

# 网约车任务分配系统优化<sup>①</sup>



陈立军, 张屹, 陈孝如, 杨微

(广州软件学院 软件工程系, 广州 510990)

通信作者: 陈立军, E-mail: cij@mail.seig.edu.cn

**摘要:** 网约车是一种广泛应用的共享移动应用, 其核心问题是将出租车请求分配给具有不同目标的司机, 尽管对网约车的任务分配进行了广泛的研究, 但在很大程度上忽视了司机之间收入的公平性, 由于优化视角的短视和分配技术的耗时, 先行者对网约车公平任务分配的研究在公平性、效用性方面还存在不足. 在本文中, 提出了公平分配学习 (LAF) 方法, 它既优化了效用又优化了公平性的高效任务分配方案, 采用强化学习以整体的方式进行分配, 并提出一套加速技术, 以实现大规模数据的快速公平分配. 实验结果表明, 公平分配学习方法在公平性、效用性和效率方面分别比现有水平高出 86.7%、29.1% 和 797%.

**关键词:** 网约车; 任务分配; 路径规划; 强化学习; 公平分配; 优化调度

引用格式: 陈立军, 张屹, 陈孝如, 杨微. 网约车任务分配系统优化. 计算机系统应用, 2022, 31(6): 19-28. <http://www.c-s-a.org.cn/1003-3254/8540.html>

## Optimization of Task Allocation System for Online Car-hailing

CHEN Li-Jun, ZHANG Yi, CHEN Xiao-Ru, YANG Wei

(Department of Software Engineering, Software Engineering Institute of Guangzhou, Guangzhou 510990, China)

**Abstract:** Online car-hailing is a kind of widely used mobile application. Its core problem is to assign requests to taxi drivers with different goals. Although extensive research on task allocation has been carried out, a largely ignored problem is the income equality of drivers. Due to the short-sighted optimization and time-consuming allocation, fairness and utility receive less attention in the research on fair task allocation. In this study, an efficient task assignment scheme, learning to assign with fairness (LAF), was proposed to optimize both utility and fairness. It adopts reinforcement learning to allocate tasks holistically and proposes a set of acceleration techniques to achieve rapid and equitable allocation on a large scale. The experimental results show that the fairness, effectiveness, and efficiency of LAF are 86.7%, 29.1%, and 797% higher than the existing level, respectively.

**Key words:** online car-hailing; task distribution; path planning; reinforcement learning; fair distribution; optimal operation

网约车是一种新兴的共享出行服务, 可显著提高城市交通容量, 例如, 中国的一个主要网约车平台在 2020 年的日出行量已超过 5000 万<sup>[1]</sup>. 网约车服务由司机、乘客和平台组成, 平台将乘客提交给司机与出租车请求进行匹配, 出租车请求和司机之间匹配, 也就是任务分配, 是网约车研究的核心问题<sup>[2]</sup>.

现有研究主要从平台或乘客的角度优化任务分配,

而很少从驾驶员的角度研究问题, 例如, 已经设计了诸多分配算法来最大化效用<sup>[2,3]</sup> 或最小化旅行成本<sup>[4]</sup> 或乘客的等待时间<sup>[5]</sup>, 直到最近, 先行工作<sup>[6]</sup> 才探索司机的收益公平性, 例如在任务分配期间平衡司机之间的收入. 本文认为, 盲目地将司机分配到最大化效用的任务可能会损害司机之间显著的收入差距<sup>[7]</sup>. 图 1 通过仅最大化效用的分配算法模拟了 4 个驾驶员的轨迹, 司

① 基金项目: 2021 年度广东省普通高校重点科研平台和科研项目 (2021 KTSCX160); 广东省质量工程 (ZXKC202105)

收稿时间: 2021-09-14; 修改时间: 2021-10-14; 采用时间: 2021-10-24; csa 在线出版时间: 2022-05-26

司机1首先被分配一个请求,该请求在偏远区域结束,完成这个订单后,他很难回到繁忙的地区,导致他每小时收入很低;相比之下,司机3很幸运,在繁忙区域结束一个订单后,又立即被分配到另一个繁忙区域请求,他可以获得更高的每小时收入;司机2留在繁忙区域,但很少被分配到请求,长时间的空闲时间导致他低小时收入;相比之下,司机4碰巧被分配了许多请求,而且空闲时间短有助于提高每小时收入.司机之间的这种每小时收入差距(在本例中为司机1、司机2与司机3、司机4)可能会使司机气馁并导致不公平的任务分配.



图1 模拟驾驶员轨迹

尽管在云计算中的负载均衡<sup>[8]</sup>等应用领域中,任务分配的公平性得到了广泛的研究,但网约车中的公平任务分配面临着独特的挑战:(1)在线设置:网约车是一种双向在线分配场景,具有高时空依赖性和变化<sup>[9]</sup>;(2)双目标优化:一个理想的网约车分配算法应该在各种实际约束下优化司机之间的效用和公平性;(3)高效率要求:现实世界的网约车平台需要对城市规模的数据进行快速分配.最先进的网约车公平分配算法<sup>[6,10]</sup>未能解决所有这些挑战.首先,这些解决方案是短视的,也就是说,他们忽略了当前分配对未来分配的影响,这在很大程度上缩小了优化空间,从而显著降低了可实现的效用和公平性;其次,它们要么依赖于线性规划<sup>[6]</sup>,要么需要多轮重新分配<sup>[10]</sup>,这使得它们对于大规模数据的实时响应效率低下.

本文提出了公平分配学习(learning to assign with fairness, LAF)方法,一个有效和高效的分配方案,优化了效用(以所有司机的期望总收入衡量)和公平(以司机之间的时间收入公平衡量).LAF通过强化学习,学习未来感知分配策略,明确地说明了作业之间的依赖性,这种学习分配策略可以整体优化效用和公平

性.为了实现高效分配,LAF将公平性和效用优化嵌入到相同的赋值操作中,并利用二部图的稀疏性进一步加速.值得一提的是,本文提出了一个加权平摊公平度量而不是传统的未加权<sup>[6]</sup>来表征司机在更细的时间粒度上的收入公平性,如小时收入公平性.

评价结果显示,在公平和效用方面,比普通水平分别提高了45.7%–86.7%和7.7%–29.1%<sup>[6,10]</sup>.本文的LAF也比文献<sup>[6,10]</sup>快到700.本文主要贡献如下:

(1)据本文所知,这是第一个明确考虑当前和未来任务之间的依赖性的工作,以提高公平的任务分配在网约车服务的表现;

(2)提出了一种新的基于强化学习的公平任务分配方案LAF,该方案适应高度动态的交通,符合实际设置,适合大规模的网约车应用;

(3)广泛的评估表明,本文的LAF在公平性、效用和效率方面远远优于最先进的文献<sup>[6,10]</sup>.

本文其余部分组织如下:第1节回顾相关工作,第2节介绍本文的问题,第3节解释本文的方法,第4节给出评估,第5节进行总结.

## 1 相关工作

本文的工作主要涉及两个研究方向:网约车中的任务分配和分配问题的公平性.

### 1.1 任务分配

由于打车服务在大城市的快速扩张,引起了广泛的研究兴趣,其中一个核心课题是为特定的优化对象向司机分配打车请求.网约车的共同目标包括最大化利润<sup>[3,11]</sup>、最小化旅行成本<sup>[9,12]</sup>、最小化乘客等待时间<sup>[1,5]</sup>等.许多任务分配算法关注于提供理论上的性能保证<sup>[5]</sup>,然而,他们经常会做出假设——比如司机和任务之间的独立性,这往往会阻止他们在现实世界的应用程序中实现预期的性能.一个很有前途的替代方案是,在很少的假设下,自动学习从历史数据中优化分配,例如,文献<sup>[13]</sup>最近的一些工作通过使用强化学习优化效用实现了最先进的性能.在这项工作中,本文也针对适合大规模应用的实际任务分配算法,但本文是第一个设计基于强化学习的解决方案,以优化效用和公平性.

### 1.2 分配任务的公平性

在各种应用中,公平一直是分配问题的一个重要因素<sup>[14,15]</sup>.关于公平分配的研究可以根据任务和员工是

静态的还是动态的而分为两类. 在静态公平任务分配中, 工作者和任务都是静态的; 动态公平分配比较困难, 因为其中一方或双方都是动态的, 这个设置适用于云计算<sup>[16]</sup>和Web请求分配<sup>[8]</sup>等应用程序, 它们的目标是平衡服务器的负载. 一些工作(如文献[17])研究了空间众包中的公平分配, 公平意味着员工承担相同数量或价值的任务, 这个公平的目标不适合网约车, 因为不同的出租车司机有不同的工作时间.

网约车公平属于动态公平分配问题, 特别令人感兴趣的是司机收入的公平性<sup>[6,10]</sup>. 由于任务和司机的强烈时空动态, 网约车中的公平分配具有挑战性. 在这项工作中, 本文提出了一种新的驾驶员收入公平指标, 该指标解释了这种时空动态, 并设计了高效且有效的分配算法, 其性能明显优于最先进的公平分配算法<sup>[6,10]</sup>.

## 2 问题陈述

与之前的研究(文献[6,10])一样, 本文考虑将司机批量分配给请求, 令 $W$ 和 $R$ 是感兴趣的时间范围 $T$ 中的司机和请求的集合, 在每个批次 $t$ 中, 该批次中可用的请求 $R(t)$ 和司机 $W(t)$ 之间的分配被表述为二部图匹配问题, 以优化效用和公平性, 批量设置在现实世界的大规模乘车应用中被广泛采用<sup>[2,13]</sup>.

定义1(请求). 一辆出租车请求 $r \in R$ , 表示为一个元组 $\langle o_r, d_r, p_r, \tau_r \rangle$ ,  $o_r, d_r, p_r, \tau_r$ 分别代表出发地、目的地、价格和持续时间.

在现实世界的网约车应用中, 请求的来源和目的地由乘客输入并上传到平台上, 本文对请求的来源和目的地不作任何假设, 平台收到请求后, 根据行程距离和交通拥堵等因素, 确定价格并估计请求持续时间, 为简单起见, 本文假设持续时间是 $t$ 的倍数, 如果请求被分配给一个司机, 则比值 $p_r/\tau_r$ 决定他的每批收益, 本文将在下面解释.

定义2(司机). 一个出租车司机 $w \in W$ 表示为一个元组 $\langle \ell_w^{(t)}, u_w^{(t)}, a_w^{(t)} \rangle$ ,  $\ell_w^{(t)}, u_w^{(t)}, a_w^{(t)}$ 分别代表当前位置、收入和第 $t$ 批次的状态.

状态 $a_w^{(t)}$ 为0或1的二进制指示灯,  $a_w^{(t)}=0$ , 代表司机处于非活动状态, 即不在平台上;  $a_w^{(t)}=1$ 代表司机是活动的, 它可以是空闲的或服务一个请求. 本文假设 $a_w^{(t)}$ 在一个批处理中保持不变, 因为实际批处理的大小很小(例如2s). 如果司机处于空闲状态, 则收益为 $u_w^{(t)}=0$ , 如果批量发送请求 $r$ , 则收益为 $a_w^{(t)}=p_r/\tau_r$ . 不像先前的

研究<sup>[9,5]</sup>, 假设独立的司机跨批分布, 本文考虑一个更现实的依赖司机设置, 即未来批次的司机分布受当前批次司机分布和当前分配的影响.

定义3(二部图). 本文使用二部图 $G^{(t)} = \langle R^{(t)}, W^{(t)}, E^{(t)} \rangle$ , 表示候选人之间的作业司机和第 $t$ 批次的请求, 节点集 $R^{(t)}$ 请求被分配到第 $t$ 批次的可用司机 $W^{(t)}$ . 如果请求 $r$ 可以分配给司机 $w$ , 则有一个边 $(r, w) \in E^{(t)}$ , 带有一个权重 $\theta_{r,w}$ .

根据之前关于网约车<sup>[11]</sup>中任务分配的研究, 为了避免乘客等待时间过长, 只有当请求-司机距离在某一阈值内时才存在优势. 本文还为每条边 $(r, w)$ 指定了一个拒绝率 $\lambda_{r,w}$ , 以考虑其他与用户体验相关的因素, 距离阈值和拒绝率由平台设置, 权重 $\theta_{r,w}$ 初始设定为请求 $r$ 的价格 $p_r$ .

本文的目标是为每个批次 $t \in \{1, 2, \dots, T\}$ 找到一个匹配 $M^{(t)}$ 的候选分配 $G^{(t)}$ , 优化总效用和时间收入公平性, 定义如下.

定义4(总效用). 给定可重用的司机集 $W$ 和动态出现的请求集 $R$ , 总实用程序是所有司机在 $T$ 批之前的预期累积收益, 即:

$$U = \sum_{w \in W} E \left[ \sum_{t \in T} u_w^{(t)} \right] \quad (1)$$

其中,  $E[\cdot]$ 代表期望, 司机 $w$ 的期望累计收益 $E \left[ \sum_{t \in T} u_w^{(t)} \right]$ 由匹配结果决定:

$$E \left[ \sum_{t \in T} u_w^{(t)} \right] = \sum_{t \in T} \sum_{(r,w) \in M^{(t)}} (1 - \lambda_{w,r}) \cdot p_r \quad (2)$$

本文最大化 $U$ 以优化司机的整体收益.

采用摊销公平性<sup>[6]</sup>的概念来描述驾驶员之间的时间收益公平性, 摊销公平性是指驾驶者在活动时间 $\sum_{t \in T} t \in T^{u_w} / \sum_{t \in T} t \in T^{a_w}$ 期间的累计收益, 然而, 本文认为, 这种朴素的平摊公平可能对有不同工作时间偏好(例如白天、晚上或高峰时间)的司机不公平, 由于一天中不同时间的出租车需求不同, 潜在收益也不同, 因此, 实行与活动时间成比例的平等收入会导致司机之间的收入不平等, 例如主要在白天工作的司机和主要在晚上工作的司机. 作为补救措施, 本文建议使用如下定义的加权公平平摊.

定义5(加权公平平摊). 驾驶员 $w$ 的加权摊销公平性为其累计收益除以其加权活跃时间:

$$F_w = \frac{\sum_{t \in T^{u_w^{(t)}}} \xi^{(t)}}{\sum_{t \in T^{a_w^{(t)}}} \xi^{(t)}} \quad (3)$$

其中,  $\xi^{(t)}$  是与收益相关的权重, 用于将一天中不同时间段的潜在收益变化正常化, 通常, 关注的时间范围  $T$  是一天, 批次  $t$  以秒为刻度 (例如 2 s), 时间权重  $\xi^{(t)}$  以每小时为基础变化, 这是因为报告显示, 出租车司机在决策时关注小时和日工资, 研究表明出租车需求按小时大幅波动. 为简单起见, 本文使用包含批次  $t$  的当前小时内司机收入的中位数为  $\xi^{(t)}$ .

为了量化各司机间的公平性分布, 本文定义了基于权重摊销的时间收益公平性, 如定义 6 所示.

定义 6 (时间收益公平). 给定可重用的司机集  $W$  和动态出现的请求集  $R$ ,  $W$  之间的时间收益公平性由加权平摊公平性的熵变量来度量:

$$F = - \sum_{w \in W} \log \left( \frac{F_w}{\max_{w \in W} F_w} \right) \quad (4)$$

较大的  $F$  表示驾驶员之间的分散加权平摊公平, 即高收入不平等. 因此, 本文的目标是最小化  $F$ .

对大型网约车应用的总效用  $U$  和时间收益公平性  $F$  进行分析是具有挑战性的, 以往的解决方案<sup>[6,10]</sup> 有以下局限性:

(1) 忽视当前和未来作业之间的时间依赖性: 许多现有的建议通过假设新司机在时间范围内独立到达来简化问题设置. 然而, 司机在下一批中的可用性和位置会受到当前批中的分配的影响, 这种依赖关系会影响效用和公平性的优化, 特别是在短期内, 出租车的供需可能会在空间和时间上发生剧烈波动.

(2) 对于大规模应用程序来说效率低下: 以往的研究采用线性规划的方法来解决双目标匹配问题, 这可能会导致大规模叫车的实时响应速度较慢.

### 3 建议的方法

在本节中, 介绍了 LAF 方法, 这是一种新的解决方案, 用于优化大规模网约车应用的总效用和时间收入公平. LAF 采用一种基于强化学习的重新加权方案, 明确地考虑了匹配过程中当前和未来任务之间的时间依赖性. 该方案采用在线方式实现, 以适应出租车供需的动态变化. LAF 还包含了一组剪枝和加速策略, 用于大规模数据上的高效双目标 (效用和公平性) 分配. 本

文在第 3.1 节对 LAF 进行了概述, 并在第 3.2 节和第 3.3 节详细阐述了细节.

#### 3.1 公平分配学习 (LAF) 概述

LAF 由一个基于学习的重新加权模块和一个有效的双目标分配模块组成, 见图 2.

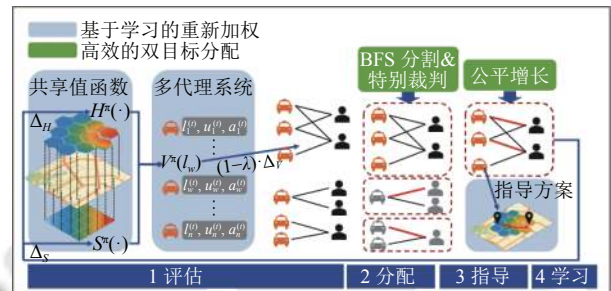


图 2 LAF 工作流程

在每个批次中, 基于学习的重新加权模块首先考虑分配之间的时间依赖性, 重新调整给定二部图的边权重, 然后高效的双目标分配模块通过公平增强和其他加速技术找到分配, 最后, 基于学习的重新加权模块根据分配结果更新权重, 并在必要时引导空闲驾驶员.

基于学习的重新加权模块 (第 3.2 节) 细化二部图中的边权重, 这些边权重被初始化为旅行价格 (见定义 3), 以反映当前分配对未来效用和公平性的影响, 权重细化策略是通过在线强化学习获得的, 本文还设计了一个驾驶员指导方案来缓解在线学习的冷启动问题.

高效的双目标分配模块 (第 3.3 节) 将出租车请求分配给司机, 同时考虑实用性和公平性, 它的核心是公平增强算法, 它应用 Kuhn-Munkres 算法<sup>[18]</sup> 来最大化效用并检查司机的收入率以确保公平.

在每个批次中, LAF 分 4 个阶段运行: 评估、分配、指导和学习 (如图 2), 每个批次都从评估开始, 给定一个边权重由旅行价格初始化的二部图, 基于学习的重新加权模块将更新权重, 以便权重反映当前和未来的收益, 然后在分配阶段, 高效的双目标分配模块在细化的二部图上计算新的匹配, 同时考虑效用和公平性. 由于: (1) 更新的权重考虑了当前分配对未来分配的影响; (2) 分配算法是双目标的, 因此产生的分配以整体方式优化了效用和公平性; 最后是学习和指导阶段. 重新加权模块将从匹配结果中学习以改进其下一批的重新加权策略, 并引导空闲驾驶员到繁忙区域以避免在线学习的冷启动.

### 3.2 基于学习的重新加权

本小节解释了如何应用在线强化学习来模拟当前分配对未来效用和公平性的影响,第3.2.1节介绍基本公式,第3.2.2节至第3.2.4节讨论实际问题,第3.2.5节介绍完整的设计。

#### 3.2.1 基于在线强化学习的公式

强化学习<sup>[8]</sup>是一种智能体,随时间与环境交互的学习方法,代理每一步都采取行动并从环境中获得奖励,基于奖励,代理更新其价值函数 $V_w^\pi$ ,如果代理按照策略 $\pi$ 采取行动,该函数将从代理 $w$ 的状态映射到预期的累积奖励,可以通过评估与环境交互获得的奖励来学习最优策略。

在本文的上下文中,每个主动司机都被视为一个代理,在每一批,一个代理(即司机)可以采取两种行动,接受出租车请求 $r$ 或保持空闲,奖励分别为 $p_r$ 或 $0$ 。司机 $w$ 的状态由他的位置代表 $\ell_w^{(t)}$ 和 $t$ 批次处理为 $(\ell_w^{(t)}, t)$ 。本文对司机 $w$ 的价值函数 $V_w^\pi(\ell_w^{(t)}, t)$ 感兴趣,其状态 $(\ell_w^{(t)}, t)$ 遵循优化总效用和时间收益公平性的策略 $\pi$ ,由于公平目标和司机之间的潜在行为冲突,明确推导出策略 $\pi$ 具有挑战性,因此,本文在不考虑策略的情况下从奖励中迭代更新价值函数,如下所示:

$$V_w^\pi(\ell_w^{(t)}, t) \leftarrow V_w^\pi(\ell_w^{(t-1)}, t-1) + \beta \cdot \Delta w \quad (\forall w \in W) \quad (5)$$

其中, $\beta$ 是学习率, $\Delta w$ 由以下因素决定:

$$\Delta w = \begin{cases} p_r + V_w^\pi(d_r, t + \tau_r) - V_w^\pi(\ell_w, t), & \text{如果 } w \text{ 获取 } r \\ 0, & \text{如果 } w \text{ 是空闲的} \end{cases} \quad (6)$$

由于本文的目标不是明确的策略 $\pi$ ,为了简洁起见,在其余的论文中省略了上标 $\pi$ 。另请注意,本文采用在线强化学习模型来适应城市交通的短期动态,将交通模式预测与强化学习相结合超出了本文的范围。

#### 3.2.2 减少状态数

上面的公式有太多的状态供智能体探索,这阻碍了有效的强化学习。简单的状态离散化是不够的,状态总数是空间状态数 $N_s$ 乘以时间状态数 $N_T$ ,考虑将一个城市划分为 $1 \text{ km}^2$ 的方块,将一天的时间范围划分为跨度为 $20 \text{ min}$ (平均旅行持续时间)的片段,那么 $N_s$ 和 $N_T$ 将分别约为 $8000$ 和 $72$ ,导致超过总共 $500000$ 个方块,这超出了代理在一天 $25200$ 次操作中的探索,假设活动时间为 $14 \text{ h}$ (允许的最大工作时间)并每 $2 \text{ s}$ 探索一个状态(一批),如果排除服务请求的时间,实际探索次数要低得多。

本文通过两种方法减少状态数:

(1) 空间值近似函数: 本文将时空状态空间中的原始价值函数近似为空间状态空间,即 $V_w(d_r, t + \tau_r) = V_w(d_r, t)$ ,这是合理的,因为大多数请求持续时间不到半小时,价值函数的变化可以忽略不计。因此,更新方程可以改写为:

$$\Delta w = \begin{cases} p_r + \gamma^{\tau_r} V_w(d_r) - V_w(\ell_w), & \text{如果 } w \text{ 获取 } r \\ 0, & \text{如果 } w \text{ 是空闲的} \end{cases} \quad (7)$$

折扣因子 $\gamma$ 纠正了长时间请求的近似误差。

(2) 代理之间的信息共享: 使用一个结合所有代理的价值函数的共享价值函数,这是合理的,因为处于相似时空状态的驾驶员应该对位置具有相似的评估,因此,更新方程进一步简化为:

$$V(\ell) \leftarrow V(\ell) + \beta' \cdot \sum_{w: \ell_w^{(t)} \in \ell} \Delta w \quad (8)$$

其中, $\beta'$ 是归一化的学习比率,而 $\ell$ 代表价值函数的所有可能位置。

第1种方法将状态总数从 $N_T \cdot N_s$ 减少到 $N_s$ ;第2种方法使所有司机探索相同的价值函数以扩展探索,由于城市中司机的数量可能在 $10000$ 的数量级,每个方块平均会被探索一百次,这足以收敛。

#### 3.2.3 适应不同的城市布局

在实际的网约车应用中,空间通常是离散的<sup>[2,13]</sup>,这限制了价值函数在城市布局上的变化,为了消除这些限制,本文从不同方向对价值函数进行平滑,这包括两个步骤:(1)将位置离散为一个两层结构;(2)对不同层值进行平滑。

在LAF中,这两层分别是六边形层和正方形层,城市在这两层中分别被分割成六边形和正方形。这两种形状提供了不同的平滑特性:六边形有更多的方向,通过这些方向的平滑有利于不规则的城市布局;正方形的边界与经纬度平行,适用于规则区域。如图3所示,六边形层为与主干道形状一致的径向模式(图3(a)),而正方形层为区域模式(图3(b)),表示部分繁忙区域。

最后,本文将价值函数平滑如下:

$$V(\ell) = \frac{1}{|DIR_H| + |DIR_S|} \left( \sum_{x \in DIR_H} H(\ell + x) + \sum_{x \in DIR_S} S(\ell + x) \right) \quad (9)$$

其中, $H$ 和 $s$ 为六边形和正方形的价值函数,由式(8)更新, $DIR_H$ 和 $DIR_S$ 为两层的平滑方向偏移量。

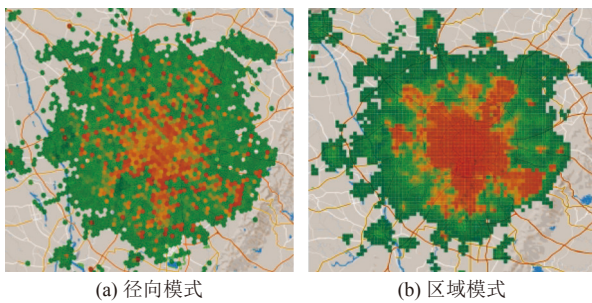


图3 下午 17:00 的六边形和正方形值函数 ( $H$  和  $s$  中间较暖的颜色表示较高的值)

### 3.2.4 在线学习中避免冷启动

由于在线学习的方式, 价值函数被初始化为零, 导致值评估退化为初始权值 (出行价格), 这样的冷启动禁止初始批的分配是未来感知的, 为此, 本文建议提前指引司机前往适当地点, 适当的区域由价值函数的距离和增量决定 (即算法 3 中的第 4 行). 为简单起见, 本文使用价值函数  $H$  中的六边形区域 ( $A_h$ ) 作为候选的引导目的地.

### 3.2.5 再加权模块

基于学习的再加权模块由评估、学习和指导 3 个阶段组成. 算法 1、算法 2 和算法 3 分别说明了过程, 在算法 3 的第 4 行中, 函数  $dist$  计算距离. 重新加权模块利用强化学习明确考虑当前作业对未来作业的影响, 它不会直接促进公平. 在 LAF 中, 公平目标的优化嵌入到总效用的优化中, 将在下面解释.

#### 算法 1. 重新加权模块的评估阶段

输入: 二部图  $G(t) = \langle W^{(t)}, R^{(t)}, E^{(t)} \rangle$ , 拒绝概率  $\lambda$ , 六边形  $H$  价值函数,  $S$  方值函数  
输出: 加权后的二部图  $G^{(t)} = \langle W^{(t)}, R^{(t)}, E^{(t)} \rangle$

```

 $E^{(t)} \leftarrow \emptyset$ 
for  $w, r, \theta_w, r \in E^{(t)}$  do
    calculate  $V(l_w)$  and  $V(d_r)$  by Eq(9)
     $\theta_{w,r} \leftarrow (1-\lambda_{w,r}) \cdot (p_r + \gamma_r V(d_r) - V(l_w))$ 
     $E^{(t)} \leftarrow E^{(t)} \cup (w, r, \theta_{w,r})$ 
end
Return  $E^{(t)}$ 
    
```

#### 算法 2. 重权模块的学习阶段

输入: 赋值结果  $M^{(t)}$ , 司机  $W^{(t)}$ , 六边形值函数  $H$ , 正方形值函数  $S$   
输出: 更新后的值函数  $H$  和  $S$

```

for  $w$  in  $W^{(t)}$  do
    if there exists an order  $r$ , s.t.  $(w, r) \in M^{(t)}$  then
         $\Delta H \leftarrow p_r + \gamma_r H(d_r) - H(l_w)$ 
         $\Delta S \leftarrow p_r + \gamma_r S(d_r) - S(l_w)$ 
    
```

```

end
else
     $\Delta H \leftarrow 0, \Delta S \leftarrow 0$ 
end
 $H \leftarrow H + \beta \cdot \Delta H, S \leftarrow S + \beta \cdot \Delta S$ 
end
return  $H, S$ 
    
```

#### 算法 3. 重称重模块引导阶段

输入: 空闲司机  $w_i^t$ , 候选目的地  $A_h$ , 从第 1 批到当前批  $t$  的权重  $\zeta$   
输出: 指导方案  $\delta$

```

 $\delta \leftarrow \emptyset$ 
基于  $\frac{\sum_{i=1}^t a_w^{(i)} / \zeta^{(i)}}{\sum_{i=1}^t a_w^{(i)}}$  递增排序司机
for  $w \in W^{(t)}$  do
     $g \leftarrow \arg \max_{g' \in A_h} \frac{V(g') - V(l_w)}{dist(g', l_w)}$ 
     $\delta \leftarrow \delta \cup (w, g)$ 
end
Return  $\delta$ 
    
```

## 3.3 有效的双目标分配

本节介绍了优化效用和公平性的分配算法.

### 3.3.1 公平增长

不像以前的双目标解决方案, 要么依赖于缓慢的线性规划<sup>[16]</sup>, 要么在公平和效用<sup>[5]</sup>的单独优化之间进行重复的重新分配, 本文建议直接将公平检查嵌入到过程中, 以最大化更快的分配的效用.

在二部图上寻找效用最大化的匹配的标准方法是 Kuhn-Munkras 算法<sup>[8]</sup>, 其中的基本操作是增长, 即首先尝试找到一条由匹配边和不匹配边交替组成的路径, 而未匹配边的权值之和大于已匹配边的权值之和, 如果发现, 则将未匹配边和已匹配边进行切换, 以增加总效用. 本文的公平增长算法的思想是, 在寻找增长路径时, 检查驾驶员的未来收益率, 拒绝收益率方差较大的增长, 然而, 拒绝增长可能会损害已实现的效用, 考虑一个新上线的司机在增长路径上, 由于他的空闲时间几乎为零, 所以他在服务请求后的收益比会非常大, 导致增长频繁被拒绝. 还要注意, 在找到一个扩展路径后, 检查所有司机的方差是非常耗时的, 因此, 本文不检查所有的司机, 只检查增长路径中相邻的两个司机, 一旦发现方差比, 立即中断并停止增长. 算法 4 说明了公平的增长算法, 时间复杂度为  $O(N^2M)$ , 其中,  $M = \max(|W(t)|, |R(t)|^2, N = \min(|W(t)|, |R(t)|)$ .

### 3.3.2 加速度

算法 5 利用二部图的稀疏性, 进一步加速了均匀

增长算法,具体地说,执行广度优先搜索(BFS),将as分配图分成几个部分,其中很多部分只包含一个司机(请求),所以本文使用特殊的判断来快速地为单个节点找到最佳的请求(司机).

#### 算法4. 公平增长

输入: 二部图  $G^{(t)} = \langle W^{(t)}, R^{(t)}, E^{(t)} \rangle$ , 当前批  $t$ , 权重  $\xi$ ,  $t$  从第一批

输出: 作业  $M^{(t)}$

```

 $M^{(t)} \leftarrow \emptyset$ 
for  $w \in W^{(t)}$  do
  while there exists an augment Path  $P$  do
     $accept \leftarrow \text{True}$ 
    for every pair of adjacent drivers  $w_{pre}, w_{cur}$  do
       $r_{pre} \leftarrow$  current assigned request of  $w_{pre}$ 
       $r_{cur} \leftarrow$  current assigned request of  $w_{cur}$ 
      if  $\left| \frac{\sum_{i=1}^t \frac{u_{w_{pre}}^{(i)}}{\xi^{(i)}} + \frac{pr_{pre}}{\xi^{(t)}}}{\sum_{i=1}^t a_{w_{pre}}^{(i)} + \tau r_{pre}} - \frac{\sum_{i=1}^t \frac{u_{w_{cur}}^{(i)}}{\xi^{(i)}} + \frac{pr_{cur}}{\xi^{(t)}}}{\sum_{i=1}^t a_{w_{cur}}^{(i)} + \tau r_{pre}} \right| > \epsilon$ 
        then
           $accept \leftarrow \text{False}$ 
          break
        end
      end
    end
    If  $accept$  then
      Update  $M^{(t)}$  based on  $P$ 
    end
  end
end
Return  $M^{(t)}$ 

```

#### 算法5. 分配算法

输入: 二部图  $G^{(t)} = \langle W^{(t)}, R^{(t)}, E^{(t)} \rangle$

输出: 作业  $M^{(t)}$

```

 $P^{(t)} \leftarrow$  split  $G^{(t)}$  by BFS.
 $M^{(t)} \leftarrow \emptyset$ 
for  $p \in P^{(t)}$  do
  if there is only one driver  $w$  (request  $r$ ) then
     $M_p \leftarrow \langle w(\text{best driver}), \text{best request}(r) \rangle$ 
  end
  else
     $M_p \leftarrow$  Fair Augmentation( $p$ )
  end
end
 $M^{(t)} \leftarrow M^{(t)} \cup M_p$ 
end
return  $M^{(t)}$ 

```

## 4 评价

### 4.1 实验设置

验证环境: 在一个大型网约车平台的模拟器上进

行了实验,在3个城市进行了21天的实验,该模拟器生成请求,模拟驾驶员和乘客的行为(即空闲驾驶员转换和乘客拒绝),构建二部图,并执行本文的分配算法.

基线: 将本文的方法与以下基线进行比较:

(1) 距离贪婪(DG): 它将每个请求分配给它最近的可用司机,这是一个广泛使用的基线叫车.

(2) 收益比率贪婪(ERG): 它首先按加权摊销公平性  $F_w$  按降序对所有可用司机进行排序,然后按每批次的奖励  $(p_r / \tau_r)$  降序对请求进行排序,然后它根据顺序将请求分别分配给司机.

(3) 整数线性规划(ILP)<sup>[6]</sup>: 这是第一个应用摊销公平性来评估网约车收益公平性的工作,它将问题转换为整数线性规划问题.为了公平比较,本文从其优化目标中省略了乘客侧的公平性.

(4) 重新分配(REA)<sup>[10]</sup>: 它首先分别计算优化效用和公平性的两个分配(调整为  $\min_w F_w$  以进行相等比较),然后它将匹配从一个分配重新分配到另一个分配,以找到效用和公平之间的权衡.这是最先进的解决方案,可优化网约车服务的实用性和公平性.

评价指标: 本文通过公平、效用和效率来评估不同方法的性能,其中,公平是通过时间收益公平(定义6)来衡量;效用是通过总效用(定义4)来衡量;效率是通过每小时累积的分配消耗时间来衡量的.

实现: 所有算法都是在 Python 3 中实现的,本文将折扣因子  $\gamma$  (式(7)) 设为 0.9, 学习率  $\beta'$  (式(8)) 设为 0.025. 实验在 Intel Xeon CPU E5-2630 v4 @ 2.20 GHz, 12 GB 内存上进行.

### 4.2 总体性能

表1总结了3个城市数据集上不同分配方法的公平性和效用指标,总体而言,LAF在工作日比基线提高了45.7%–81.4%,在周末提高了52.0%–86.7%.LAF总效用最高,工作日比基线高7.7%–29.1%,周末比基线高11.3%–28.5%.特别是,本文的LAF在公平性和效用方面分别比ILP平均高出68.4%和22.1%.对于REA,LAF在公平性和效用上的平均提高分别为69.3%和20.1%.此外,LAF在这3个城市的表现都优于基线.

图4比较了不同算法的效率,简单基线DG和ERG的时间复杂度分别为  $O(|E| \log |E|)$  和  $O(|W| \log |W|)$ ,但在公平方面表现不佳,在周末,公平和效用方面,本文这两个简单的解决方案几乎一样快,LAF的运行时间相对稳定,ILP和REA比LAF慢797%.此外,他

们的运行时间是敏感的变化, 交通在高峰期, 当请求数量急剧增加时, 这两种方法的时间成本也会显著增加。

表 1 不同分配算法的公平性  $F$  和效用  $U$  的总体结果

城市	方法	$F$ (工作日)	$U$ (工作日)	$F$ (周末)	$U$ (周末)
A	DG	41 767	2 242 977	31 962	2 131 996
	ERG	48 549	2 256 277	35 903	2 172 489
	ILP <sup>[16]</sup>	44 072	2 246 077	30 573	2 122 834
	REA <sup>[5]</sup>	44 251	2 250 118	27 865	2 134 534
	<b>LAF(本文)</b>	<b>22 656</b>	<b>2 656 773</b>	<b>13 384</b>	<b>2 565 060</b>
B	DG	31 217	1 297 735	28 563	1 360 659
	ERG	38 594	1 373 478	37 464	1 452 488
	ILP <sup>[16]</sup>	29 496	1 281 273	25 077	1 336 712
	REA <sup>[5]</sup>	31 744	1 319 958	33 671	1 373 691
	<b>LAF(本文)</b>	<b>7 420</b>	<b>1 479 348</b>	<b>4 976</b>	<b>1 616 385</b>
C	DG	12 897	872 693	9 044	865 031
	ERG	13 704	862 588	8, 439	863, 245
	REA <sup>[5]</sup>	13 491	880 847	8 684	881 519
	<b>LAF(本文)</b>	<b>2 553</b>	<b>1 114 000</b>	<b>2 392</b>	<b>1 109 474</b>

### 4.3 实验结果分析

现在深入研究不同方法在 3 个城市中规模最大的城市 A 的数据集上的性能, 以进一步了解本文方法的有效性。

#### 4.3.1 LAF 与其他算法的差异

为了说明不同分配算法之间的差异, 本文在图 5 中绘制了驾驶员的轨迹。

(1) DG 不顾公平, 将最近的请求分配给司机, 第

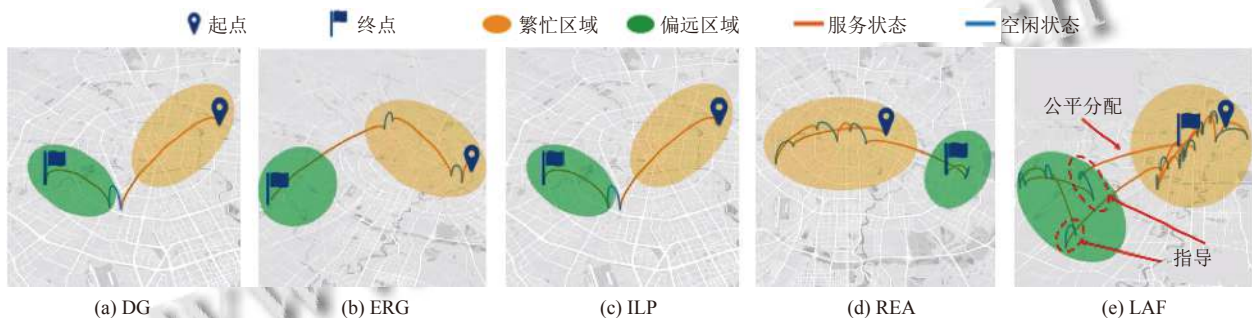


图 5 不同分配算法下的驾驶员轨迹

(4) 本文的 LAF 通过考虑它们对未来任务的影响来进行任务, 该指导还避免了司机被困在偏远地区。如图 5(e) 所示, 在繁忙区域服务了一些请求后, 司机的收益比相对较高, 然后, 为了公平起见, 将请求分配给其他驾驶员, 然后, 为了避免长时间的闲置, 司机会被引导到他/她可以得到请求的区域, 然后返回到繁忙的区域。

#### 4.3.2 流量动态与公平性的相关性

本文的贡献之一是使用加权平摊公平  $F_w$  作为公

2 次分配后, 驾驶员被困在偏远地区, 无法返回进行其他请求 (图 5(a)); ERG 将收益比率最低的驾驶员分配给最高的  $p_r/\tau_r$ , 在第 2 个请求后, 司机的收益比较低, 并被分配到一个长途请求 (图 5(b)).

(2) ILP 对这个司机做了与 DG 相同的分配 (图 5(c)), 这是因为司机在完成长途请求后, 其收益比较高, 导致下一批请求的距离较近, 而在偏远地区, 由于两次服务后的收入比较高, 司机很难再接到另一次服务。

(3) REA 兼顾了公平性和实用性, 实用程序的优化增加了分配给司机的请求数量 (图 5(d)), 但是, REA 仍然不知道目的地, 司机只能选择一个短距离的请求, 可能被困在偏远地区。

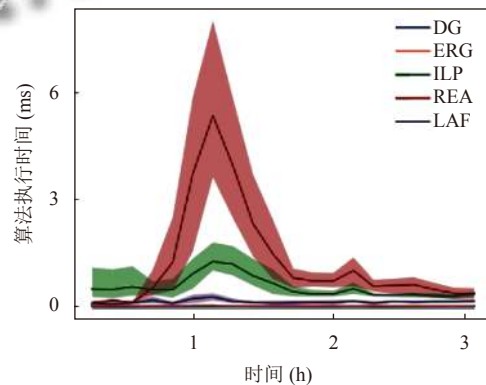


图 4 每个算法在 A 城市每天的执行时间

平度量 (定义 5), 通过显示  $F_w$  中的权重  $\zeta^{-1}(t)$  与流量动态之间的相关性来证明本文的加权平摊公平度量的优势。

图 6(a) 绘制了一天中不同时间内司机和请求的数量以及权重  $\zeta^{-1}(t)$ , 司机和请求的数量反映了交通的变化, 如图 6 所示, 交通流量从早上 6 点到 8 点上升, 白天波动, 晚上下降, 权重  $\zeta^{-1}(t)$  也呈现出同样的趋势。这表明本文在高峰时段更注重公平, 因为那里有更多的司机, 因此需要更多的公平。还需要注意的是, 司机在



当前小时内的收入中值,如 $\xi$ 是一个不错的选择,因为它的变化略早于交通,这是因为有经验的司机往往在高峰时间之前更早开始活跃,导致收入中值提前下降,因此权重 $\xi$ 可以预见地增加。

图6(b)显示了随机抽样的3500名司机的时薪,如图所示,以公平性度量作为优化目标时,驾驶员的时薪分布相对均匀,而LAF算法在减少时薪极高/极低的驾驶员数量方面表现最好。

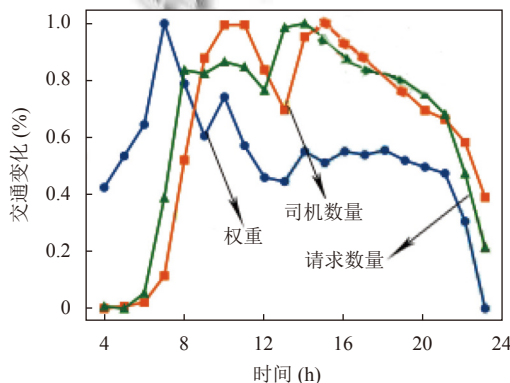
#### 4.3.3 优化未加权平摊公平的有效性

表2比较了优化未加权平摊公平性(即在所有时间 $t$ 内将 $\xi^{(t)}$ 设置为1)和效用的不同算法,DG被排除在外是因为它不公平,算法的效用结果与表2相同。结果表明,尽管ILP和REA的设计是为了优化未加权平摊公平性,但LAF的公平性提高了57.9%–62.2%,效用提高了17.3%–17.6%。因此,本文的LAF在未加权和加权摊销公平性方面都优于目前的水平。

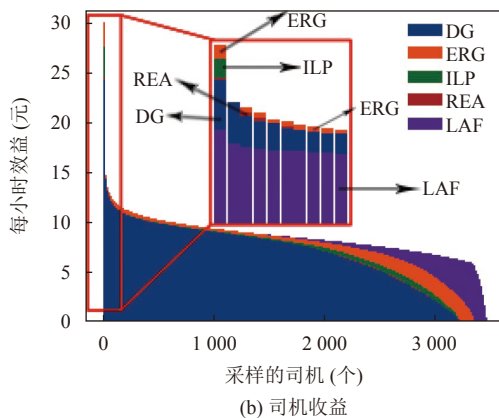
效地优化了总效用(驾驶员的预期总收入)和驾驶员的公平性(驾驶员收入的加权摊销公平性)。关键的创新之处在于应用强化学习来进行分配,这些分配明确说明了分配之间的依赖性,从而可以从整体角度优化效用和公平性。LAF还结合了一组技术来适应交通动态和不同的城市布局,并在大规模数据上进行快速分配。实验结果表明,LAF在公平性、实用性和效率方面比最先进的网约车公平分配算法提高了86.7%、29.1%和797%。此工作将成为在现实世界网约车应用中实际采用公平任务分配的指南。

表2 未加权平摊公平性与效用的比较

方法	$F$ (工作日)	$U$ (工作日)	$F$ (周末)	$U$ (周末)
DG	44 442	2 242, 977	33 057	2 131 996
ERG	54 270	2 248 669	38 315	2 166 980
ILP	40 066	2 243 334	30 772	2 120 947
REA	42 003	2 242 561	31 832	2 130 431
LAF(本文)	20 539	2 359 431	12 963	2 447 596



(a) 司机和请求的变化引起交通的变化



(b) 司机收益

图6 不同时间司机和请求的数量

#### 参考文献

- 1 Anthony BM, Chung C. Online bottleneck matching. *Journal of Combinatorial Optimization*, 2014, 27: 100–114. [doi: 10.1007/s10878-012-9581-9]
- 2 Tang XC, Qin ZW, Zhang F, *et al.* A deep value-network based approach for multi-driver order dispatching. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Anchorage: ACM, 2019. 1780–1790.
- 3 Wang YS, Tong YX, Cheng L, *et al.* Adaptive dynamic bipartite graph matching: A reinforcement learning approach. *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. Macao: IEEE, 2019. 1478–1489.
- 4 Tong YX, She JY, Ding BL, *et al.* Online minimum matching in real-time spatial data: Experiments and analysis. *Proceedings of the VLDB Endowment*, 2016, 9(12): 1053–1064. [doi: 10.14778/2994509.2994523]
- 5 Xu P, Shi YX, Cheng H, *et al.* A unified approach to online matching with conflict-aware constraints. *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. Honolulu: AAAI, 2019. 2221–2228.
- 6 Sühr T, Biega AJ, Zehlike M, *et al.* Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Anchorage: ACM, 2019. 3082–3092.

## 5 结论

在本文中,提出了公平分配学习(LAF)方法,它有

- 7 Bokányi E, Hannák A. Understanding inequalities in ride-hailing services through simulations. *Scientific Reports*, 2020, 10(1): 6500. [doi: [10.1038/s41598-020-63171-9](https://doi.org/10.1038/s41598-020-63171-9)]
- 8 Buchbinder N, Naor J. Fair online load balancing. *Proceedings of the 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures*. Cambridge: ACM, 2006. 291–298.
- 9 Tong YX, Chen YQ, Zhou ZM, *et al.* The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Halifax: ACM, 2017. 1653–1662.
- 10 Lesmana NS, Zhang X, Bei XH. Balancing efficiency and fairness in on-demand ridesourcing. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Vancouver: NeurIPS, 2019. 5310–5320.
- 11 Zhang LY, Hu T, Min Y, *et al.* A taxi order dispatch model based on combinatorial optimization. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Halifax: ACM, 2017. 2151–2159.
- 12 Wang X, Agatz N, Erera A. Stable matching for dynamic ride-sharing systems. *Transportation Science*, 2018, 52(4): 850–867. [doi: [10.1287/trsc.2017.0768](https://doi.org/10.1287/trsc.2017.0768)]
- 13 Xu Z, Li ZX, Guan QW, *et al.* Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London: ACM, 2018. 905–913.
- 14 Garfinkel RS. Technical note—An improved algorithm for the bottleneck assignment problem. *Operations Research*, 1971, 19(7): 1747–1751. [doi: [10.1287/opre.19.7.1747](https://doi.org/10.1287/opre.19.7.1747)]
- 15 Lorenz MO. Methods of measuring the concentration of wealth. *Publications of the American Statistical Association*, 1905, 9(70): 209–219. [doi: [10.2307/2276207](https://doi.org/10.2307/2276207)]
- 16 Kash I, Procaccia AD, Shah N. No agent left behind: Dynamic fair division of multiple resources. *Journal of Artificial Intelligence Research*, 2014, 51: 579–603. [doi: [10.1613/jair.4405](https://doi.org/10.1613/jair.4405)]
- 17 Chen Z, Cheng P, Chen L, *et al.* Fair task assignment in spatial crowdsourcing. *Proceedings of the VLDB Endowment*, 2020, 13(12): 2479–2492. [doi: [10.14778/3407790.3407839](https://doi.org/10.14778/3407790.3407839)]
- 18 Edmonds J, Karp RM. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 1972, 19(2): 248–264. [doi: [10.1145/321694.321699](https://doi.org/10.1145/321694.321699)]