的技能水平。将任务分配给技能水平较低的操作员的班次会产生$d_{is}$惩罚。完全不具备某项技能的操作员不能被分配需要相应技能的任务。每项任务都属于任务组集合 $P$中的一个任务组 $\sigma$ 。所有任务的集合 $V$ 由必须分配的必选任务集合 $U$ 和可选任务集合 $W$ （$V = U \cup W$）组成。

我们将时间视为离散的，并设可用的时间桶数量为 $|T|$。每位操作员定义一组可用时段（称为可工作时段）。每个可工作时段内最多只能安排一个班次。因此，每个班次均受其对应可工作时段的限制，该时段由最早开始时间 $e_s$ 和最晚结束时间 $l_s$ 给出； $T_s$ 提供了可以安排班次 $s$ 的时间桶集合。同样，每个任务 $i$ 也有一组其占用的时间桶 $R_i$ 。于是，集合 $V_{st}$ 给出了在时间桶 $t$ 中可以分配给班次 $s$ 的任务集合。此外，还规定了最小班次时长 $\alpha_s$ 和最大班次时长 $\omega_s$ 。对于每位操作员 $o$ ，每个班次都有特定要求：当无休息班次的最大时长 $\chi_s$ 超过时，必须安排时长为 $\gamma_o$ 的休息；操作员 $o$ 的班次之间还必须具有时长为 $\delta_o$ 的休息间隔。这意味着某些可工作时段可能不会安排任何班次。总休息时长可以通过多个较短的休息来满足。然而，每次部分休息都必须达到最短时长 $\beta_o$，具体取决于操作员 $o$；此外，每次单独的休息都需要时长为 $\rho$ 的准备时间，在此期间操作员不能执行任何任务。该准备时间不计入休息时长。每个班次还对休息作了进一步限制：只有在班次开始 $\nu_s$ 小时之后才能休息，且最迟只能在进入班次 $\lambda_s$ 小时之前安排休息。参数和变量所使用的完整符号列表见表 1。利用这些符号，我们将 PTSP-TS 表述为如下 0-1线性规划：

$$\max \quad w_1 \sum_{i \in V} \sum_{s \in S_i} p_i x_{is} - w_2 \sum_{i \in V} \sum_{s \in S_i} d_{is} x_{is} \tag{1}$$

满足以下约束：

$$\sum_{s \in S_i} x_{is} = 1 \qquad \forall i \in U \tag{2}$$

$$\sum_{s \in S_i} x_{is} \leq 1 \qquad \forall i \in W \tag{3}$$

$$x_{is} \leq z_s \qquad \forall i \in V, s \in S_i \tag{4}$$

$$\sum_{i \in V_{st}} x_{is} b_{is} \leq u_{st} \qquad \forall s \in S, t \in T_s \tag{5}$$

$$u_{st+1} \geq u_{st} \qquad \forall s \in S, t \in T_s, t \leq |T_s| - 1 \tag{6}$$

$$\sum_{t=e_s}^{l_s} u_{st} \leq z_s M \qquad \forall s \in S \tag{7}$$

$$r_{st+1} \geq r_{st} \qquad \forall s \in S, t \in T_s, t \leq |T_s| - 1 \tag{8}$$

$$\sum_{i \in V_{st}} x_{is} b_{is} \leq 1 - r_{st} \qquad \forall s \in S, t \in T_s \tag{9}$$

$$\sum_{t=e_s}^{l_s} r_{st} \leq z_s M \qquad \forall s \in S \tag{10}$$

表 1：符号说明

| 集合与参数 | |
|---|---|
| $T$ | 时间桶集合 |
| $U$ | 必选任务集合 |
| $W$ | 可选任务集合 |
| $V = U \cup W$ | 所有任务集合 |
| $O$ | 操作员集合 |
| $S$ | 班次集合 |
| $S_o$ | 操作员 $o$ 的班次集合 |
| $S_i$ | 可覆盖任务 $i$ 的班次集合 |
| $T_s$ | 班次 $s$ 可安排的可用时段中的时间桶集合 |
| $R_i$ | 任务 $i$ 的时间桶集合 |
| $V_{st}$ | 可分配到时间段 $t$ 中班次 $s$ 的任务集合 |
| $P$ | 任务组集合 |
| $[e_s, l_s]$ | 班次 $s$ 的最早开始时间和最晚结束时间（可用性） |
| $b_{is}$ | 任务 $i$ 在班次 $s$ 中所需的带宽 |
| $p_i$ | 任务 $i$ 的优先级评分 |
| $d_{is}$ | 若将任务 $i$ 分配到班次 $s$，其技能偏差 |
| $\alpha_s$ | 班次 $s$ 的最短时长 |
| $\omega_s$ | 班次 $s$ 的最长时长 |
| $\chi_s$ | 班次 $s$ 无需休息的最长时长 |
| $\rho$ | 准备时间 |
| $\beta_o$ | 操作员 $o$ 的最短部分休息时长 |
| $\gamma_o$ | 操作员 $o$ 的最短总休息时长 |
| $\delta_o$ | 操作员 $o$ 的最短休息时长 |
| $[\nu_s, \lambda_s]$ | 班次 $s$ 的休息时间窗口 |
| $w_1$,$w_2$ | 目标函数中的目标权重 |
| $k$ | 并行任务可涉及的不同组数上限 |
| 决策变量 | |
| $x_{is} = \begin{cases} 1, & \text{如果任务 } i \text{ 被分配到班次 } s \\ 0, & \text{否则} \end{cases}$ | |
| $u_{st} = \begin{cases} 1, & \text{如果班次 } s \text{ 在时间段 } t\text{内被激活} \\ 0, & \text{否则} \end{cases}$ | |
| $r_{st} = \begin{cases} 1, & \text{如果班次 } s \text{ 在时间段 } t\text{内被停用} \\ 0, & \text{否则} \end{cases}$ | |
| $v_{st} = \begin{cases} 1, & \text{如果班次 } s \text{ 中的时间段 } t \text{ 计为休息} \\ 0, & \text{否则} \end{cases}$ | |
| $a_{ts\sigma} = \begin{cases} 1, & \text{如果组 } \sigma \text{ 的任务被安排在班次 } s \text{ 的时间段 } t \\ 0, & \text{否则} \end{cases}$ | |
| $z_s = \begin{cases} 1, & \text{如果班次 } s \text{ 已启用} \\ 0, & \text{否则} \end{cases}$ | |
| $\hat{z}_s = \begin{cases} 1, & \text{如果班次 } s \text{ 需要休息} \\ 0, & \text{否则} \end{cases}$ | |

$$\sum_{t=e_s}^{l_s} (u_{st} - r_{st}) \geq \alpha_s z_s \qquad \forall s \in S \tag{11}$$

$$\sum_{t=e_s}^{l_s}(u_{st}-r_{st}) \le \chi_s z_s + (\omega_s-\chi_s)\hat{} z_s \qquad \forall s\in S \tag{12}$$

$$\sum_{t'=t+1}^{t+\rho}\sum_{i\in V_{st'}} x_{is}b_{is} \le \rho-\rho v_{st} \qquad \forall s\in S, t\in T_s, t\le l_s-\rho \tag{13}$$

$$\sum_{i\in V_{st}} x_{is}b_{is} \le 1-v_{st} \qquad \forall s\in S, t\in T_s \tag{14}$$

$$\sum_{t=e_s}^{l_s} v_{st} \ge \gamma_o\hat{} z_s \qquad \forall o\in O, s\in S_o \tag{15}$$

$$v_{st} \le u_{st}-r_{st} \qquad \forall s\in S, t\in T_s \tag{16}$$

$$v_{st} \le u_{s,t-\nu_s} \qquad \forall s\in S, t\in T_s, t\ge \nu_s \tag{17}$$

$$v_{st} \le 1-u_{s,\ t-\lambda_s+1} \qquad \forall s\in S,\ \ t\in T_s,\ \ t\ge\lambda_s \tag{18}$$

$$\sum_{t'=t+1}^{t+\beta_o-1} v_{st'} \le \beta_o\ (v_{st}+v_{s,\ t+\beta_o}) \qquad \forall o\in O,\ \ s\in S_o,\ \ t\in T_s,\ \ t\le|T_s|-\beta_o \tag{19}$$

$$a_{ts\sigma} \ge x_{is} \qquad \forall s\in S, i\in V, t\in R_i, \sigma\in P, \tag{20}$$

$$\sum_{\sigma\in P} a_{ts\sigma} \le k \qquad \forall s\in S, t\in T_s \tag{21}$$

$$\sum_{t'=t+1}^{t+\delta_o} u_{s't'} \le\ (1-u_{st}+r_{st})\ \delta_o \qquad \forall o\in O, s\in S_o, s'\in S_o|s'\ne s, t\in T_s \tag{22}$$

$$x_{is}\in\{0,1\} \qquad \forall i\in V,\ \ S\in S_i \tag{23}$$

$$u_{st}\in\{0,1\} \qquad \forall s\in S, t\in T_s \tag{24}$$

$$r_{st}\in\{0,1\} \qquad \forall s\in S, t\in T_s \tag{25}$$

$$v_{st}\in\{0,1\} \qquad \forall s\in S, t\in T_s \tag{26}$$

$$a_{ts\sigma}\in\{0,1\} \qquad \forall s\in S, t\in T_s, \sigma\in P \tag{27}$$

$$z_s\in\{0,1\} \qquad \forall s\in S \tag{28}$$

$$\hat{}z_s\in\{0,1\} \qquad \forall s\in S \tag{29}$$

目标函数 (1) 由两部分组成：所有已分配任务的优先级加权总和 $p_i$ ，以及由于将任务分配给技能水平低于相应任务组最优水平的操作员而产生的所有惩罚之和 $d_{is}$。总体目标函数最大化总优先级与总惩罚之间的差值，两者分别使用 $w_1$ 和 $w_2$ 加权。

第一组约束条件 (2)、(3) 和 (4) 确保任务分配符合限制。任务分为必选任务 $U$ 和可选任务 $W$。所有必选任务都必须且只能分配到一个班次，而其他任务则可以选择是否分配。此外，约束条件 (4) 要求，只有使用二元变量 $z_s$ 启用某个班次后，才能将任务分配到该班次。

约束条件 (5)、(6) 和 (7) 限定了二元变量 $u_{st}$ 的取值，该变量表示给定班次 $s$ 是否在时间段 $t$ 内处于激活状态。约束条件 (5) 给出了带宽使用量的上限。每项任务都需要占用一定量的操作员注意力，这一数值称为带宽 $b_{is}$。带宽取决于操作员的技能以及任务的复杂程度。只有当班次确实处于激活状态时，任意时刻使用的带宽才可以大于零。此外，由于时间被视为离散的，一旦某个班次被激活，后续时间段对应的二元变量

将一直为 1，直至最后一个时间段。最后，约束条件 (7) 确保必须使用 $z_s$ 启用整个班次，该班次才能被激活。如果操作员容量有富余，操作员的班次可以保持未启用状态，这意味着他或她不会在该班次工作。

与前四个约束类似，必须对班次停用变量 $r_{st}$ 施加约束。约束 (8) 确保班次 $s$ 一旦停用，就会保持停用状态。约束 (9) 规定，班次 $s$ 停用后，已用带宽上限设为零。约束 (10) 表示，只有班次 $s$ 已启用时，才能将其停用。约束 (11) 和 (12) 确保每个班次的最小和最大时长（$\alpha_s$ 和 $\omega_s$）均得到满足，并符合休息要求。

以下七个约束确保每个班次 $s$ 都包含规定的休息时间。约束 (13) 确保考虑每次休息后重新开始工作所需的准备时间。对于操作员 $o$，只有在某个时间段 $t$ 之后，准备时间 $\rho$内没有任何任务分配时，该时间段才能计入规定的休息时间 $\gamma_o$。约束 (14) 确保只有在带宽为零时，时间段 $t$ 才能计作休息。约束 (15) 要求当班次时长超过特定限值时，休息时间总和至少达到最短时长。接下来的三组约束（(16) 至 (18)）规定时间段何时才能计作休息。约束 (16) 要求只有在班次 $s$ 已开启但尚未关闭时，才能计入休息。约束 (17) 和 (18) 要求所有休息均符合最早休息开始时间 $\nu_s$ 和最晚休息结束时间 $\lambda_s$。最后，约束 (19) 要求每位操作员 $o$均满足所有最短休息时长要求 $\beta_o$。

此外，某个班次 $s$ 在某个时间段 $t$ 内允许覆盖的任务组数量 $P$ 受到限制，最多为 $k$。约束条件 (20) 要求，对于每个任务组 $\sigma$，如果该组至少有一项任务在该时段被分配，则任务组上限 $a_{ts\sigma}$ 应设为 1。约束条件 (21) 确保任意时刻设为 1 的任务组计数变量总数不超过 $k$ 的上限。

出于法律要求，同一操作员 $o$ 所覆盖的各班次之间必须保持休息时间 $\delta_o$。约束条件 (22) 确保每位操作员 $o$ 的所有班次 $S_o$ 之间均遵守这一休息时间要求。

变量的定义域在 (23)-(29) 中给出。

## 4 自适应大邻域搜索

为求解 PTSP-TS，我们设计了一种自适应大邻域搜索类算法。大邻域搜索（LNS）是一种元启发式搜索概念，由 Shaw（1998年）提出，与 Schrimpf 等人（2000年）提出的“破坏与重建”原则有关。随后，Ropke 和 Pisinger（2006年）将其进一步应用于路径规划与调度问题，并引入自适应层，根据修复与破坏算子的过往性能调整其选择概率。此后，它已成为路径规划与调度领域最受欢迎的启发式搜索方案之一（Pisinger 和 Ropke，2010年）。

ALNS 的工作流程如下（见算法1）。首先，生成一个可行解 $s$。每次迭代中，根据操作员的权重选择一个移除算子 $d$ 和一个修复算子 $r$，权重会随性能变化。接下来，使用所选移除算子 $d$ 从 $s$ 中移除任务，并将其放入未分配任务池。然后，使用所选修复算子 $r$将任务重新插入班次计划。如果生成的解 $s$符合接受准则，则用其替换 $s$。如果该解优于迄今为止找到的最优解，则用其替换 $s_{best}$。重复此过程，直到满足某个停止准则。

**算法1**自适应大邻域搜索

1: 生成一个初始解 $s$；2: 设置 $s_{best} := s$3: **重复**4: 选择一个移除算子 $d$ 和一个修复 $r$ 算子5: 将 $d$ 应用于 $s$，得到 $s'$6: 将 $r$ 应用于 $s'$，得到 $s''$7: 判断是否接受 $s''$ 作为新的当前最优解；如果是，则 $s \leftarrow s''$ 8: 检查 $s''$ 是否优于 $s_{best}$；如果是，则 $s_{best} \leftarrow s''$9: 更新操作员的得分和权重10: **直到**满足给定的停止准则11: 返回 $s_{best}$

下面将详细介绍各个组成部分：所采用的破坏启发式和修复启发式、所开发的随机化方法、接受机制以及自适应机制。

## 4.1 破坏启发式

我们采用了四种不同的移除启发式。其中三种启发式——随机移除、最差移除和算子移除——基于现有研究（例如 Kovacs 等，2012；Ropke 和 Pisinger，2006），并针对 PTSP-TS 的特点进行了调整。第四种操作称为基于时间的移除，它利用了我们的问题中任务具有固定的开始和结束时间这一特点。

### 4.1.1 随机移除

随机移除会随机选择已分配任务，将其从解中移除，并加入未分配任务池。其实现步骤如下：首先，从范围 chancelimits 中抽取一个随机数 $l$。然后，对每个当前已分配任务抽取一个随机数 $r$；如果 $r \leq l$，则将相应任务从排班表中移除。

### 4.1.2 最差移除

最差移除会彻底移除对目标函数贡献最低的班次。该方法专门用于选择那些产生了不必要高额惩罚的班次，以及分配密度较低的班次。被移除的班次数量按以下方式随机确定：从 chancelimits 中抽取一个随机数 $l$，并令 $N$ 表示当前计划班次的数量，然后从解中移除最差的 $Nl$ 个班次。

### 4.1.3 算子移除

算子移除算法会移除指定操作员的所有班次。该算法遵循与随机移除启发式算法相似的逻辑。它不会从解中移除单个任务，而是为每个操作员抽取一个随机数 $r$；如果满足条件 $r \leq l$（见第 4.1.1 节），则从解中移除相应操作员的所有班次。

### 4.1.4 基于时间的移除

基于时间的移除启发式算法会对整个解进行垂直切片。其目的是使任务能够在不同班次之间切换，以适应相似的时间段。该移除方法会随机选择一个开始时间 $t \in T$ 以及一个长度 $l$，其范围为 widthlimits,

移除所有满足 $R_i \cap \{t, ..., t+l\} \neq \emptyset$ 的已分配任务 $i$，也就是说，即使任务仅有部分落在该时间区间内，也会被移除。

## 4.2 修复启发式

所有修复启发式都依赖于根据不同标准对所有可能的任务到操作员的分配进行排序，然后按照该顺序执行所有可行的任务到操作员的分配。前三种启发式算法（A、B、C）依赖于基于惩罚和优先级的排序，下一种启发式算法（D）会考虑目标函数，最后一种启发式算法（E）则根据任务分配给操作员时所占用的时间和带宽来评估该分配的价值。以下将进一步详细介绍这些启发式算法。

### 4.2.1 惩罚和优先级排序

前三种修复启发式算法依赖于根据任务 $Priority\ p_i$和分配 $Penalty\ d_{is}$对所有任务到操作员的分配进行排序。我们采用多个按顺序执行的排序标准，以确保分配排序的确定性。我们使用 $Penalty$ 作为主要排序因素，以最大限度地增加分配给具有所需技能水平的操作员的任务数量。$Priority$用作第二排序值，优先选择优先级较高的分配。前两种修复启发式算法都首先按照 $Penalty$ 进行排序，然后依据 $Priority$，并引入以下额外的平局决胜项:

- $A$: $\{Penalty, Priority, skillFitting\}$
- $B$: $\{Penalty, Priority, skillFitting * (-1)\}$
- $C$: $\{Penalty, Priority, Bandwidth * (-1)\ \ skillFitting\}$

作为第一个额外的平局判定标准，我们使用一个称为 $skillFitting$ 的值。该值的计算取决于操作员是否满足所考虑任务的期望技能水平。如果满足，我们将 $skillFitting$ 的值计算为 $DesiredSkill - ActualSkill$。这样可以优先将任务分配给最接近期望技能要求的操作员，从而确保技能水平最高的操作员的容量留给最困难的任务。如果操作员的技能水平低于期望水平，则该值等于 $Penalty\ d_{is}$。对于修复算子 $A$ 和 $B$，我们将 $skillFitting$ 作为第三个平局判定标准；但是，对于操作员 $B$，该值取反。使用这一平局判定标准具有不同的目标：常规使用 $skillFitting$ 值旨在尽可能保留高技能操作员的容量，而使用取反后的值则旨在通过分配具有最高 $Priority$ 的任务来填满高技能操作员的班次。对于排序算子 $C$，我们引入另一个区分值 $Bandwidth$。$Bandwidth$ 是将任务分配到特定班次（上文记为 $b_{is}$）时该任务所占用的容量。由于我们的问题允许并行分配多个任务，因此该值的范围为从 1（满容量）到1 6。特定分配的 $Bandwidth$ 值取决于任务以及操作员用于并行覆盖的技能。由于通常更倾向于分配占用较少容量的任务分配，因此我们使用该值的倒数。
¯

### 4.2.2 目标排序

排序算子 $D$ 现在将 $Penalty$ 和 $Priority$ 的使用情况合并为值 $Objective = w_1 * p_i - w_2 * d_{is}$，该值就是一项分配对目标函数的实际贡献值。

- $D$: $\{Objective, Bandwidth, skillFitting\}$

与排序算子 $C$ 类似，我们使用 $Bandwidth$ 和 $skillFitting$ 作为后续的平局决胜项。

#### 4.2.3 效率排序

对于第五个也是最后一个排序算子 $E$，我们引入一个名为 $Efficiency$的新值。该值通过以下公式计算： $Efficiency = Objective$ /($Bandwidth * Length$)。$Efficiency$因此表示操作员每单位已占用容量的贡献。对于优先级相同的任务分配，操作员已占用的$Bandwidth$ 越低，且任务总体上越短，其$Efficiency$就越高。

- $E$: $\{Efficiency, Priority, Bandwidth*(-1)\ \ Groups*(-1)\}$

作为第一和第二平局判定标准，我们使用此前为 $Priority$ 和$Bandwidth$介绍的值。然而，由于 $Efficiency$值的结构，我们需要采用不同于 $skillFitting$的平局判定标准。在此，我们优先考虑灵活性最低、能够承担不同任务组的操作员。选作平局判定标准的值 $Groups$定义为操作员具备技能的不同任务组数量。此处不考虑技能水平。由于我们的目标是优先分配灵活性较低的员工，因此将 $Groups$的值取反。

### 4.3 初始解

为获得初始解，我们使用修复算子“效率排序”（见4.2.3）对实例中的所有任务进行排序。随后，按照排序机制给出的顺序，将任务插入最初为空的分配中，以此通过“修复”构造任务分配。这里只考虑可行任务分配。

### 4.4 接受准则

对于自适应大邻域搜索的实现，我们考虑了处理新生成解的三种可能方式。达到全局最优目标值的新解将立即被接受并存储，以供后续使用。如果新解的目标值至多比当前最优解低 .5%，则该解将被接受为下一次迭代的当前最优解。所有比当前解差 .5%以上的解都将被舍弃。在这种情况下，下一次迭代将使用当前最优解的任务分配开始。

### 4.5 破坏与修复启发式的选择

按照 Ropke 和 Pisinger（2006年）的方法，我们为每一种可能的修复启发式和破坏启发式 $\eta_h$ 分别设置权重 $h$。在每次自适应大邻域搜索迭代之前，根据各自的权重随机选择破坏启发式和修复启发式。每次迭代的结果（总体改进、临时改进和拒绝）都会记录到所选启发式算法中。经过给定次数的迭代 $SegLen$（称为分段）后，根据所有启发式算法在上一分段中的性能更新其权重。在后续分段中，使用这些更新后的值进行随机选择。搜索过程中会定期更新权重，每个分段的长度保持不变。某次迭代选择某一修复启发式的概率 $\phi_h$ 计算如下:

$$\phi_h = \frac{\eta_h}{\sum_{h'=1}^{n_h} \eta_{h'}} \tag{30}$$

破坏启发式的选择方式相同。

为在每个分段结束后更新启发式算法的权重，我们采用基于 Ropke 和 Pisinger（2006年）的方法。每种修复和破坏启发式算法都会获得一个初始得分

$\sigma$ 该得分会根据启发式算法的性能，在每次使用该算法进行迭代时递增。得分按如下方式更新：

$$\sigma = \begin{cases} \sigma + \epsilon_1, \text{如果 } s'' \text{改进了全局最优解 } s_{best}, \sigma + \epsilon_2 & s \quad \text{，如果 } s'' \\ & \\ \text{改进了当前最优解，} \sigma + \epsilon_3 & s \quad \text{，如果 } s'' \text{虽较差但仍被接受为新的当前} \\ \text{最优解} \sigma & \text{，否则。} \end{cases}$$

在我们的计算研究中，我们使用不同的参数设置来评估这一方法，并且还考虑保留 $\epsilon_3 = 0$; 这样一来，即使较差的解被接受，也不会获得奖励。每个分段共包含 100 次迭代。在每个分段结束时，使用以下公式更新每种破坏和修复启发式的权重：

$$\eta_h = (1 - p_{react})\eta_h + p_{react}\frac{\sigma_h}{\max(1, \Omega_h)} \tag{31}$$

其中，$\Omega_h$ 表示所考虑的修复启发式的使用次数， $p_{react}$ 是一个预设权重，用于控制新得分对历史权重的影响。

### 4.6 任务堆叠

为了提高带宽使用量，我们在修复启发式中引入了任务堆叠。 $Stacking$会使修复算法尝试将相似任务组成任务堆（种子任务即待插入的任务），并一次性插入所有堆叠任务。如果任务与种子任务具有相同的开始时间和结束时间，且属于同一任务组，就会被纳入堆叠考虑。由于可堆叠任务是并行分配的，理想情况下，所选任务能够充分利用操作员容量。

### 4.7 切片打乱

为了在修复启发式中引入一些随机化，我们设计了一种切片打乱机制。由于每次应用某个修复启发式时，所有可能的任务分配总是按相同方式排序，因此我们引入一种半随机打乱算子，以防算法反复生成相同的解。切片打乱会将已排序的可能任务分配列表按长度为 $\mu$划分为若干切片，然后打乱每个切片中任务分配的顺序。通过采用不同的参数设置进行初步实验，我们确定了集合 $sl$。集合 $sl$ 包含 $\mu$ 在给定迭代中可能取用的所有值：每个修复步骤开始时，都会从 $sl$ 中随机选取 $\mu$ 的值。确保切片不会过大，既能让我们探索任务分配的不同选项，又能避免修复启发式执行的初始排序失去意义。切片过小则可能产生非常相似的结果，因为可能有大量任务的排序值相同。

## 5 计算结果

ALNS 算法和混合整数规划模型均使用 Python 实现，其中混合整数规划模型采用 CPLEX 12.8 作为混合整数规划求解器。所有测试均在一个计算集群上进行，该集群有 64 个节点，每个节点配备 2 个 Intel Xeon X5570 处理器（共 4 个核心）以及 48 GB 内存。

每次运行均使用单线程。所有 ALNS 结果均基于每个实例的 10 次随机运行；混合整数规划求解器的运行时间上限为 6 小时。混合整数规划模型中的每个时间桶代表 5 分钟。

### 5.1 数据集

本文使用的测试实例基于合作伙伴公司 Sportradar 提供的信息以及真实世界的体育数据。Sportradar 的操作员分布在四个不同的国家，因此各国的工作时长和休息时长规定各不相同。无论工作地点在哪里，准备时间$\rho$ 均假定为 20 分钟。任务为体育比赛，属于 22 个不同任务组之一，所需技能水平最高可达 100。就带宽而言，一项任务可能需要占用操作员的全部带宽（1），也可能只需要其中一部分。最低值为$1\ 6$。这意味着最多可并行处理六项任务。此外，并行排班的任务最多只能来自两个不同的任务组。生成的测试实例分为四组，每组包含十个实例。小型实例仅包含少量工作人员，约有 100 项待分配任务，规划时域限于 8 小时。中型实例最多包含十名工作人员，任务数量在 400 至 700 项之间，总规划时域为 18 小时。大型实例包含更多操作员，旨在反映整天的任务分配，其规划时域为 24 小时，任务数量在 600 至 1100 项之间。最后，十个超大规模实例包含覆盖 72 小时时段的班次和任务，因此数据集包含 180 至 230 个班次以及 2000 至 3300 项任务。规划时域的长度大致符合 Sportradar 的需求，该公司会提前三天安排班次。实例名称采用以下格式：$|T|$ $|S|$ $|V|$ 表示实例的规划时域（小时）（$|T|$）、可用班次数量（$|S|$）以及可用任务数量（$|V|$）。

### 5.2 基准实例的结果

首先，我们评估所开发的自适应大邻域搜索在混合整数规划求解器也能处理的那些实例上的性能。

根据初步测试，所有自适应大邻域搜索参数均已设定。参数集为{chancelimits、widthlimits、$sl$、$\rho$、$\epsilon_1$、$\epsilon_2$、$\epsilon_3$、$p_{react}$}。用于定义随机移除算子中某个任务从解中被移除的概率，以及操作员的班次从解中被删除的概率的范围 chancelimits，设为 (0.1,0.5)。用于抽取基于时间的移除算子中切片长度取值的范围 widthlimits，设为 $[180,300]$（分钟）。集合 $sl$ 包含切片长度的所有可能取值，该切片长度 $\mu$ 用于切片重排步骤。该集合定义为 $\{0,4,8,16\}$。用于奖励表现良好的破坏与修复算子的参数 $\{\epsilon_1, \epsilon_2, \epsilon_3\}$ 初始设为 $\{40,25,8\}$。不过，接下来我们还将测试设置 $\{40,25,0\}$，即不奖励那些虽有所退化、但被接受为新的当前最优解的解。所有计算中，$p_{react}$ 的值均设为 0.1。目标函数中权重的基准设置为 $w_1 = 1$ 和 $w_2 = 1$。

表2报告了小型和中型实例的平均值、相对于MIP解的平均平均差距（AGap）以及平均最佳差距（BGap），涵盖设置$\epsilon_3 = 8$ 、设置 $\epsilon_3 = 0$ 以及不同的自适应大邻域搜索迭代上限（迭代次数）。

按单个实例划分的详细结果见附录中的表5至表12，其中表5至表8列出了设置 $\epsilon_3 = 8$ 的结果，表9至表12列出了设置$\epsilon_3 = 0$的结果。所有表格的结构如下：对于每个实例，我们给出平均目标值（$Avg$）、最优目标值（$Best$）、平均运行时间（$Time$，单位为秒），以及

表2：对于小型和中型实例，比较 ALNS 使用不同 $\epsilon_3$ 值时得到的平均解与混合整数规划求解器得到的最优解值。

| ALNS | | | | |
|---|---|---|---|---|
| 设置 | 迭代次数 | AGap | BGap | Avg |
| $\epsilon_3 = 8$ | 1,000 | 3.27 | 1.78 | 5123.4 |
| | 10,000 | 1.57 | 0.63 | 5226.2 |
| | 25,000 | 1.14 | 0.39 | 5252.5 |
| | 100,000 | 0.67 | 0.09 | 5286.1 |
| $\epsilon_3 = 0$ | 1,000 | 3.22 | 1.98 | 5127.2 |
| | 10,000 | 1.48 | 0.52 | 5231.7 |
| | 25,000 | 1.19 | 0.34 | 5249.8 |
| | 100,000 | 0.91 | 0.22 | 5269.2 |

自适应大邻域搜索（ALNS）计算的目标值与混合整数规划求解器所得目标值之间的平均差距（$AGap$）和最优差距（$BGap$），以百分比表示。此外，我们还给出平均分配任务数（$|M|$）。对于混合整数规划求解器获得的解，我们给出所得最优下界（$LB$）和运行时间（$Time$，单位为秒）。对于混合整数规划求解器未能在六小时时限内求解的实例，我们报告 CPLEX 在运行时间结束时给出的最优性差距（$MIPGap$）。最后一列给出混合整数规划求解器所得最优解中分配的任务数（$|M|$）。对于每组基准实例，我们分别在不同表格中列出迭代次数为1,000、10,000、25,000和100,000时的结果。每张表中还设有汇总行，给出该基准实例组中各实例平均差距的平均值，以及所达到的最优差距的平均值。

混合整数规划求解器得到的解（见表5；为便于比较，表6至表12中也重复列出）表明，对于规划时域为8小时的最小规模问题，大多数都能在时限内求得最优解。然而，即使是规划时域为18小时的中型实例，混合整数规划求解器也已达到其极限。对于规划时域为整整一天的大型实例，混合整数规划实现甚至在根节点就已无法求解。因此，在这组初步实验中，我们仅报告小型和中型实例的结果。

总体而言，设置 $\epsilon_3 = 8$ 的表现略优于设置 $\epsilon_3 = 0$。在比较找到的最优目标值时，这一差异主要体现在迭代次数最多的情况下。在测试过的最大迭代次数下，使用 $\epsilon_3 = 8$ 得到的最优解与混合整数规划求解器得到的解之间的差距不到 0.1%。不过，我们注意到，这一平均值包含了启发式算法显著优于混合整数规划解的情况，也包含了启发式算法略逊一筹的情况。启发式算法优于混合整数规划实现以及反之时，最大百分比差异约为 1.5%。我们还注意到，尽管混合整数规划求解器获准使用六小时来计算解，我们在最长时间的比较中选择了 100,000 次迭代的上限，使启发式算法的运行时间明显低于 30 分钟。我们还注意到，我们使用 Python 3 实现了该算法。改用其他编程语言时，速度可能提升 10 倍（Tricoire，2021）。下文报告的所有自适应大邻域搜索结果均采用设置 $\epsilon_3 = 8$。

### 5.3 附加目标与权衡分析

除了 PTSP-TS 中考虑的目标函数的两个组成部分外，Sportradar 还希望优化另外三个目标，以下段落将对这些目标进行详细介绍。我们将它们命名为 $Groups$、 $Consec$和 $Workload$。

任务可以根据相似性进行分组。在同一班次内执行相似任务时，服务质量会提高。反之，员工在一个班次中被分配到的不同组的任务越多，其所能提供的服务质量就越低。因此，第三个目标是尽量减少整个班次中分配的不同任务组数量，记为 $Groups$。虽然并发任务中不同组任务的数量存在上限，但该目标旨在减少整个班次中的不同任务组数量。

第四个目标引入了连续任务 $Consec$ 的概念，并在这些任务被安排在同一班次时给予奖励。由于某些任务组可能包含需要紧接着开始的任务，因此通常将这些连续任务安排在同一班次会有所帮助。在自适应大邻域搜索的基本设置中，并未直接利用这一信息。下文将说明如何将该信息纳入自适应大邻域搜索，以增加安排在同一班次中的连续任务数量。

最终目标是通过最大化工作量占总可用时间的百分比，尽量减少员工闲置。我们将这项指标命名为 $Workload$。本文提出的模型未纳入这一目标，因为通常认为，通过选择最有价值的任务即可实现优化。引入这一目标可能会促使模型分配优先级较低的任务，以减少员工闲置时间。

为了评估考虑这些附加目标所产生的影响，我们评估了不同的权重设置。令 $\{Priority,\ Penalty,\ Groups,\ Consec,\ Workload\}$ 表示所考虑目标的集合。我们将这些目标合并为加权和，并根据所选重点为每个目标赋予不同权重。Control 权重设置为基准设置，采用与上文相同的权重，即 $\{1,1,0,0,0\}$。Priority 权重设置仅关注第一个目标，使用权重集 $\{1,0,0,0,0\}$。Penalty 设置为第二个目标赋予相对较高的权重，使用权重集 $\{1,100,0,0,0\}$。TaskGroups 设置侧重于将相似任务分组，因此其权重为 $\{1,1,\text{-}10,0,0\}$。需要指出的是，$Groups$ 和 $Penalty$ 目标仅作为次要目标考虑，因为显然，如果不优化 $Priority$，那么对于不含不同组并行任务的解， $Groups$ 目标值将为 1；对于未分配任何任务的解，该值则为 0（可能达到的最优值）。对于 $Penalty$ 目标，后一种情况同样成立。最大化 $Workload$ 目标（Workload 设置）所采用的权重为 $\{1,1,0,0,10000\}$。

最后，我们分析一种设置：公司考虑的五个目标均被赋予权重，以大致反映它们从公司角度看来的重要性： $Priority$ 是主要目标， $Penalty$和 $Workload$是次要目标，而 $Groups$和 $Consec$则是最不重要的目标。在这项实验中，我们将后继匹配的概念引入所提出的自适应大邻域搜索（ALNS）：分配一场比赛时，我们增加了同时将其直接前驱/后继分配到同一班次的选项（若可行）。这是一项额外步骤，旨在增加分配给同一操作员的连续比赛数量。此外，在 ALNS 的破坏步骤中，我们引入后继破坏机制，允许破坏已被破坏比赛的任意已分配后继和前驱，即使它们在当前解中被分配到了不同班次。每次开始新的破坏步骤时，都会随机决定是否启用后继破坏。最后这些测试采用五个目标的权重 $\{1,1,\text{-}1,1,1000\}$ 进行。我们将加入这些额外要素的 ALNS 记为 ALNS+，并将该权重设置称为 AllObjectives。

我们在表3中总结了不同权重设置的结果。各

表3：采用不同权重设置，通过 100,000 次自适应大邻域搜索迭代得到的所有实例的平均解。对于 AllObjectives 设置，使用 ALNS+。

| | 平均单个目标值 | | | | | | |
|---|---|---|---|---|---|---|---|
| 权重设置 | $Priority$ | $Penalty$ | $Groups$ | $Consec$ | $Workload$ | $OF$ | $OFsmall$ |
| Control | 44733.6 | -206.55 | 2.34 | 11.05 | 0.86 | 44527.05 | 5286.10 |
| 优先级 | 44875.28 | -482.92 | 2.34 | 11.00 | 0.86 | 44392.35 | 5264.68 |
| 惩罚 | 43998.70 | -0.00 | 2.32 | 10.70 | 0.86 | 43998.70 | 5234.90 |
| TaskGroups | 44728.35 | -206.06 | 2.28 | 11.18 | 0.86 | 44522.29 | 5288.38 |
| 工作量 | 44576.25 | -195.04 | 2.38 | 10.89 | 0.88 | 44381.21 | 5168.35 |
| AllObjectives | 44751.4 | -202.16 | 2.32 | 23.55 | 0.87 | 44549.24 | 5288.87 |

实例的平均结果见附录中的表13–18。

在表3中，除了报告每个单独目标的平均值外，我们还报告了原始目标函数（$OF$）的平均值，以及仅考虑小型和中型实例时原始目标函数的平均值（$OFsmall$）。表中给出的数值表明，在所有经过测试的方法中，四种专门化设置各自在所考虑的目标上都取得了最优值。此外，仅考虑优先级和惩罚的 Control 设置，在所有测试实例中取得了原始目标函数的最佳平均值。仅考虑小型和中型实例时，对任务组最小化赋予额外权重（Taskgroups 设置）会对惩罚目标产生积极的副作用。

最后，AllObjectives 设置在原始目标函数方面平均而言取得了略好的解，这表明，着眼于连续比赛的额外要素通过将更多连续比赛安排在一起，帮助自适应大邻域搜索找到原始目标函数略优的解。

不同权重设置下得到的结果，使我们得以总结出一些管理启示，并为后续应用和实际落地提供参考。首先，这些结果表明，与两个主要目标相比，实现次要目标所需付出的代价相对较小。所有采用聚焦目标函数的结果，都显示聚焦目标相较于基准设置或 Control 设置有显著改善。

其次，加入侧重于连续比赛规划的额外因素，并为所有目标设置权重（AllObjectives 设置），会产生积极影响。该设置得到的解改善了全部三个次要目标，且两个主要目标的平均值也略有提升。这意味着在算法设计中应当利用并充分发挥额外信息的作用（在本研究中，额外信息是指有关哪些比赛可以且应当连续安排的信息）。

在下文的案例研究中，我们将使用部署于 AllObjectives 设置下的 ALNS+，在公司数据上测试所提出的算法，并将其与采用 Control 设置的 ALNS 进行比较。

# 6 案例研究

该问题由 Sportradar 提出。需要为操作员分配体育比赛，让他们进行监督，并判断提供的博彩赔率是否反映了实际情况。所考虑的问题包含前文介绍的全部五个目标。此外，Sportradar 要求一次性规划三天的时间范围。本节仅比较使用设置 AllObjectives 的 ALNS+ 所获得的解、

使用设置 Control 的 ALNS 所获得的解，以及该公司制定的任务分配。

表4：使用自适应大邻域搜索+ ，采用案例研究数据的 AllObjectives 权重设置所得平均解与原始公司分配的比较，以及仅对两个主要目标加权（Control）的自适应大邻域搜索结果

| 目标值 | 公司 | AllObjectives | Control |
|---|---|---|---|
| 总优先级（$Priority$） | 111390 | 179902.0 | 179865.0 |
| 总惩罚 ($Penalty$) | -10577 | -752.6 | -816.7 |
| 平均体育项目 ($Groups$) | 1.982 | 2.553 | 2.515 |
| 最大体育项目数 | 5 | 6.2 | 6.1 |
| 连续相同 ($Consec$) | 12 | 23.2 | 18.7 |
| 连续不同 | 16 | 82.4 | 82.9 |
| 连续安排 | 22 | 82.6 | 83.0 |
| 连续可用 | 83 | 83.0 | 83.0 |
| 工作时间与总时间之比 ($Workload$) | 0.822 | 0.841 | 0.841 |
| 已安排比赛 | 1584 | 2684.8 | 2696.6 |
| 比赛总数 | 5266 | 5266.0 | 5266.0 |
| 目标值 ($OF$) | 100813 | 179149.4 | 179048.3 |

作为比较我们的方法与原公司分配方案性能的依据，我们使用了涵盖五天的公司数据。由此得到的数据集共包含 6706 场比赛，可分配至 379 个潜在班次，这些班次属于 124 名操作员。为提高测试分配与原始分配之间的可比性，并模拟现实问题的连续性，我们选择不将这五天中第一天和最后一天的分配纳入目标函数。因此，在三天的时间范围内，共有 5266 场比赛可供分配。表4使用现实数据，将所提出的 ALNS+ 算法、ALNS 算法与公司所做的分配进行了比较。运行 ALNS 时仅考虑两个主要目标（权重设置为 Control），而 ALNS+ 则使用涵盖全部五个目标的权重（权重设置为 AllObjectives）。表中数值均是在两种权重设置下，迭代上限为 100,000 次时获得的。比较结果表明，若考察两个主要目标，使用所提出方法得到的分配方案显著优于公司以往的分配方案。第三个目标 $Groups$ 表明，与仅将其视为次要目标的公司方案相比，我们的 ALNS+ 分配了更多来自不同任务组的任务。所有班次中的最高值（“最大体育项目数”）也有同样的情况。对于分配连续比赛这一目标 $Consec$ 的比较表明，生成的分配方案中连续比赛的数量远多于公司方案。除了将连续比赛分配至同一班次这一实际目标外，我们还报告了分配至任意班次的连续比赛数值（“连续不同”），其中包括同一班次和不同班次；此外，还报告了至少有一场比赛被纳入整体方案的连续比赛数值（“连续安排”）。所报告的工作时间与总时间之比（$Workload$）表明，我们的分配方案平均达到

0.841，两种设置下均如此，显著高于公司分配方案的数值。最后，我们报告了方案中安排的比赛数量，以及 Control 设置下所用目标函数（$OF$）的取值。

综上所述，所提出的启发式算法在所有考量的性能指标中，除一项（$Groups$）外，均显著优于 Sportradar 制定的可比排班方案。前两个主要目标的目标值差距较大，部分原因在于各场比赛重要性尺度的上限。由于比赛的优先级通常设定在 0 到 100 之间，该公司的方案采用了直接实现这一目标的方法。因此，优先级较高的任务会尽可能早地安排。然而，对于某位工人而言，优先级较高但时长较长的比赛，可能不如优先级平均但时长较短的比赛，因为前者需要占用更多工作能力。比较第五个目标（即最小化工人空闲时间）以及各方案中已安排比赛的总数时，这一点体现得十分明显。我们的方案多安排了 1,000 多场比赛，但平均而言，工作时间占总时间的比例仅比公司方案高出约两个百分点。

## 7 结论

本文对带任务选择的人员任务调度问题（PTSP-TS）进行了建模并求解。我们开发了一种针对该问题定制的自适应大邻域搜索算法。我们对文献中已有的修复与破坏算子进行了调整，使其适用于所提出的问题。此外，我们还提出了新的定制算子。我们还为所研究的问题建立了数学模型，并将 ALNS 算法与使用 CPLEX 求解混合整数规划公式的结果进行了比较。即使对于中等规模的实例，CPLEX 所需的运行时间也很长，无法用于现实应用中所关注的时间范围。与 Sportradar 之前的规划方案相比，所提出的 ALNS 表现出很强的竞争力，目前已投入常规使用。

未来的研究将着重于进一步分析不同目标之间的权衡，例如，通过在多目标方法中同时考虑部分目标，从而得到帕累托前沿（或其近似）。

## 致谢

本研究全部或部分由奥地利科学基金（FWF） [P 31366]资助。为实现开放获取，作者已对本次投稿所产生的任何作者接受稿版本采用知识共享署名许可协议（CC BY）公共版权许可。我们还要感谢 Sportradar 为我们提供现实数据。

## 参考文献

Abououf, M., Mizouni, R., Singh, S., Otrok, H., and Ouali, A. (2019). 移动众包中的多工人多任务选择框架。《网络与计算机应用》, 130:52–62.

Arkin, E. M. 和 Silverberg, E. B. (1987). 具有固定开始和结束时间的作业调度。《离散应用数学》, 18(1):1–8. Barrena, E., Laporte, G., Coelho, L. C., 和 Ortiz, D. C. (2013). 一种快速高效的自适应

大邻域搜索启发式算法，用于解决具有动态需求的客运列车时刻表编制问题。技术报告 2013-64，CIRRELT。Bonomo, F., Dur´an, G., 和 Marenco, J. (2009). 探究着色与列表着色之间的复杂性边界。《运筹学年鉴》, 169(1):3–16. Chu, S. C. (2007). 轮班乘务职责的生成、调度和人员排班：在香港国际机场的应用。《欧洲运筹学杂志》, 177(3):1764–1778. Coffman, Jr, E. G., Garey, M. R., 和 Johnson, D. S. (1978). 装箱问题在多处理器调度中的应用。《*SIAM*计算杂志》, 7(1):1–17. Cuevas, R., Ferrer, J.-C., Klapp, M., 和 Mu˜noz, J.-C. (2016). 一种用于多技能员工队伍调度的混合整数规划方法。《调度学报》, 19(1):91–106. De Bruecker, P., Beli¨en, J., Van den Bergh, J., 和 Demeulemeester, E. (2018). 一种三阶段混合整数规划方法，用于优化飞机维护的技能组合与培训计划。《欧洲运筹学杂志》, 267(2):439–452. Deng, D., Shahabi, C., Demiryurek, U., 和 Zhu, L. (2016). 从工作者视角看空间众包中的任务选择。*GeoInformatica*, 20(3):529–568. Dong, X., Li, D., Yin, Y., Ding, S., 和 Cao, Z. (2020). 采用扩展自适应大邻域搜索元启发式方法，对通勤铁路的列车停站规划与时刻表编制进行集成优化。《交通研究*C*辑：新兴技术》, 117:102681. Garey, M. R., Graham, R. L., Johnson, D. S., 和 Yao, A. C.-C. (1976). 作为广义装箱问题的资源受限调度。《组合理论杂志*A*辑》, 21(3):257–298. Hauder, V. A., Beham, A., Raggl, S., Parragh, S. N., 和 Affenzeller, M. (2020). 具有活动和时间灵活性的资源受限多项目调度。《计算机与工业工程》, 150:106857. He, K., Tole, K., Ni, F., Yuan, Y., 和 Liao, L. (2021). 用于求解圆形装箱问题的自适应大邻域搜索。《计算机与运筹学研究》, 127:105140. Kolen, A. W., Lenstra, J. K., Papadimitriou, C. H., 和 Spieksma, F. C. (2007). 区间调度：综述。《海军研究后勤》*(NRL)*, 54(5):530–543. Kovacs, A. A., Parragh, S. N., Doerner, K. F., 和 Hartl, R. (2012). 服务技术人员路径规划与调度问题的自适应大邻域搜索。《调度学报》, 15:579– 600. Krishnamoorthy, M. 和 Ernst, A. T. ( 2001). 人员任务调度问题。载于《优化方法与应用》，第343–368页。施普林格。Leinberger, W., Karypis, G., 和 Kumar, V. (1999). 多容量装箱算法及其在多重约束下作业调度中的应用。载于《*1999*年国际并行处理会议论文集》，第404–412页。IEEE。Li, H., Xiong, L., Liu, Y., 和 Li, H. (2018a). 一种用于具有广义优先关系的资源平衡问题的有效遗传算法。《国际生产研究杂志》, 56(5):2054–2075.

Li, M., Zhang, J., and Wang, W.（2018b）。食品配送中的任务选择与调度：一种博弈论方法。载于*2018*年*IEEE*全球通信大会（*GLOBECOM*），第1–6页。IEEE。Li, X. and Zhang, K.（2018）。具有二维装箱约束的单批处理机调度。《国际生产经济学杂志》，196：113–121。Lova, A., Maroto, C., and Tormos, P.（2000）。一种改善多项目调度中资源分配的多准则启发式方法。《欧洲运筹学杂志》，127(2)：408–424。Maenhout, B. and Vanhoucke, M.（2010）。一种用于航空业个性化机组排班的混合散布搜索启发式算法。《欧洲运筹学杂志》，206(1)：155–167。Mansini, R. and Zanotti, R.（2020）。大型医院病房中的医生排班问题优化。《调度学报》，23(3)：337–361。Muller, L. F.（2009）。一种用于资源受限项目调度问题的自适应大邻域搜索算法。载于第八届元启发式算法国际会议（*MIC*）论文集。Muller, L. F.（2011）。一种用于多模式资源受限项目调度问题的自适应大邻域搜索算法。*DTU*管理工程，3：25。Naber, A. and Kolisch, R.（2014）。具有灵活资源配置的资源受限项目调度混合整数规划模型。《欧洲运筹学杂志》，239(2)：335–348。Palpant, M., Artigues, C., and Michelon, P.（2004）。LSSPER：用大邻域搜索求解资源受限项目调度问题。《运筹学年鉴》，131(1)：237–257。Pisinger, D. and Ropke, S.（2010）。大邻域搜索。载于《元启发式算法手册》，第399–419页。施普林格。Prot, D., Lap`egue, T., and Bellenguez-Morineau, O.（2015）。一种以公平性为目标的班次设计与人员任务调度问题两阶段方法。《国际生产研究杂志》，53(24)：7286–7298。Ropke, S. and Pisinger, D.（2006）。一种用于带时间窗取送货问题的大邻域搜索启发式。《交通科学》，40：455–472。Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., and Dueck, G.（2000）。运用破坏与重建原则取得破纪录的优化结果。《计算物理学杂志》，159(2)：139–171。Shaghaghi, M. and Adve, R. S.（2017）。多功能多通道雷达中的任务选择与调度。载于*2017*年*IEEE*雷达会议（*RadarConf*），第0969–0974页。IEEE。Shaw, P.（1998）。运用约束规划和局部搜索方法求解车辆路径问题。载于*CP-98*，第四届约束规划原理与实践国际会议。Smet, P., Wauters, T., Mihaylov, M., and Vanden Berghe, G.（2014）。班次数量最小化人员任务调度问题：一种新的混合方法及计算分析。《欧米伽》，46：64–73。

Solos, I. P., Tassopoulos, I. X., and Beligiannis, G. N.（2013）。一种用于有效求解护士排班问题的通用两阶段随机变邻域方法。《算法》，6(2)：278–308。Song, B., Hassan, M. M., and Huh, E.-n.（2010）。动态协作云服务平台中一种基于启发式算法的新型任务选择与分配框架。载于*2010*年*IEEE*第二届云计算技术与科学国际会议，第360–367页。IEEE。Sørensen, M., Kristiansen, S., and Stidsen, T. R.（2012）。2011年国际排课竞赛：一种自适应大邻域搜索算法。载于第九届自动排课实践与理论国际会议论文集（*PATAT 2012*），第489页。Su, B., Xie, N., and Yang, Y.（2021）。一种基于装箱策略的遗传算法，用于求解非相关并行工作组调度问题。《智能制造杂志》，32(4)：957–969。Tao, S. and Dong, Z. S.（2017）。具有备选活动链的资源受限项目问题调度。《计算机与工业工程》，114：288–296。Tricoire, F.（2021）。Broute：用于实现标准车辆路径算法的基准测试套件。*arXiv*预印本 *arXiv:2107.13492*。Van den Bergh, J., Beli¨en, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L.（2013）。人员排班：文献综述。《欧洲运筹学杂志》，226(3)：367–385。Vijayakumar, B., Parikh, P. J., Scott, R., Barnes, A., and Gallimore, J.（2013）。一种双重装箱方法，用于安排公立医院的手术病例。《欧洲运筹学杂志》，224(3)：583–591。Vijaykumar, T. and Sohi, G. S.（1998）。多标量处理器的任务选择。载于第*31*届*ACM/IEEE*微体系结构国际研讨会论文集，第81–92页。IEEE。Witteman, M., Deng, Q., a nd Santos, B. F.（2021）。一种用于求解飞机维修任务分配问题的装箱方法。《欧洲运筹学杂志》，294(1)：365–376。Yunes, T. H., Moura, A. V., and De Souza, C. C.（2005）。用于城市公共交通乘务员管理问题的混合列生成方法。《交通科学》，39(2)：273–288。Zeng, J. and Zhang, X.（2021a）。一种用于带二维矩形装箱约束的单机批处理调度的自适应大邻域搜索。《*IEEE*可靠性汇刊》。Zeng, J. and Zhang, X.（2021b）。一种用于带冲突惩罚的二维装箱问题的自适应大邻域搜索。载于*2021*年第八届可信系统及其应用国际会议（*DSA*），第12–19页。IEEE。

# 详细计算结果

表5：小型和中型实例的结果（1,000 次自适应大邻域搜索迭代，$\epsilon_3 = 8$）

| | LNS | | | | | | MIP | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 实例 | Avg | Best | Time | AGap | BGap | $\|M\|$ | MIP | Time | 混合整数规划间隙 | $\|M\|$ |
| 8_5_101 | 3400.6 | 3469.0 | 7.1 | 3.09 | 1.14 | 52.6 | 3509 | 942 | 0 | 44 |
| 8_4_160 | 4176.8 | 4229.0 | 8.2 | 2.09 | 0.87 | 60.6 | 4266 | 459 | 0 | 53 |
| 8_2_119 | 1449.0 | 1460.0 | 4.3 | 0.75 | 0.0 | 22.0 | 1460 | 0.5 | 0 | 22 |
| 8_5_72 | 2681.7 | 2704.0 | 6.0 | 1.91 | 1.1 | 35.0 | 2734 | 2 | 0 | 35 |
| 8_2_109 | 2307.0 | 2360.0 | 4.9 | 2.25 | 0.0 | 29.0 | 2360 | 14 | 0 | 30 |
| 8_4_172 | 4968.0 | 5060.0 | 7.9 | 5.37 | 3.62 | 57.0 | 5250 | 753 | 0 | 61 |
| 8_5_124 | 3430.0 | 3565.0 | 6.5 | 3.79 | 0.0 | 47.3 | 3565 | 82 | 0 | 40 |
| 8_5_145 | 4610.6 | 4656.0 | 8.0 | 2.32 | 1.36 | 66.4 | 4720 | 21600 | 3.97 | 69 |
| 8_5_80 | 2619.6 | 2689.0 | 6.3 | 3.62 | 1.07 | 33.1 | 2718 | 61 | 0 | 33 |
| 8_6_136 | 4234.5 | 4285.0 | 8.7 | 2.32 | 1.15 | 75.2 | 4335 | 21600 | 0.62 | 75 |
| 18_10_567 | 8970.2 | 9135.0 | 18.2 | 6.41 | 4.69 | 127.9 | 9585 | 21600 | 6.17 | 135 |
| 18_9_548 | 9775.0 | 9926.0 | 16.0 | 4.41 | 2.93 | 137.9 | 10226 | 21600 | 8.51 | 141 |
| 18_4_412 | 3344.0 | 3420.0 | 8.1 | 2.79 | 0.58 | 37.6 | 3440 | 12560 | 0 | 39 |
| 18_6_440 | 5945.6 | 6007.0 | 9.7 | 3.12 | 2.12 | 68.9 | 6137 | 21600 | 4.31 | 70 |
| 18_4_404 | 4798.0 | 4865.0 | 8.9 | 3.56 | 2.21 | 63.7 | 4975 | 21600 | 4.32 | 66 |
| 18_8_514 | 9003.0 | 9178.0 | 13.0 | 4.04 | 2.17 | 102.6 | 9382 | 21600 | 4.12 | 108 |
| 18_10_710 | 8569.5 | 8755.0 | 18.7 | 5.2 | 3.15 | 116.5 | 9040 | 21600 | 10.55 | 119 |
| 18_8_557 | 8202.4 | 8400.0 | 14.4 | 5.78 | 3.51 | 110.8 | 8706 | 21600 | 8.66 | 116 |
| 18_4_363 | 3333.0 | 3337.0 | 7.3 | 0.12 | 0.0 | 39.8 | 3337 | 43 | 0 | 40 |
| 18_9_368 | 6627.9 | 6677.0 | 11.1 | 2.44 | 1.72 | 81.0 | 6794 | 21600 | 7.85 | 81 |
| 平均值 | | | | 3.27 | 1.67 | | | | | |

表6：小型和中型实例的结果（10,000 次自适应大邻域搜索迭代，$\epsilon_3 = 8$）

| | LNS | | | | | | MIP | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 实例 | Avg | Best | Time | AGap | BGap | $\|M\|$ | MIP | Time | 混合整数规划间隙 | $\|M\|$ |
| 8_5_101 | 3448.9 | 3479.0 | 30.6 | 1.71 | 0.85 | 52.8 | 3509 | 942 | 0 | 44 |
| 8_4_160 | 4211.3 | 4229.0 | 33.7 | 1.28 | 0.87 | 60.5 | 4266 | 459 | 0 | 53 |
| 8_2_119 | 1459.0 | 1460.0 | 14.7 | 0.07 | 0.0 | 22.0 | 1460 | 0.5 | 0 | 22 |
| 8_5_72 | 2700.5 | 2734.0 | 24.0 | 1.23 | 0.0 | 35.1 | 2734 | 2 | 0 | 35 |
| 8_2_109 | 2332.0 | 2360.0 | 17.1 | 1.19 | 0.0 | 29.2 | 2360 | 14 | 0 | 30 |
| 8_4_172 | 5092.0 | 5230.0 | 37.2 | 3.01 | 0.38 | 58.8 | 5250 | 753 | 0 | 61 |
| 8_5_124 | 3532.8 | 3565.0 | 27.2 | 0.9 | 0.0 | 48.0 | 3565 | 82 | 0 | 40 |
| 8_5_145 | 4665.8 | 4680.0 | 34.5 | 1.15 | 0.85 | 67.3 | 4720 | 21600 | 3.97 | 69 |
| 8_5_80 | 2669.3 | 2705.0 | 24.1 | 1.79 | 0.48 | 33.3 | 2718 | 61 | 0 | 33 |
| 8_6_136 | 4288.5 | 4315.0 | 38.4 | 1.07 | 0.46 | 75.6 | 4335 | 21600 | 0.62 | 75 |
| 18_10_567 | 9301.9 | 9485.0 | 117.7 | 2.95 | 1.04 | 131.3 | 9585 | 21600 | 6.17 | 135 |
| 18_9_548 | 10001.4 | 10056.0 | 101.2 | 2.2 | 1.66 | 140.0 | 10226 | 21600 | 8.51 | 141 |
| 18_4_412 | 3422.0 | 3440.0 | 39.9 | 0.52 | 0.0 | 38.4 | 3440 | 12560 | 0 | 39 |
| 18_6_440 | 6019.3 | 6037.0 | 52.9 | 1.92 | 1.63 | 69.8 | 6137 | 21600 | 4.31 | 70 |
| 18_4_404 | 4864.0 | 4960.0 | 47.1 | 2.23 | 0.3 | 64.6 | 4975 | 21600 | 4.32 | 66 |
| 18_8_514 | 9197.6 | 9316.0 | 78.6 | 1.97 | 0.7 | 104.8 | 9382 | 21600 | 4.12 | 108 |
| 18_10_710 | 8804.4 | 8924.0 | 127.3 | 2.61 | 1.28 | 118.2 | 9040 | 21600 | 10.55 | 119 |
| 18_8_557 | 8441.9 | 8510.0 | 91.0 | 3.03 | 2.25 | 113.5 | 8706 | 21600 | 8.66 | 116 |
| 18_4_363 | 3337.0 | 3337.0 | 36.1 | 0.0 | 0.0 | 40.0 | 3337 | 43 | 0 | 40 |
| 18_9_368 | 6741.7 | 6817.0 | 66.1 | 0.77 | -0.34 | 81.3 | 6794 | 21600 | 7.85 | 81 |
| 平均值 | | | | 1.58 | 0.62 | | | | | |

Tab表 7：小型和中型实例的结果（25,000 次自适应大邻域搜索迭代，$\epsilon_3 = 8$）

| | LNS | | | | | | MIP | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 实例 | Avg | Best | Time | AGap | BGap | $\|M\|$ | MIP | Time | 混合整数规划间隙 | $\|M\|$ |
| 8_5_101 | 3466.0 | 3509.0 | 64.8 | 1.23 | 0.0 | 52.9 | 3509 | 942 | 0 | 44 |
| 8_4_160 | 4222.2 | 4229.0 | 70.6 | 1.03 | 0.87 | 60.2 | 4266 | 459 | 0 | 53 |
| 8_2_119 | 1460.0 | 1460.0 | 30.2 | 0.0 | 0.0 | 22.0 | 1460 | 0.5 | 0 | 22 |
| 8_5_72 | 2700.5 | 2734.0 | 49.3 | 1.23 | 0.0 | 35.1 | 2734 | 2 | 0 | 35 |
| 8_2_109 | 2337.0 | 2360.0 | 34.7 | 0.97 | 0.0 | 29.2 | 2360 | 14 | 0 | 30 |
| 8_4_172 | 5141.0 | 5250.0 | 81.8 | 2.08 | 0.0 | 59.3 | 5250 | 753 | 0 | 61 |
| 8_5_124 | 3558.0 | 3565.0 | 57.6 | 0.2 | 0.0 | 48.0 | 3565 | 82 | 0 | 40 |
| 8_5_145 | 4676.8 | 4680.0 | 71.8 | 0.92 | 0.85 | 67.2 | 4720 | 21600 | 3.97 | 69 |
| 8_5_80 | 2690.4 | 2706.0 | 49.2 | 1.02 | 0.44 | 33.2 | 2718 | 61 | 0 | 33 |
| 8_6_136 | 4303.5 | 4315.0 | 83.8 | 0.73 | 0.46 | 75.7 | 4335 | 21600 | 0.62 | 75 |
| 18_10_567 | 9337.9 | 9525.0 | 285.0 | 2.58 | 0.63 | 131.4 | 9585 | 21600 | 6.17 | 135 |
| 18_9_548 | 10104.8 | 10240.0 | 238.6 | 1.19 | -0.14 | 139.9 | 10226 | 21600 | 8.51 | 141 |
| 18_4_412 | 3422.0 | 3440.0 | 90.9 | 0.52 | 0.0 | 38.4 | 3440 | 12560 | 0 | 39 |
| 18_6_440 | 6029.3 | 6037.0 | 119.9 | 1.75 | 1.63 | 70.0 | 6137 | 21600 | 4.31 | 70 |
| 18_4_404 | 4891.5 | 4960.0 | 105.8 | 1.68 | 0.3 | 64.7 | 4975 | 21600 | 4.32 | 66 |
| 18_8_514 | 9262.4 | 9332.0 | 178.8 | 1.27 | 0.53 | 106.2 | 9382 | 21600 | 4.12 | 108 |
| 18_10_710 | 8884.8 | 9010.0 | 307.7 | 1.72 | 0.33 | 119.9 | 9040 | 21600 | 10.55 | 119 |
| 18_8_557 | 8499.0 | 8560.0 | 215.3 | 2.38 | 1.68 | 113.7 | 8706 | 21600 | 8.66 | 116 |
| 18_4_363 | 3337.0 | 3337.0 | 80.6 | 0.0 | 0.0 | 40.0 | 3337 | 43 | 0 | 40 |
| 18_9_368 | 6765.9 | 6819.0 | 155.1 | 0.41 | -0.37 | 81.8 | 6794 | 21600 | 7.85 | 81 |
| 平均值 | | | | 1.15 | 0.36 | | | | | |

Tab表 8：小型和中型实例的结果（100,000 次自适应大邻域搜索迭代，$\epsilon_3 = 8$）

| | LNS | | | | | | MIP | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 实例 | Avg | Best | Time | AGap | BGap | $\|M\|$ | MIP | Time | 混合整数规划间隙 | $\|M\|$ |
| 8_5_101 | 3491.7 | 3509.0 | 233.9 | 0.49 | 0.0 | 52.6 | 3509 | 942 | 0 | 44 |
| 8_4_160 | 4229.0 | 4229.0 | 256.4 | 0.87 | 0.87 | 60.0 | 4266 | 459 | 0 | 53 |
| 8_2_119 | 1460.0 | 1460.0 | 100.2 | 0.0 | 0.0 | 22.0 | 1460 | 0.5 | 0 | 22 |
| 8_5_72 | 2705.2 | 2734.0 | 172.6 | 1.05 | 0.0 | 35.2 | 2734 | 2 | 0 | 35 |
| 8_2_109 | 2348.0 | 2360.0 | 123.0 | 0.51 | 0.0 | 29.2 | 2360 | 14 | 0 | 30 |
| 8_4_172 | 5218.0 | 5250.0 | 288.9 | 0.61 | 0.0 | 60.6 | 5250 | 753 | 0 | 61 |
| 8_5_124 | 3565.0 | 3565.0 | 207.6 | 0.0 | 0.0 | 48.0 | 3565 | 82 | 0 | 40 |
| 8_5_145 | 4678.4 | 4680.0 | 249.2 | 0.88 | 0.85 | 67.4 | 4720 | 21600 | 3.97 | 69 |
| 8_5_80 | 2700.7 | 2707.0 | 176.2 | 0.64 | 0.4 | 33.2 | 2718 | 61 | 0 | 33 |
| 8_6_136 | 4315.0 | 4315.0 | 309.5 | 0.46 | 0.46 | 76.0 | 4335 | 21600 | 0.62 | 75 |
| 18_10_567 | 9411.8 | 9525.0 | 1077.5 | 1.81 | 0.63 | 132.2 | 9585 | 21600 | 6.17 | 135 |
| 18_9_548 | 10168.4 | 10280.0 | 952.7 | 0.56 | -0.53 | 142.1 | 10226 | 21600 | 8.51 | 141 |
| 18_4_412 | 3428.0 | 3440.0 | 331.1 | 0.35 | 0.0 | 38.4 | 3440 | 12560 | 0 | 39 |
| 18_6_440 | 6049.4 | 6137.0 | 453.6 | 1.43 | 0.0 | 70.1 | 6137 | 21600 | 4.31 | 70 |
| 18_4_404 | 4924.0 | 4980.0 | 391.3 | 1.03 | -0.1 | 65.1 | 4975 | 21600 | 4.32 | 66 |
| 18_8_514 | 9312.4 | 9352.0 | 644.7 | 0.74 | 0.32 | 106.5 | 9382 | 21600 | 4.12 | 108 |
| 18_10_710 | 9011.6 | 9130.0 | 1168.4 | 0.31 | -1.0 | 120.6 | 9040 | 21600 | 10.55 | 119 |
| 18_8_557 | 8587.4 | 8620.0 | 825.5 | 1.36 | 0.99 | 114.1 | 8706 | 21600 | 8.66 | 116 |
| 18_4_363 | 3337.0 | 3337.0 | 301.6 | 0.0 | 0.0 | 40.0 | 3337 | 43 | 0 | 40 |
| 18_9_368 | 6792.0 | 6849.0 | 590.2 | 0.03 | -0.81 | 81.8 | 6794 | 21600 | 7.85 | 81 |
| 平均值 | | | | 0.66 | 0.1 | | | | | |

Ta表 9：小型和中型实例的结果（1,000 次自适应大邻域搜索迭代，$\epsilon_3 = 0$）

| | LNS | | | | | | MIP | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 实例 | Avg | Best | Time | AGap | BGap | $\|M\|$ | MIP | Time | 混合整数规划间隙 | $\|M\|$ |
| 8_5_101 | 3405.4 | 3429.0 | 7.4 | 2.95 | 2.28 | 51.7 | 3509 | 942 | 0 | 44 |
| 8_4_160 | 4165.0 | 4200.0 | 7.7 | 2.37 | 1.55 | 61.1 | 4266 | 459 | 0 | 53 |
| 8_2_119 | 1454.0 | 1460.0 | 4.7 | 0.41 | 0.0 | 22.0 | 1460 | 0.5 | 0 | 22 |
| 8_5_72 | 2673.7 | 2731.0 | 6.4 | 2.21 | 0.11 | 35.0 | 2734 | 2 | 0 | 35 |
| 8_2_109 | 2308.0 | 2350.0 | 5.1 | 2.2 | 0.42 | 28.9 | 2360 | 14 | 0 | 30 |
| 8_4_172 | 4990.0 | 5120.0 | 8.1 | 4.95 | 2.48 | 57.5 | 5250 | 753 | 0 | 61 |
| 8_5_124 | 3455.0 | 3565.0 | 6.8 | 3.09 | 0.0 | 47.6 | 3565 | 82 | 0 | 40 |
| 8_5_145 | 4605.0 | 4650.0 | 8.3 | 2.44 | 1.48 | 66.2 | 4720 | 21600 | 3.97 | 69 |
| 8_5_80 | 2630.4 | 2667.0 | 6.5 | 3.22 | 1.88 | 33.1 | 2718 | 61 | 0 | 33 |
| 8_6_136 | 4238.0 | 4295.0 | 8.9 | 2.24 | 0.92 | 76.0 | 4335 | 21600 | 0.62 | 75 |
| 18_10_567 | 8988.9 | 9085.0 | 18.2 | 6.22 | 5.22 | 127.2 | 9585 | 21600 | 6.17 | 135 |
| 18_9_548 | 9802.4 | 9912.0 | 16.1 | 4.14 | 3.07 | 137.8 | 10226 | 21600 | 8.51 | 141 |
| 18_4_412 | 3369.0 | 3420.0 | 8.0 | 2.06 | 0.58 | 37.9 | 3440 | 12560 | 0 | 39 |
| 18_6_440 | 5935.1 | 6007.0 | 9.8 | 3.29 | 2.12 | 69.1 | 6137 | 21600 | 4.31 | 70 |
| 18_4_404 | 4818.5 | 4870.0 | 9.0 | 3.15 | 2.11 | 63.5 | 4975 | 21600 | 4.32 | 66 |
| 18_8_514 | 8957.4 | 9150.0 | 12.8 | 4.53 | 2.47 | 102.0 | 9382 | 21600 | 4.12 | 108 |
| 18_10_710 | 8600.9 | 8758.0 | 18.7 | 4.86 | 3.12 | 116.7 | 9040 | 21600 | 10.55 | 119 |
| 18_8_557 | 8230.5 | 8330.0 | 14.4 | 5.46 | 4.32 | 110.7 | 8706 | 21600 | 8.66 | 116 |
| 18_4_363 | 3337.0 | 3337.0 | 7.3 | 0.0 | 0.0 | 40.0 | 3337 | 43 | 0 | 40 |
| 18_9_368 | 6622.7 | 6695.0 | 11.2 | 2.52 | 1.46 | 80.6 | 6794 | 21600 | 7.85 | 81 |
| 平均值 | | | | 3.12 | 1.78 | | | | | |

Tab表 10：小型和中型实例的结果（10,000 次自适应大邻域搜索迭代，$\epsilon_3 = 0$）

| | LNS | | | | | | MIP | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 实例 | Avg | Best | Time | AGap | BGap | $\|M\|$ | MIP | Time | 混合整数规划间隙 | $\|M\|$ |
| 8_5_101 | 3457.4 | 3499.0 | 30.4 | 1.47 | 0.28 | 52.8 | 3509 | 942 | 0 | 44 |
| 8_4_160 | 4200.9 | 4229.0 | 35.0 | 1.53 | 0.87 | 61.1 | 4266 | 459 | 0 | 53 |
| 8_2_119 | 1460.0 | 1460.0 | 16.3 | 0.0 | 0.0 | 22.0 | 1460 | 0.5 | 0 | 22 |
| 8_5_72 | 2691.4 | 2731.0 | 24.2 | 1.56 | 0.11 | 35.0 | 2734 | 2 | 0 | 35 |
| 8_2_109 | 2337.0 | 2360.0 | 19.7 | 0.97 | 0.0 | 29.2 | 2360 | 14 | 0 | 30 |
| 8_4_172 | 5111.0 | 5250.0 | 37.5 | 2.65 | 0.0 | 59.1 | 5250 | 753 | 0 | 61 |
| 8_5_124 | 3531.0 | 3565.0 | 26.4 | 0.95 | 0.0 | 48.2 | 3565 | 82 | 0 | 40 |
| 8_5_145 | 4642.8 | 4680.0 | 34.9 | 1.64 | 0.85 | 67.2 | 4720 | 21600 | 3.97 | 69 |
| 8_5_80 | 2679.9 | 2707.0 | 24.4 | 1.4 | 0.4 | 33.4 | 2718 | 61 | 0 | 33 |
| 8_6_136 | 4295.0 | 4315.0 | 40.2 | 0.92 | 0.46 | 75.3 | 4335 | 21600 | 0.62 | 75 |
| 18_10_567 | 9247.3 | 9395.0 | 120.7 | 3.52 | 1.98 | 130.0 | 9585 | 21600 | 6.17 | 135 |
| 18_9_548 | 10034.9 | 10102.0 | 103.5 | 1.87 | 1.21 | 139.9 | 10226 | 21600 | 8.51 | 141 |
| 18_4_412 | 3420.0 | 3440.0 | 40.5 | 0.58 | 0.0 | 38.4 | 3440 | 12560 | 0 | 39 |
| 18_6_440 | 6031.0 | 6107.0 | 52.2 | 1.73 | 0.49 | 70.1 | 6137 | 21600 | 4.31 | 70 |
| 18_4_404 | 4894.0 | 4975.0 | 46.9 | 1.63 | 0.0 | 64.3 | 4975 | 21600 | 4.32 | 66 |
| 18_8_514 | 9209.4 | 9338.0 | 76.8 | 1.84 | 0.47 | 106.2 | 9382 | 21600 | 4.12 | 108 |
| 18_10_710 | 8837.9 | 9005.0 | 128.0 | 2.24 | 0.39 | 119.5 | 9040 | 21600 | 10.55 | 119 |
| 18_8_557 | 8499.4 | 8630.0 | 91.6 | 2.37 | 0.87 | 112.9 | 8706 | 21600 | 8.66 | 116 |
| 18_4_363 | 3337.0 | 3337.0 | 38.2 | 0.0 | 0.0 | 40.0 | 3337 | 43 | 0 | 40 |
| 18_9_368 | 6713.7 | 6759.0 | 66.5 | 1.18 | 0.52 | 80.6 | 6794 | 21600 | 7.85 | 81 |
| 平均值 | | | | 1.5 | 0.44 | | | | | |

Tab表 11：小型和中型实例的结果（25,000 次自适应大邻域搜索迭代，$\epsilon_3 = 0$）

| | LNS | | | | | | MIP | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 实例 | Avg | Best | Time | AGap | BGap | $\|M\|$ | MIP | Time | 混合整数规划间隙 | $\|M\|$ |
| 8_5_101 | 3478.7 | 3499.0 | 65.1 | 0.86 | 0.28 | 52.7 | 3509 | 942 | 0 | 44 |
| 8_4_160 | 4202.9 | 4229.0 | 74.8 | 1.48 | 0.87 | 60.8 | 4266 | 459 | 0 | 53 |
| 8_2_119 | 1460.0 | 1460.0 | 32.2 | 0.0 | 0.0 | 22.0 | 1460 | 0.5 | 0 | 22 |
| 8_5_72 | 2691.4 | 2731.0 | 48.4 | 1.56 | 0.11 | 35.0 | 2734 | 2 | 0 | 35 |
| 8_2_109 | 2341.0 | 2360.0 | 39.0 | 0.81 | 0.0 | 29.2 | 2360 | 14 | 0 | 30 |
| 8_4_172 | 5175.0 | 5250.0 | 80.9 | 1.43 | 0.0 | 60.0 | 5250 | 753 | 0 | 61 |
| 8_5_124 | 3545.0 | 3565.0 | 55.5 | 0.56 | 0.0 | 48.3 | 3565 | 82 | 0 | 40 |
| 8_5_145 | 4643.8 | 4680.0 | 72.1 | 1.61 | 0.85 | 67.1 | 4720 | 21600 | 3.97 | 69 |
| 8_5_80 | 2688.5 | 2707.0 | 49.1 | 1.09 | 0.4 | 33.5 | 2718 | 61 | 0 | 33 |
| 8_6_136 | 4301.5 | 4315.0 | 88.7 | 0.77 | 0.46 | 75.6 | 4335 | 21600 | 0.62 | 75 |
| 18_10_567 | 9301.5 | 9405.0 | 284.5 | 2.96 | 1.88 | 131.3 | 9585 | 21600 | 6.17 | 135 |
| 18_9_548 | 10074.0 | 10200.0 | 244.8 | 1.49 | 0.25 | 140.5 | 10226 | 21600 | 8.51 | 141 |
| 18_4_412 | 3426.0 | 3440.0 | 88.3 | 0.41 | 0.0 | 38.6 | 3440 | 12560 | 0 | 39 |
| 18_6_440 | 6031.0 | 6107.0 | 117.7 | 1.73 | 0.49 | 70.1 | 6137 | 21600 | 4.31 | 70 |
| 18_4_404 | 4899.5 | 4975.0 | 103.7 | 1.52 | 0.0 | 64.5 | 4975 | 21600 | 4.32 | 66 |
| 18_8_514 | 9264.6 | 9352.0 | 175.8 | 1.25 | 0.32 | 106.7 | 9382 | 21600 | 4.12 | 108 |
| 18_10_710 | 8881.3 | 9035.0 | 320.4 | 1.76 | 0.06 | 119.3 | 9040 | 21600 | 10.55 | 119 |
| 18_8_557 | 8525.6 | 8630.0 | 214.7 | 2.07 | 0.87 | 112.4 | 8706 | 21600 | 8.66 | 116 |
| 18_4_363 | 3337.0 | 3337.0 | 87.3 | 0.0 | 0.0 | 40.0 | 3337 | 43 | 0 | 40 |
| 18_9_368 | 6731.8 | 6759.0 | 153.2 | 0.92 | 0.52 | 81.1 | 6794 | 21600 | 7.85 | 81 |
| 平均值 | | | | 1.21 | 0.37 | | | | | |

Tabl表 12：小型和中型实例的结果（100,000 次自适应大邻域搜索迭代，$\epsilon_3 = 0$）

| | LNS | | | | | | MIP | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 实例 | Avg | Best | Time | AGap | BGap | $\|M\|$ | MIP | Time | 混合整数规划间隙 | $\|M\|$ |
| 8_5_101 | 3492.3 | 3509.0 | 226.7 | 0.48 | 0.0 | 52.2 | 3509 | 942 | 0 | 44 |
| 8_4_160 | 4202.9 | 4229.0 | 271.9 | 1.48 | 0.87 | 60.8 | 4266 | 459 | 0 | 53 |
| 8_2_119 | 1460.0 | 1460.0 | 108.6 | 0.0 | 0.0 | 22.0 | 1460 | 0.5 | 0 | 22 |
| 8_5_72 | 2691.4 | 2731.0 | 166.7 | 1.56 | 0.11 | 35.0 | 2734 | 2 | 0 | 35 |
| 8_2_109 | 2350.0 | 2360.0 | 132.8 | 0.42 | 0.0 | 29.2 | 2360 | 14 | 0 | 30 |
| 8_4_172 | 5211.0 | 5250.0 | 285.2 | 0.74 | 0.0 | 60.5 | 5250 | 753 | 0 | 61 |
| 8_5_124 | 3551.0 | 3565.0 | 190.3 | 0.39 | 0.0 | 48.4 | 3565 | 82 | 0 | 40 |
| 8_5_145 | 4643.8 | 4680.0 | 253.3 | 1.61 | 0.85 | 67.1 | 4720 | 21600 | 3.97 | 69 |
| 8_5_80 | 2697.0 | 2718.0 | 166.1 | 0.77 | 0.0 | 33.4 | 2718 | 61 | 0 | 33 |
| 8_6_136 | 4308.0 | 4320.0 | 328.8 | 0.62 | 0.35 | 75.7 | 4335 | 21600 | 0.62 | 75 |
| 18_10_567 | 9367.0 | 9575.0 | 1117.0 | 2.27 | 0.1 | 131.8 | 9585 | 21600 | 6.17 | 135 |
| 18_9_548 | 10123.2 | 10200.0 | 935.0 | 1.01 | 0.25 | 139.0 | 10226 | 21600 | 8.51 | 141 |
| 18_4_412 | 3426.0 | 3440.0 | 326.3 | 0.41 | 0.0 | 38.6 | 3440 | 12560 | 0 | 39 |
| 18_6_440 | 6041.0 | 6127.0 | 428.4 | 1.56 | 0.16 | 70.0 | 6137 | 21600 | 4.31 | 70 |
| 18_4_404 | 4918.0 | 4980.0 | 389.3 | 1.15 | -0.1 | 65.0 | 4975 | 21600 | 4.32 | 66 |
| 18_8_514 | 9297.4 | 9362.0 | 633.5 | 0.9 | 0.21 | 107.0 | 9382 | 21600 | 4.12 | 108 |
| 18_10_710 | 8947.3 | 9065.0 | 1246.6 | 1.03 | -0.28 | 120.4 | 9040 | 21600 | 10.55 | 119 |
| 18_8_557 | 8549.6 | 8630.0 | 803.4 | 1.8 | 0.87 | 112.7 | 8706 | 21600 | 8.66 | 116 |
| 18_4_363 | 3337.0 | 3337.0 | 322.1 | 0.0 | 0.0 | 40.0 | 3337 | 43 | 0 | 40 |
| 18_9_368 | 6735.6 | 6767.0 | 568.9 | 0.86 | 0.4 | 81.0 | 6794 | 21600 | 7.85 | 81 |
| 平均值 | | | | 0.95 | 0.19 | | | | | |

表13: 使用权重集 {1,0,0,0,0}，迭代 100,000 次时，设置为优先级的自适应大邻域搜索结果 s.

| 实例 | 优先级 | 惩罚 | 组 | 连续 | 工作量 | OF |
|---|---|---|---|---|---|---|
| 8_5_101 | 3496.0 | -15.8 | 4.18 | 0.0 | 0.9 | 3480.2 |
| 8_4_160 | 4257.0 | -20.9 | 3.23 | 0.0 | 0.91 | 4236.1 |
| 8_2_119 | 1460.0 | -3.0 | 2.0 | 0.0 | 0.88 | 1457.0 |
| 8_5_72 | 2816.0 | -155.4 | 3.32 | 0.0 | 0.88 | 2660.6 |
| 8_2_109 | 2341.0 | 0.0 | 3.65 | 0.0 | 0.87 | 2341.0 |
| 8_4_172 | 5219.0 | 0.0 | 2.5 | 0.0 | 0.87 | 5219.0 |
| 8_5_124 | 3548.0 | -11.8 | 2.4 | 0.0 | 0.85 | 3536.2 |
| 8_5_145 | 4702.0 | -27.7 | 1.66 | 0.0 | 0.89 | 4674.3 |
| 8_5_80 | 2814.0 | -151.5 | 2.82 | 0.8 | 0.89 | 2662.5 |
| 8_6_136 | 4321.0 | -7.5 | 2.93 | 0.0 | 0.92 | 4313.5 |
| 18_10_567 | 9426.0 | -14.1 | 2.66 | 2.6 | 0.9 | 9411.9 |
| 18_9_548 | 10153.0 | -9.8 | 2.76 | 2.7 | 0.9 | 10143.2 |
| 18_4_412 | 3434.0 | 0.0 | 1.35 | 3.9 | 0.92 | 3434.0 |
| 18_6_440 | 6049.0 | -33.0 | 2.48 | 6.3 | 0.88 | 6016.0 |
| 18_4_404 | 4958.0 | -75.0 | 3.25 | 0.1 | 0.9 | 4883.0 |
| 18_8_514 | 9684.0 | -517.3 | 1.74 | 3.1 | 0.86 | 9166.7 |
| 18_10_710 | 9090.0 | -117.3 | 2.13 | 1.7 | 0.87 | 8972.7 |
| 18_8_557 | 8610.0 | -3.8 | 2.38 | 0.8 | 0.9 | 8606.2 |
| 18_4_363 | 3420.0 | -83.0 | 1.75 | 1.2 | 0.87 | 3337.0 |
| 18_9_368 | 6836.0 | -93.5 | 2.16 | 4.5 | 0.89 | 6742.5 |
| 24_45_1072 | 31824.0 | -495.8 | 2.71 | 8.2 | 0.85 | 31328.2 |
| 24_38_906 | 32734.0 | -186.5 | 2.29 | 10.6 | 0.87 | 32547.5 |
| 24_45_621 | 32865.0 | -254.5 | 1.63 | 15.8 | 0.83 | 32610.5 |
| 24_43_767 | 35408.0 | -476.3 | 1.75 | 18.3 | 0.85 | 34931.7 |
| 24_42_749 | 36360.0 | -313.6 | 2.1 | 9.9 | 0.84 | 36046.4 |
| 24_40_825 | 38155.0 | -296.5 | 2.32 | 6.4 | 0.84 | 37858.5 |
| 24_45_1441 | 39270.0 | -150.9 | 1.9 | 4.7 | 0.84 | 39119.1 |
| 24_45_1076 | 36358.0 | -182.9 | 2.25 | 5.2 | 0.87 | 36175.1 |
| 24_44_610 | 32844.0 | -241.0 | 2.14 | 9.3 | 0.84 | 32603.0 |
| 24_52_662 | 36459.0 | -690.6 | 1.7 | 15.0 | 0.84 | 35768.4 |
| 72_230_2505 | 123114.0 | -1261.3 | 2.33 | 28.7 | 0.82 | 121852.7 |
| 72_211_2438 | 130057.0 | -676.6 | 2.09 | 37.2 | 0.83 | 129380.4 |
| 72_224_2436 | 142137.0 | -991.9 | 1.97 | 40.2 | 0.81 | 141145.1 |
| 72_211_2400 | 145159.0 | -3375.0 | 1.98 | 45.6 | 0.82 | 141784.0 |
| 72_216_2027 | 128556.0 | -838.7 | 2.19 | 24.0 | 0.8 | 127717.3 |
| 72_230_3127 | 155038.0 | -2106.8 | 2.36 | 21.3 | 0.81 | 152931.2 |
| 72_213_3244 | 145748.0 | -1056.9 | 2.23 | 20.7 | 0.83 | 144691.1 |
| 72_218_2456 | 126024.0 | -1151.5 | 2.04 | 31.1 | 0.82 | 124872.5 |
| 72_234_2543 | 146958.0 | -1800.5 | 2.26 | 34.0 | 0.82 | 145157.5 |
| 72_180_2258 | 93309.0 | -1428.8 | 1.98 | 25.9 | 0.81 | 91880.2 |
| 平均值 | 44875.28 | -482.92 | 2.34 | 11.0 | 0.86 | 44392.35 |

表14: 使用权重集 {1,100,0,0,0}，进行100,000次迭代的惩罚设置下的自适应大邻域搜索结果 s

| 实例 | 优先级 | 惩罚 | 组 | 连续 | 工作量 | OF |
|---|---|---|---|---|---|---|
| 8_5_101 | 3464.0 | 0.0 | 4.42 | 0.0 | 0.89 | 3464.0 |
| 8_4_160 | 4200.0 | 0.0 | 3.0 | 0.0 | 0.92 | 4200.0 |
| 8_2_119 | 1460.0 | 0.0 | 2.0 | 0.0 | 0.88 | 1460.0 |
| 8_5_72 | 2650.0 | 0.0 | 2.88 | 0.0 | 0.86 | 2650.0 |
| 8_2_109 | 2356.0 | 0.0 | 3.75 | 0.0 | 0.86 | 2356.0 |
| 8_4_172 | 5240.0 | 0.0 | 2.5 | 0.0 | 0.87 | 5240.0 |
| 8_5_124 | 3480.0 | 0.0 | 2.46 | 0.0 | 0.86 | 3480.0 |
| 8_5_145 | 4680.0 | 0.0 | 1.6 | 0.0 | 0.89 | 4680.0 |
| 8_5_80 | 2510.0 | 0.0 | 2.74 | 0.4 | 0.89 | 2510.0 |
| 8_6_136 | 4288.0 | 0.0 | 3.0 | 0.0 | 0.92 | 4288.0 |
| 18_10_567 | 9311.0 | 0.0 | 2.55 | 1.6 | 0.89 | 9311.0 |
| 18_9_548 | 10162.0 | 0.0 | 2.74 | 2.3 | 0.9 | 10162.0 |
| 18_4_412 | 3434.0 | 0.0 | 1.35 | 2.9 | 0.92 | 3434.0 |
| 18_6_440 | 6018.0 | 0.0 | 2.43 | 7.5 | 0.88 | 6018.0 |
| 18_4_404 | 4886.0 | 0.0 | 2.55 | 0.2 | 0.91 | 4886.0 |
| 18_8_514 | 9256.0 | 0.0 | 1.88 | 3.3 | 0.87 | 9256.0 |
| 18_10_710 | 8959.0 | 0.0 | 2.22 | 1.7 | 0.88 | 8959.0 |
| 18_8_557 | 8630.0 | 0.0 | 2.44 | 0.7 | 0.9 | 8630.0 |
| 18_4_363 | 3040.0 | 0.0 | 1.5 | 1.4 | 0.91 | 3040.0 |
| 18_9_368 | 6674.0 | 0.0 | 2.42 | 6.5 | 0.9 | 6674.0 |
| 24_45_1072 | 31328.0 | 0.0 | 2.72 | 5.4 | 0.86 | 31328.0 |
| 24_38_906 | 32452.0 | 0.0 | 2.63 | 9.4 | 0.88 | 32452.0 |
| 24_45_621 | 32668.0 | 0.0 | 1.67 | 14.7 | 0.84 | 32668.0 |
| 24_43_767 | 35086.0 | 0.0 | 1.69 | 17.1 | 0.84 | 35086.0 |
| 24_42_749 | 35799.0 | 0.0 | 2.09 | 9.1 | 0.84 | 35799.0 |
| 24_40_825 | 37765.0 | 0.0 | 2.31 | 5.8 | 0.85 | 37765.0 |
| 24_45_1441 | 38991.0 | 0.0 | 1.89 | 5.7 | 0.84 | 38991.0 |
| 24_45_1076 | 36187.0 | 0.0 | 2.23 | 5.8 | 0.87 | 36187.0 |
| 24_44_610 | 31753.0 | 0.0 | 1.85 | 10.2 | 0.83 | 31753.0 |
| 24_52_662 | 35600.0 | 0.0 | 1.81 | 17.2 | 0.84 | 35600.0 |
| 72_230_2505 | 121500.0 | 0.0 | 2.37 | 29.3 | 0.83 | 121500.0 |
| 72_211_2438 | 128646.0 | 0.0 | 2.1 | 34.4 | 0.83 | 128646.0 |
| 72_224_2436 | 141154.0 | 0.0 | 1.93 | 46.2 | 0.81 | 141154.0 |
| 72_211_2400 | 140377.0 | 0.0 | 2.02 | 38.3 | 0.82 | 140377.0 |
| 72_216_2027 | 126875.0 | 0.0 | 2.22 | 25.0 | 0.8 | 126875.0 |
| 72_230_3127 | 152438.0 | 0.0 | 2.45 | 20.0 | 0.82 | 152438.0 |
| 72_213_3244 | 141764.0 | 0.0 | 2.28 | 18.6 | 0.83 | 141764.0 |
| 72_218_2456 | 124510.0 | -0.1 | 2.06 | 27.5 | 0.83 | 124509.9 |
| 72_234_2543 | 140511.0 | 0.0 | 2.06 | 29.8 | 0.81 | 140511.0 |
| 72_180_2258 | 89846.0 | 0.0 | 1.95 | 29.9 | 0.82 | 89846.0 |
| 平均值 | 43998.7 | -0.0 | 2.32 | 10.7 | 0.86 | 43998.7 |

表15：使用权重集 {1,1,-10,0,0} ，迭代 100,000 次时，设置为 TaskGroups 的自适应大邻域搜索结果。

| 实例 | 优先级 | 惩罚 | 组 | 连续 | 工作量 | OF |
|---|---|---|---|---|---|---|
| 8_5_101 | 3492.0 | -3.8 | 4.0 | 0.0 | 0.89 | 3488.2 |
| 8_4_160 | 4240.0 | -11.0 | 3.25 | 0.0 | 0.91 | 4229.0 |
| 8_2_119 | 1460.0 | 0.0 | 2.0 | 0.0 | 0.88 | 1460.0 |
| 8_5_72 | 2734.0 | -24.7 | 2.56 | 0.0 | 0.84 | 2709.3 |
| 8_2_109 | 2345.0 | 0.0 | 3.3 | 0.0 | 0.87 | 2345.0 |
| 8_4_172 | 5234.0 | 0.0 | 2.15 | 0.0 | 0.87 | 5234.0 |
| 8_5_124 | 3570.0 | -5.0 | 2.4 | 0.0 | 0.86 | 3565.0 |
| 8_5_145 | 4679.0 | -1.9 | 1.62 | 0.0 | 0.89 | 4677.1 |
| 8_5_80 | 2769.0 | -65.2 | 2.58 | 0.6 | 0.87 | 2703.8 |
| 8_6_136 | 4318.0 | -4.5 | 2.75 | 0.0 | 0.92 | 4313.5 |
| 18_10_567 | 9430.0 | -11.7 | 2.6 | 1.7 | 0.89 | 9418.3 |
| 18_9_548 | 10157.0 | -0.8 | 2.77 | 2.3 | 0.9 | 10156.2 |
| 18_4_412 | 3432.0 | 0.0 | 1.3 | 4.2 | 0.92 | 3432.0 |
| 18_6_440 | 6060.0 | -11.7 | 2.5 | 7.7 | 0.87 | 6048.3 |
| 18_4_404 | 4953.0 | -29.0 | 2.98 | 0.0 | 0.91 | 4924.0 |
| 18_8_514 | 9440.0 | -99.4 | 1.91 | 3.3 | 0.86 | 9340.6 |
| 18_10_710 | 9082.0 | -61.3 | 2.18 | 1.7 | 0.87 | 9020.7 |
| 18_8_557 | 8578.0 | -1.6 | 2.36 | 0.5 | 0.9 | 8576.4 |
| 18_4_363 | 3420.0 | -83.0 | 1.75 | 1.2 | 0.86 | 3337.0 |
| 18_9_368 | 6825.0 | -35.8 | 2.16 | 6.5 | 0.89 | 6789.2 |
| 24_45_1072 | 31630.0 | -141.0 | 2.66 | 5.9 | 0.86 | 31489.0 |
| 24_38_906 | 32608.0 | -29.0 | 2.5 | 8.2 | 0.88 | 32579.0 |
| 24_45_621 | 32788.0 | -76.1 | 1.66 | 14.6 | 0.83 | 32711.9 |
| 24_43_767 | 35352.0 | -73.7 | 1.73 | 19.9 | 0.84 | 35278.3 |
| 24_42_749 | 36239.0 | -170.4 | 2.13 | 9.6 | 0.84 | 36068.6 |
| 24_40_825 | 38075.0 | -117.7 | 2.27 | 7.6 | 0.83 | 37957.3 |
| 24_45_1441 | 39238.0 | -97.3 | 1.88 | 5.7 | 0.84 | 39140.7 |
| 24_45_1076 | 36301.0 | -58.8 | 2.26 | 5.6 | 0.86 | 36242.2 |
| 24_44_610 | 32804.0 | -163.2 | 2.02 | 9.1 | 0.83 | 32640.8 |
| 24_52_662 | 36167.0 | -216.0 | 1.76 | 15.3 | 0.84 | 35951.0 |
| 72_230_2505 | 122365.0 | -320.5 | 2.32 | 28.9 | 0.83 | 122044.5 |
| 72_211_2438 | 129745.0 | -316.4 | 2.09 | 39.0 | 0.83 | 129428.6 |
| 72_224_2436 | 141899.0 | -374.8 | 1.93 | 45.8 | 0.81 | 141524.2 |
| 72_211_2400 | 144527.0 | -1641.1 | 1.98 | 41.6 | 0.82 | 142885.9 |
| 72_216_2027 | 128356.0 | -388.4 | 2.2 | 27.2 | 0.8 | 127967.6 |
| 72_230_3127 | 154881.0 | -913.2 | 2.35 | 23.4 | 0.81 | 153967.8 |
| 72_213_3244 | 145543.0 | -695.8 | 2.19 | 20.7 | 0.82 | 144847.2 |
| 72_218_2456 | 125263.0 | -277.8 | 2.02 | 32.6 | 0.83 | 124985.2 |
| 72_234_2543 | 146888.0 | -1120.7 | 2.22 | 27.6 | 0.82 | 145767.3 |
| 72_180_2258 | 92247.0 | -600.2 | 1.92 | 29.0 | 0.81 | 91646.8 |
| 平均值 | 44728.35 | -206.06 | 2.28 | 11.18 | 0.86 | 44522.29 |

表16：使用权重集 {1,1,0,0,10000} 的工作量设置下，自适应大邻域搜索结果，迭代100,000次。

| 实例 | 优先级 | 惩罚 | 组 | 连续 | 工作量 | OF |
|---|---|---|---|---|---|---|
| 8_5_101 | 3435.0 | -6.4 | 4.36 | 0.0 | 0.94 | 3428.6 |
| 8_4_160 | 4163.0 | -18.7 | 3.25 | 0.0 | 0.93 | 4144.3 |
| 8_2_119 | 1355.0 | -2.0 | 2.0 | 0.0 | 0.96 | 1353.0 |
| 8_5_72 | 2656.0 | -22.5 | 3.18 | 0.0 | 0.93 | 2633.5 |
| 8_2_109 | 2272.0 | 0.0 | 3.4 | 0.0 | 0.96 | 2272.0 |
| 8_4_172 | 5061.0 | 0.0 | 2.62 | 0.0 | 0.92 | 5061.0 |
| 8_5_124 | 3420.0 | -5.0 | 2.56 | 0.0 | 0.89 | 3415.0 |
| 8_5_145 | 4586.0 | 0.0 | 1.4 | 0.0 | 0.92 | 4586.0 |
| 8_5_80 | 2704.0 | -79.6 | 2.76 | 0.7 | 0.92 | 2624.4 |
| 8_6_136 | 4183.0 | 0.0 | 3.28 | 0.0 | 0.94 | 4183.0 |
| 18_10_567 | 9224.0 | -13.5 | 2.75 | 2.6 | 0.91 | 9210.5 |
| 18_9_548 | 9963.0 | -1.8 | 3.12 | 2.4 | 0.92 | 9961.2 |
| 18_4_412 | 3434.0 | 0.0 | 1.35 | 4.3 | 0.93 | 3434.0 |
| 18_6_440 | 5921.0 | -10.4 | 2.73 | 4.5 | 0.92 | 5910.6 |
| 18_4_404 | 4939.0 | -18.5 | 3.02 | 0.4 | 0.92 | 4920.5 |
| 18_8_514 | 9069.0 | -29.6 | 2.11 | 2.6 | 0.91 | 9039.4 |
| 18_10_710 | 8850.0 | -65.5 | 2.15 | 2.0 | 0.89 | 8784.5 |
| 18_8_557 | 8475.0 | -0.4 | 2.36 | 0.6 | 0.91 | 8474.6 |
| 18_4_363 | 3340.0 | -68.0 | 1.75 | 1.0 | 0.93 | 3272.0 |
| 18_9_368 | 6697.0 | -38.2 | 2.33 | 6.1 | 0.91 | 6658.8 |
| 24_45_1072 | 31413.0 | -139.5 | 2.72 | 8.3 | 0.86 | 31273.5 |
| 24_38_906 | 32427.0 | -38.1 | 2.52 | 8.6 | 0.88 | 32388.9 |
| 24_45_621 | 32659.0 | -76.9 | 1.69 | 13.2 | 0.84 | 32582.1 |
| 24_43_767 | 35094.0 | -51.3 | 1.73 | 20.4 | 0.85 | 35042.7 |
| 24_42_749 | 36053.0 | -239.9 | 2.16 | 8.9 | 0.85 | 35813.1 |
| 24_40_825 | 37760.0 | -133.7 | 2.39 | 4.8 | 0.85 | 37626.3 |
| 24_45_1441 | 39095.0 | -98.4 | 1.94 | 6.1 | 0.85 | 38996.6 |
| 24_45_1076 | 36057.0 | -60.2 | 2.2 | 5.7 | 0.87 | 35996.8 |
| 24_44_610 | 32615.0 | -162.2 | 2.03 | 7.9 | 0.84 | 32452.8 |
| 24_52_662 | 35937.0 | -165.9 | 1.84 | 16.3 | 0.85 | 35771.1 |
| 72_230_2505 | 122360.0 | -363.2 | 2.36 | 25.8 | 0.83 | 121996.8 |
| 72_211_2438 | 129625.0 | -253.7 | 2.08 | 37.1 | 0.83 | 129371.3 |
| 72_224_2436 | 141830.0 | -368.2 | 1.95 | 43.6 | 0.82 | 141461.8 |
| 72_211_2400 | 144502.0 | -1543.8 | 1.99 | 44.3 | 0.82 | 142958.2 |
| 72_216_2027 | 128236.0 | -432.7 | 2.16 | 23.5 | 0.81 | 127803.3 |
| 72_230_3127 | 154722.0 | -905.5 | 2.45 | 20.3 | 0.82 | 153816.5 |
| 72_213_3244 | 145427.0 | -640.8 | 2.19 | 22.5 | 0.83 | 144786.2 |
| 72_218_2456 | 125136.0 | -285.8 | 2.05 | 33.5 | 0.83 | 124850.2 |
| 72_234_2543 | 146546.0 | -1056.9 | 2.23 | 28.8 | 0.82 | 145489.1 |
| 72_180_2258 | 91809.0 | -404.8 | 1.95 | 28.8 | 0.82 | 91404.2 |
| 平均值 | 44576.25 | -195.04 | 2.38 | 10.89 | 0.88 | 44381.21 |

表17: 设置为 Control、使用权重集 {1,1,0,0,0} 并进行100,000次迭代的自适应大邻域搜索结果 s.

| 实例 | 优先级 | 惩罚 | 组 | 连续 | 工作量 | OF |
|---|---|---|---|---|---|---|
| 8_5_101 | 3493.0 | -3.0 | 4.22 | 0.0 | 0.9 | 3490.0 |
| 8_4_160 | 4240.0 | -11.0 | 3.27 | 0.0 | 0.91 | 4229.0 |
| 8_2_119 | 1460.0 | 0.0 | 2.0 | 0.0 | 0.88 | 1460.0 |
| 8_5_72 | 2747.0 | -28.6 | 2.98 | 0.0 | 0.86 | 2718.4 |
| 8_2_109 | 2357.0 | 0.0 | 3.75 | 0.0 | 0.86 | 2357.0 |
| 8_4_172 | 5188.0 | 0.0 | 2.45 | 0.0 | 0.87 | 5188.0 |
| 8_5_124 | 3559.0 | -5.0 | 2.4 | 0.0 | 0.85 | 3554.0 |
| 8_5_145 | 4682.0 | -2.6 | 1.64 | 0.0 | 0.89 | 4679.4 |
| 8_5_80 | 2755.0 | -63.0 | 2.76 | 0.7 | 0.87 | 2692.0 |
| 8_6_136 | 4319.0 | -5.0 | 3.03 | 0.0 | 0.92 | 4314.0 |
| 18_10_567 | 9408.0 | -11.2 | 2.64 | 2.8 | 0.89 | 9396.8 |
| 18_9_548 | 10147.0 | -3.8 | 2.82 | 2.2 | 0.9 | 10143.2 |
| 18_4_412 | 3434.0 | 0.0 | 1.35 | 3.6 | 0.92 | 3434.0 |
| 18_6_440 | 6036.0 | -9.1 | 2.6 | 7.1 | 0.88 | 6026.9 |
| 18_4_404 | 4945.0 | -33.0 | 3.1 | 0.0 | 0.91 | 4912.0 |
| 18_8_514 | 9439.0 | -96.6 | 1.89 | 2.8 | 0.86 | 9342.4 |
| 18_10_710 | 9088.0 | -55.3 | 2.25 | 0.6 | 0.88 | 9032.7 |
| 18_8_557 | 8622.0 | -1.0 | 2.49 | 0.9 | 0.9 | 8621.0 |
| 18_4_363 | 3420.0 | -83.0 | 1.77 | 1.2 | 0.87 | 3337.0 |
| 18_9_368 | 6829.0 | -34.9 | 2.16 | 5.6 | 0.89 | 6794.1 |
| 24_45_1072 | 31669.0 | -149.7 | 2.61 | 7.4 | 0.86 | 31519.3 |
| 24_38_906 | 32617.0 | -34.9 | 2.43 | 8.8 | 0.87 | 32582.1 |
| 24_45_621 | 32847.0 | -93.0 | 1.67 | 17.2 | 0.83 | 32754.0 |
| 24_43_767 | 35291.0 | -85.8 | 1.71 | 19.4 | 0.84 | 35205.2 |
| 24_42_749 | 36242.0 | -182.5 | 2.1 | 8.6 | 0.84 | 36059.5 |
| 24_40_825 | 38064.0 | -188.2 | 2.29 | 6.5 | 0.83 | 37875.8 |
| 24_45_1441 | 39229.0 | -98.6 | 1.91 | 4.0 | 0.84 | 39130.4 |
| 24_45_1076 | 36382.0 | -63.8 | 2.23 | 6.1 | 0.86 | 36318.2 |
| 24_44_610 | 32774.0 | -154.1 | 2.05 | 10.1 | 0.83 | 32619.9 |
| 24_52_662 | 36156.0 | -177.5 | 1.76 | 16.4 | 0.84 | 35978.5 |
| 72_230_2505 | 122565.0 | -420.0 | 2.35 | 27.4 | 0.83 | 122145.0 |
| 72_211_2438 | 129817.0 | -317.1 | 2.14 | 37.0 | 0.83 | 129499.9 |
| 72_224_2436 | 141791.0 | -378.8 | 1.95 | 41.6 | 0.81 | 141412.2 |
| 72_211_2400 | 144650.0 | -1632.4 | 1.98 | 45.0 | 0.82 | 143017.6 |
| 72_216_2027 | 128449.0 | -449.7 | 2.16 | 27.5 | 0.8 | 127999.3 |
| 72_230_3127 | 154951.0 | -946.6 | 2.44 | 19.6 | 0.82 | 154004.4 |
| 72_213_3244 | 145484.0 | -634.4 | 2.19 | 20.1 | 0.83 | 144849.6 |
| 72_218_2456 | 125358.0 | -276.8 | 2.05 | 28.4 | 0.83 | 125081.2 |
| 72_234_2543 | 146766.0 | -1096.9 | 2.21 | 32.2 | 0.82 | 145669.1 |
| 72_180_2258 | 92074.0 | -435.0 | 1.96 | 31.4 | 0.81 | 91639.0 |
| 平均值 | 44733.6 | -206.55 | 2.34 | 11.05 | 0.86 | 44527.05 |

表 18：采用权重集 {1,1,-1,1,1000}，设置为 AllObjectives，进行 100,000 次迭代时的 ALNS+结果。

| 实例 | 优先级 | 惩罚 | 组 | 连续 | 工作量 | OF |
|---|---|---|---|---|---|---|
| 8_5_101 | 3493.0 | -6.3 | 4.26 | 0.0 | 0.91 | 3486.7 |
| 8_4_160 | 4239.0 | -11.0 | 3.25 | 0.0 | 0.92 | 4228.0 |
| 8_2_119 | 1440.0 | -5.0 | 2.0 | 0.0 | 0.94 | 1435.0 |
| 8_5_72 | 2764.0 | -31.2 | 3.24 | 0.0 | 0.92 | 2732.8 |
| 8_2_109 | 2313.0 | 0.0 | 3.5 | 0.0 | 0.94 | 2313.0 |
| 8_4_172 | 5235.0 | 0.0 | 2.65 | 0.0 | 0.89 | 5235.0 |
| 8_5_124 | 3559.0 | -5.0 | 2.4 | 0.0 | 0.86 | 3554.0 |
| 8_5_145 | 4682.0 | -3.7 | 1.66 | 0.0 | 0.9 | 4678.3 |
| 8_5_80 | 2745.0 | -64.9 | 2.66 | 1.0 | 0.89 | 2680.1 |
| 8_6_136 | 4316.0 | -4.5 | 3.1 | 0.0 | 0.92 | 4311.5 |
| 18_10_567 | 9380.0 | -11.8 | 2.64 | 4.3 | 0.9 | 9368.2 |
| 18_9_548 | 10192.0 | -3.4 | 2.79 | 4.8 | 0.91 | 10188.6 |
| 18_4_412 | 3430.0 | 0.0 | 1.25 | 11.1 | 0.93 | 3430.0 |
| 18_6_440 | 6050.0 | -13.0 | 2.67 | 11.5 | 0.89 | 6037.0 |
| 18_4_404 | 4957.0 | -25.0 | 2.83 | 1.1 | 0.91 | 4932.0 |
| 18_8_514 | 9453.0 | -99.0 | 1.89 | 4.2 | 0.87 | 9354.0 |
| 18_10_710 | 9100.0 | -55.3 | 2.24 | 1.9 | 0.88 | 9044.7 |
| 18_8_557 | 8643.0 | -1.4 | 2.48 | 2.6 | 0.91 | 8641.6 |
| 18_4_363 | 3400.0 | -73.0 | 1.75 | 6.5 | 0.91 | 3327.0 |
| 18_9_368 | 6841.0 | -41.0 | 2.18 | 10.4 | 0.9 | 6800.0 |
| 24_45_1072 | 31661.0 | -144.1 | 2.6 | 9.3 | 0.85 | 31516.9 |
| 24_38_906 | 32665.0 | -39.8 | 2.11 | 22.1 | 0.87 | 32625.2 |
| 24_45_621 | 32744.0 | -78.4 | 1.58 | 34.1 | 0.83 | 32665.6 |
| 24_43_767 | 35349.0 | -75.6 | 1.69 | 35.1 | 0.84 | 35273.4 |
| 24_42_749 | 36397.0 | -212.1 | 2.05 | 17.9 | 0.84 | 36184.9 |
| 24_40_825 | 38074.0 | -172.8 | 2.3 | 11.8 | 0.83 | 37901.2 |
| 24_45_1441 | 39311.0 | -111.1 | 1.9 | 8.0 | 0.84 | 39199.9 |
| 24_45_1076 | 36299.0 | -62.9 | 2.2 | 8.2 | 0.86 | 36236.1 |
| 24_44_610 | 32837.0 | -159.8 | 1.99 | 18.1 | 0.83 | 32677.2 |
| 24_52_662 | 36184.0 | -199.0 | 1.77 | 37.2 | 0.84 | 35985.0 |
| 72_230_2505 | 122491.0 | -409.5 | 2.33 | 57.6 | 0.82 | 122081.5 |
| 72_211_2438 | 129959.0 | -296.7 | 2.1 | 78.2 | 0.83 | 129662.3 |
| 72_224_2436 | 142132.0 | -333.3 | 1.93 | 103.1 | 0.81 | 141798.7 |
| 72_211_2400 | 144478.0 | -1488.3 | 2.0 | 95.1 | 0.82 | 142989.7 |
| 72_216_2027 | 128468.0 | -425.6 | 2.14 | 54.4 | 0.8 | 128042.4 |
| 72_230_3127 | 154910.0 | -854.4 | 2.38 | 40.6 | 0.82 | 154055.6 |
| 72_213_3244 | 145566.0 | -690.8 | 2.16 | 35.1 | 0.83 | 144875.2 |
| 72_218_2456 | 125318.0 | -248.3 | 2.03 | 68.3 | 0.82 | 125069.7 |
| 72_234_2543 | 146794.0 | -1123.7 | 2.21 | 75.9 | 0.82 | 145670.3 |
| 72_180_2258 | 92187.0 | -505.6 | 1.95 | 72.5 | 0.81 | 91681.4 |
| 平均值 | 44751.4 | -202.16 | 2.32 | 23.55 | 0.87 | 44549.24 |