迈向容量感知的经纪人匹配：从推荐到分配

Shuyue Wei[1], Yongxin Tong[1], Zimu Zhou[2],Qiaoyang Liu[1],Lulu Zhang[3],Yuxiang Zeng[1],Jieping Ye[3]
[1] 软件开发环境国家重点实验室，北京航空航天大学，中国北京[2] 香港城市大学，中国香港 [3] 贝壳控股有限公司，中国北京[1] {weishuyue, yxtong, qiaoyangliu, turf1013}@buaa.edu.cn, [2] zimuzhou@cityu.edu.hk[3] {zhanglulu026,yejieping}@ke.com

arXiv:2303.03024v1 [cs.DB] 6 Mar 2023

摘要—**在线房地产平台日益受到欢迎，其中一个核心问题是为潜在的房屋交易匹配经纪人与客户。主流平台通过前*k* 名推荐来匹配经纪人。然而，我们通过广泛的数据分析发现，这种前*k* 名推荐往往会使顶级经纪人超负荷工作，从而显著降低其服务质量。本文提出将经纪人匹配从推荐转变为分配，以避免此类过载。为此，我们设计了基于上下文老虎机的学习型分配（LACB），这是一种数据驱动的容量感知经纪人匹配方案，可在线估计经纪人特定工作负载容量，并从全局视角为客户分配经纪人，以最大化整体服务质量。在来自某工业在线房地产平台的合成数据集和真实世界数据集上的大量评估验证了我们方案的效率和有效性。**

索引词—**容量感知、经纪人匹配、房地产平台**

I. 引言

Compass[1], 、Zillow[2] 和贝壳控股有限公司（又称贝壳）[3] 等在线房地产平台正日益采用数据驱动的方法，以改进业务并提升服务质量。这些平台的一项核心功能，是为有购房意向的客户匹配合适的经纪人[4] ，以提供后续服务 [1]。这类经纪人匹配目前采用的是 top-*k* 推荐 [2], [3]。以中国最大的在线房地产平台贝壳为例。当客户在平台应用中点击查看某套房屋的详细信息时，应用会推荐与该房屋相关的三位经纪人（见图1）。

通过对在线房地产平台进行广泛的数据分析，我们发现*(i)* 经纪人的工作负载容量有限，且*(ii)* Top-*k* 推荐机制会导致顶级经纪人过载，即过载现象，从而损害服务质量和平台的长期发展。具体而言，我们的研究表明，由于 Top-*k* 机制，如果经纪人每天回应超过 40 个客户请求，其注册率可能会从 14.3%∼27.5% 降至 2.5%∼17.8%（见 II-B 节）。这里

[1]https://www.compass.com/[2]https://www.zillow.com[3] https://www.ke.com[4]我们不区分代理人与经纪人，统称为经纪人。

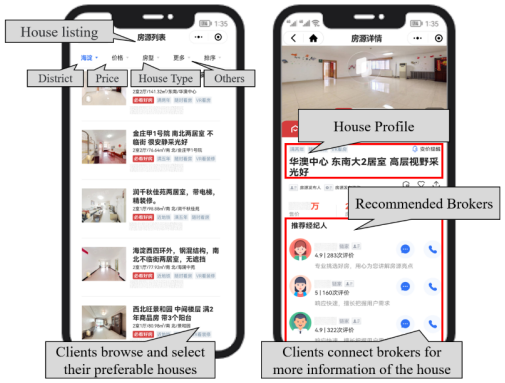


图1：贝壳这一大型在线房地产平台上的 top-*k* 经纪人推荐示意图。当购房者点击查看心仪房屋的更多信息时，平台应用默认推荐三位经纪人。

经纪人的注册率是服务质量的常见指标，计算方式为与其签约的客户数量占其服务过的客户总数的比例。我们还观察到，采用 Top-*k* 推荐机制时会出现马太效应 [4]。也就是说，许多请求被顶级经纪人占据，其他经纪人鲜有提升自身技能的机会。这可能会打击那些被忽视的经纪人，并从长远来看损害平台。

我们认为，过载现象是由于忽视经纪人的工作负载容量所致，这促使我们从分配 [5]–[7]视角来研究容量感知经纪人匹配。也就是说，我们不再盲目地向所有客户推荐少数几位顶级经纪人，而是提出先估算每位经纪人的工作负载容量，再从全局视角将经纪人分配给客户，同时避免经纪人超负荷。然而，在经纪人匹配中实施容量感知分配面临两项实际挑战。

- 挑战 *1*：如何以在线方式估计经纪人特定工作负载容量？我们观察到，不同经纪人的工作负载容量存在差异（见第 II 节），因此有必要进行个性化估计。然而，事先收集经纪人在所有可能工作负载下的服务质量数据并不可行，这使得在线估计工作负载容量更具优势。以往的工作负载容量估计方案 [8], [9] 未能

支持这种个性化估计的在线学习。

- 挑战 *2*：如何在容量限制下分配经纪人，以最大化长期总体效用？当前房地产交易量通常会影响近期的交易量。因此，不同批次之间的经纪人分配往往存在关联，难以从整体上分配经纪人。大多数分配方案 [10], [11] 会独立考虑每个批次中的客户和经纪人，因此在多个批次的集体效用方面并非最优。

为应对这些挑战，我们提出基于上下文老虎机的学习型分配（LACB），这是一种用于房地产经纪人匹配的数据驱动型容量感知分配方案。该方法通过上下文老虎机实现数据高效的在线个性化容量估计，从而应对挑战 *1*。LACB 通过容量感知价值函数克服挑战 *2*，该函数在匹配时兼顾经纪人的短期和长期效用。我们的主要贡献和结果概述如下。

- 据我们所知，我们首次发现了在线房地产平台中的顶级经纪人过载问题。大量数据分析表明，经纪人的工作负载容量有限，过载时其服务质量往往会下降，这促使经纪人匹配从推荐转向分配。

- 我们设计了 LACB，这是一种数据驱动、考虑容量的经纪人匹配分配方案。该方案在线估算每位经纪人的容量，并从全局视角为客户分配经纪人。我们还提出了 LACB-Opt，通过经纪人选择加速分配。
- 我们在合成数据集以及来自贝壳（中国最大的中国在线房地产平台）的真实数据集上进行了大量实验。实验结果验证了我们方案的效率和有效性。

本文其余部分安排如下：首先在第 II 节指出过载现象，并在第 III 节对问题进行形式化。随后在第 IV 节概述我们的解决方案，并分别在第 V 节和第 VI 节介绍各个模块。我们在第 VII 节展示评估结果，在第 VIII 节回顾相关工作，最后在第 IX 节总结全文。

II. 动机

我们通过对贝壳这一中国在线房地产平台的数据进行测量，阐述本研究的动机。我们观察到一种称为头部经纪人过载的现象：少数经纪人承担的服务请求量超出其容量，最终导致经纪人的服务质量和平台总体效用下降。

A. 经纪人容量有限

我们的首要动机是经纪人的容量有限。与其他服务行业一样，我们假设房地产经纪人的容量有限，即其在单位时间内能够以高质量提供的服务数量。由于房屋交易中的服务质量低下容易导致客户流失，我们假设经纪人的服务质量

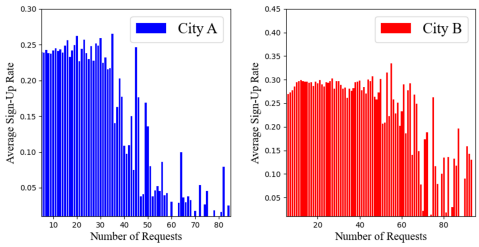


图2：两个城市经纪人的平均注册率。随着每日服务请求数增加，注册率往往会下降。我们通过以下测量结果检验这一假设。

测量。我们分析了中国两个主要城市的在线房地产平台在2021年6月1日至2021年8月31日期间的数据，以探究经纪人的服务质量与容量之间的关系。我们使用经纪人的注册率（即注册该经纪人的客户数量占其服务客户总数的比例）作为服务质量的代理指标。我们在城市和个人两个层面上，将注册率随工作负载增加的变化作为衡量指标；工作负载以每日服务请求数量表示（即每日服务请求数量）。

观察结果。我们观察到，随着工作负载增加，注册率往往会下降，而且下降模式似乎较为复杂，并且因经纪人而异。

- 图2展示了两个城市的经纪人平均注册率随每日服务请求数增加的变化情况。以A市（蓝色）为例。当每日服务请求数少于40时，平均注册率为14.3∼27.5%。一旦经纪人每天需要处理超过40个请求，其平均注册率便降至2.5∼17.8%。通过采用韦尔奇t检验，我们发现注册率与每日服务请求数量存在统计学上的显著相关性（p 值为 < 0.0001）。因此，经纪人过重的工作负载会降低服务质量，甚至导致客户流失。B市（红色）也呈现出类似的下降趋势。

- 我们进一步研究了在A市服务请求最多的排名前50的经纪人，其中有21人偶尔会服务超过40个请求。图3绘出了这21位工作负载较高的经纪人在A市的注册率。在这21位经纪人中，随着每日服务请求数量增加，他们的注册率呈下降趋势。
- 尽管呈下降趋势，但正如图2和图3所示，注册率与服务请求数量之间的关系往往复杂、非线性，并且因经纪人而异。

B. 过载的顶级经纪人

我们的第二个动机是，在当前的在线房地产平台中，由于采用了 top-k 推荐机制，顶级经纪人往往会过载。这是因为平台列出排名前 top-k 的经纪人时没有考虑他们的接待能力，而客户倾向于从平台列出的顶级经纪人中进行选择。我们将对此进行如下检验。

测量。我们分析了2021年6月A市同一在线房地产平台的数据，并绘制了

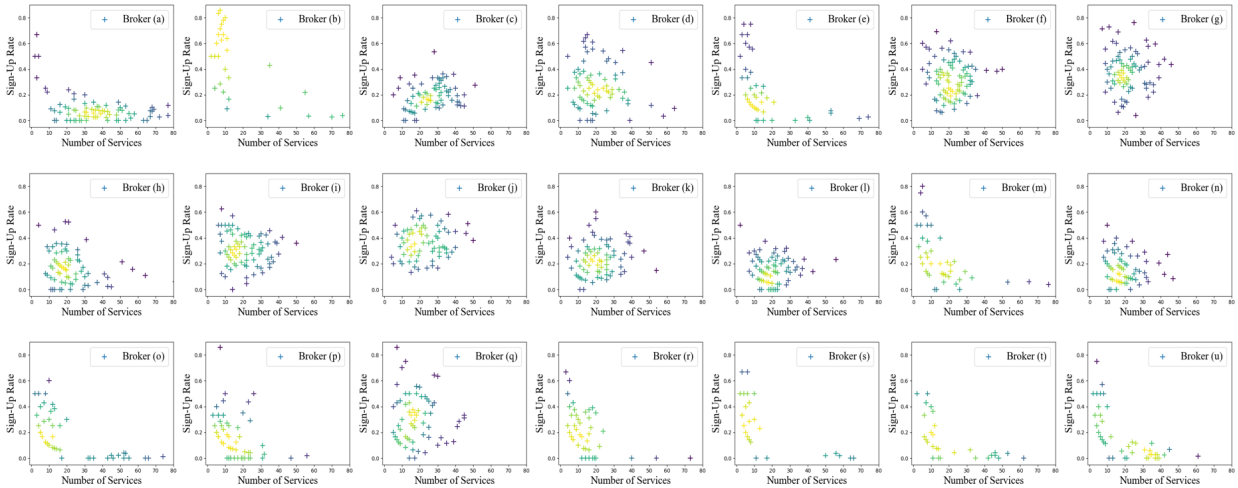


图3：A市顶级经纪人的平均注册率。我们使用高斯核密度估计来拟合经纪人绩效随工作负载变化的经验分布。绩效分布的中心区域颜色较浅，代表他们习惯的工作负载范围。与工作负载较高的区域相比，大多数顶级经纪人在工作负载较低的区域表现更好。

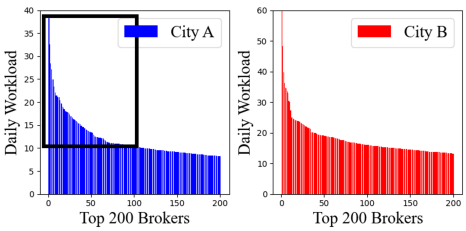


图4：A市和B市顶级经纪人的工作负载分布。人们倾向于听从推荐经纪人的建议，大多数请求都由这些顶级经纪人处理。

平台推荐的经纪人和未被平台列出的经纪人的工作负载分布明细。默认情况下，平台会推荐排名前三的经纪人（见图1）。

观察结果。 我们观察到，工作负载分布明显向平台推荐的排名前三的经纪人倾斜。在A市，排名第一的经纪人平均每天处理38.26个请求，而一名经纪人平均每天处理3.18个请求，即，排名第一的经纪人的工作负载是平均工作负载的12.03×倍。图4绘制了A市和B市排名前200的经纪人的工作负载分布。如图所示，他们的工作负载都明显高于城市平均工作负载。请注意，根据图3，工作负载为10至20时注册率更高。因此，黑箱中的约一百名经纪人面临超出其有限容量的风险。

此外，如果顶级经纪人不断被分派请求，可能会出现马太效应 。也就是说，大多数请求都由顶级经纪人承接，而其他经纪人则被忽视。因此，被忽视的经纪人很少有机会提升自己的找房技能，这会对平台的发展产生负面影响。在B市，顶级经纪人过载的情况也很常见。

C. 关键观察结果与见解

总而言之，我们观察到，过去在线房地产平台使用的 top-k 推荐机制往往会使顶级经纪人不堪重负，我们将其称为顶级经纪人过载问题。过载的经纪人服务质量会下降，最终导致注册率降低。过载问题的出现，是因为 top-k 推荐忽略了经纪人的容量。

过载现象促使我们从分配 的角度重新思考经纪人匹配。我们不再盲目地向所有客户推荐少数顶级经纪人，而是提出从全局视角为客户分配经纪人，同时考虑顶级经纪人的容量。接下来，我们将这些观点形式化为容量感知分配问题，并针对工作负载容量未知的情况，提出高效经纪人匹配的实用解决方案。

III. 问题陈述

本节将正式定义容量感知分配（CAA）问题，以避免经纪人匹配过程中顶级经纪人过载。

定义1 (经纪人)。 经纪人 $b \in B$ 是一个三元组$(\boldsymbol{x}_b, w_b, s_b)$，其中 $\boldsymbol{x}_b$、 w_b和 s_b 分别表示经纪人 b的属性、每日服务请求数量和每日签约率。

向量 $\mathbf{x}_b$ 包含工作年限、职位、平均对话轮次等属性（详情见表 II），反映了经纪人 b当前的工作状态。与第 II 节中的测量保持一致，我们使用每日服务请求数量 w_b 来量化经纪人的工作负载，并以每日注册率 s_b 作为其服务质量的代理指标。为便于表述，我们使用 $\mathcal{T}_b$ 表示经纪人 b的试验三元组，即 *i.e.*， $\mathcal{T}_b = \{(\mathbf{x}_1, w_1, s_1) ..., (\mathbf{x}_d, w_d, s_d)\}$。

在经纪人匹配中，经纪人负责处理客户请求。遵循房地产行业的惯例，我们以批量方式进行经纪人匹配，并采用注册率作为分配效用。我们采用固定时间窗口的批量分配，即平台预先设定时间区间，并为所有出现的请求分配经纪人。记时间区间 $i \in I$ 内出现的请求为 $R^{(i)}$。

从分配视角来看，经纪人匹配可以表述为以下问题。

定义 2（容量感知分配（CAA）问题）。给定一组经纪人 B、每个区间内的请求 $R^{(i)}$，以及它们之间的匹配效用 $u_{r,b}$（将经纪人 b 分配给请求 r的效用），我们的目标是在未知容量约束下，为每个时间区间 i 找到分配 $\mathcal{M}^{(i)}$，使总效用最大化。

- 最大化总效用。

$$\max \sum_{i \in I} \sum_{r,b} u_{r,b} \mathcal{M}_{r,b}^{(i)} \tag{1}$$

- 容量约束。

$$\forall b, \quad \sum_{i \in I} \sum_{r} \mathcal{M}_{r,b}^{(i)} \leq c_b \tag{2}$$

其中，$\mathcal{M}_{r,b}^{(i)} = 1$ 若经纪人 b 被分配给请求 r，则$\mathcal{M}_{r,b}^{(i)} = 0$ 否则。$u_{r,b}$ 表示将经纪人 b 分配给请求 r 所获得的效用，该值是输入，可通过 XGboost 等模型从历史分配中学习得到[12]。c_b 是待估计的经纪人 b 未知容量。表 I 总结了本文中使用的主要概念。

讨论。关于 CAA 问题，我们提出两点说明。

- 尽管批量分配建模已广泛应用于众包应用例如网约车服务 [10], [13],，但这是首次将其用于在线房地产平台上的经纪人匹配。

- CAA问题相较于一般批量分配所面临的一项独特挑战在于，经纪人容量 c_b并非预先给定。因此，在分配之前，有必要进行有效的容量估计。因此，我们旨在为在线房地产应用设计容量估计算法和分配算法。

- 其他知名应用还包括在线医疗（例如众安保险[5]）和法律咨询（例如律师馆[6]），由于工作人员服务所分配请求的工作负载容量有限，这些领域也存在类似的容量感知匹配问题。

表 I：主要符号汇总。

符号	描述
$\mathbf{x}_b, w_b, s_b, c_b$	经纪人 b的属性、工作负载、注册率和容量
$\mathcal{T}_b$	经纪人 b的试验三元组
I, i	时间间隔时域和时间区间 i
B, B_+	所有经纪人的集合和具有剩余容量的经纪人集合
$R, R^{(i)}$	所有请求的集合和时间间隔 i内的请求集合
$u_{r,b}$	将经纪人 b 分配给请求 r的效用
$\mathcal{M}^{(i)}, \mathcal{M}_{r,b}^{(i)}$	区间 i 内的分配以及 (r, b) 的分配指示变量
$UCB_{\mathbf{x},c}$	上下文 $\mathbf{x}$ 下容量 c 的上置信界
$\mathcal{B}_{\theta,\mathbf{D}}$	具有学习参数 θ，$\mathbf{D}$ 的上下文老虎机
$\mathcal{S}_\theta, g_\theta$	奖励映射函数及其梯度
$\mathcal{L}(\theta)$	上下文老虎机中神经网络的损失函数
$\mathcal{B}_b$	经纪人 b 的独占式上下文老虎机
$\mathcal{V}(cr)$	剩余容量 cr下的容量感知价值函数
β, γ, δ	学习率、折扣因子和更新阈值

[5]https://www.zhongan.com/ [6]http://www.lvshiguan.com

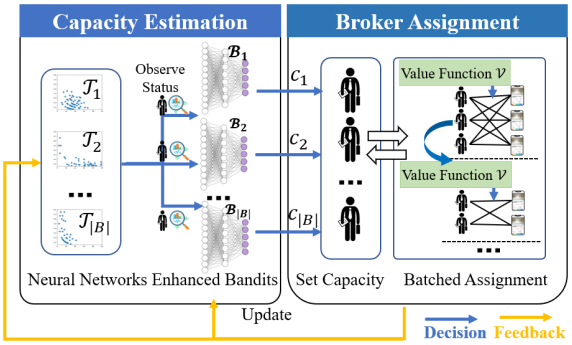


图 5：LACB 的工作流程。IV. 我们解决方案的概述

为解决CAA问题，我们提出了基于上下文老虎机的学习型分配（LACB），该方法通过上下文老虎机学习未知的经纪人容量，并从全局视角分配经纪人，以最大化总效用，同时避免顶级经纪人超负荷。我们首先概述LACB，并介绍各个功能模块。

LACB 由两个功能模块组成：容量估计和基于容量的分配。

- 容量估计模块通过神经网络增强型上下文老虎机，根据经纪人当前状态确定每日工作负载容量。

- 基于容量的分配模块通过容量感知价值函数，选择满足容量约束的一组经纪人，并将他们分配给请求。

图 5 展示了 LACB 的工作流程。该流程分为两个阶段：估计和分配。首先，我们观察经纪人的工作状态，并通过神经网络增强的多臂老虎机为其设定每日工作负载容量。在分配阶段，我们采用经纪人估计的容量，并使用容量价值函数来指导分配，从而捕捉具有不同工作负载的经纪人的长期效用。最后，我们将批量分配的结果存储为反馈，以改进未来的决策。

V. 容量估计

本节介绍我们的容量估计方法。我们将工作负载容量估算器建模为上下文老虎机，并提出一种神经网络增强的策略，用于决定每位经纪人的每日工作负载容量。

A. 基本思路

我们的方法源于估计经纪人容量时面临的以下三个挑战。

- 在线训练。在模型部署之前，收集经纪人在所有可能工作负载下的注册率数据并不现实。工作负载容量估算器需要在部署后，根据观察到的工作负载和注册率定期进行自适应调整。我们的解决方案是应用上下文老虎机算法 [14],这一学习方法，探索未知环境，并根据各种上下文信息做出决策。

表 II：经纪人的属性。

属性类型	属性	描述
基本信息。	Age	经纪人的年龄。
	工作年限	作为经纪人的工作年限。
	教育背景	教育背景（例如，本科生、硕士）。
	职称	职位（例如，助理、职员、经理）。
工作档案	响应率	经纪人在一分钟内响应请求的比率。
	对话轮次	最近 7/14/30/90 天通过应用进行的平均对话轮次。
	房屋带看次数	最近 7/14/30/90 天经纪人在线下带看的房屋数量。
	VR带看次数	经纪人在最近 7/14/30/90 天内通过 VR 带看房屋的次数。
	VR带看时间	经纪人在最近 7/14/30/90 天内通过 VR 带看房屋的时间。
	电话咨询次数	经纪人在最近 7/14/30/90 天内通过电话为客户提供咨询的次数。
	电话咨询时长	经纪人在最近 7/14/30/90 天内通过电话接听客户咨询的时长。
	应用咨询次数	经纪人在最近 7/14/30/90 天内通过应用接听客户咨询的次数。
	应用咨询时长	经纪人在最近 7/14/30/90 天内通过应用回复客户的时间。
	维护房屋数量	经纪人当前维护的房屋数量。
	服务客户数量	经纪人在最近 7/14/40/90 天内服务的客户数量。
	房屋交易数量	经纪人在最近 7/14/40/90 天内促成的房屋交易数量。
偏好	区域信息	经纪人偏好的社区以及兴趣点周边区域。
	住房信息	经纪人偏好的房屋价格、面积和类型。

- 经纪人的绩效与工作负载之间的复杂关系。经纪人的绩效与当前工作状态密切相关。例如，经纪人在销售旺季可能会更加疲惫，因此对繁重工作负载的承受能力较弱。正如第II-A节中的观察结果所示，经纪人的绩效与其工作负载之间呈非线性关系，我们通过神经网络对这种非线性复杂关系进行建模。

- 个性化工作负荷容量。如图3所示，注册率与工作负载之间的关系不仅复杂，而且因经纪人而异。然而，由于经纪人特定数据稀疏，直接学习个性化容量估计器颇具挑战。我们首先学习一个通用容量估计模型，再针对各个经纪人对其进行微调。

接下来，我们将详细介绍后续各项设计。

B. 将工作负载容量估算器建模为上下文老虎机

如上所述，我们利用上下文老虎机，通过与房地产平台交互，以在线方式学习通用的经纪人容量估算器。强化学习[15]（例如，Q-learning）主要对决策对状态的影响进行建模。然而，在我们的场景中，经纪人自身的工作状态不会受到我们的决策影响，因此，Q-learning 等方法不适用于容量估计。

上下文老虎机回顾。我们先简要回顾一下上下文老虎机。在未知环境中，多臂老虎机广泛用于跨越 n个批次进行在线决策，其中每个臂代表一个决策。在每个批次中，老虎机选择一个臂（决策），并从环境中获得奖励。随后，它会根据奖励更新决策策略，并尝试最大化 n 个批次中的总奖励。上下文老虎机还允许老虎机在每个批次开始时，利用额外信息（即上下文）做出决策。

我们的形式化。现在，我们说明如何将工作负荷容量估算器形式化为上下文老虎机。我们将经纪人的候选工作负荷容量视为老虎机的臂（表示为 $\mathcal{C}$）。经纪人的工作状态 $\mathbf{x}_b$ 被视为上下文，老虎机据此选择一个容量 $c_b \in \mathcal{C}$。工作负载为 w_b 时的每日签约率 s_b 用作奖励。工作负荷容量估算器与房地产平台交互，该平台被视为未知环境。在每个批次中，房地产平台执行分配算法并揭示奖励 s_b。我们使用 $(\mathbf{x}_b,\ w_b,\ s_b)$ 作为试验三元组，以更新老虎机（工作负荷容量估算器）的奖励函数，因为经纪人的工作负载 w_b 通常低于其容量 c_b。

C. 使用神经网络增强型*UCB*选择容量

将工作负载容量估算器构建为上下文多臂老虎机后，下一个问题是确定策略，以便针对给定的经纪人工作状态选择能够最大化每日注册率的工作负载容量。

标准UCB。上下文多臂老虎机的一种常见决策策略是上置信界（UCB）算法[14]。该算法假设环境处于合理可能的最佳状态，并使用上下文向量计算期望奖励的上置信界。随后，它选择具有最大上置信界的臂作为决策。对于我们的工作负载容量估计，可以使用经纪人的工作状态$\mathbf{x}$ 作为上下文，并使用下方的公式 [14]计算每个工作负载容量 c 的上置信界$UCB_{\mathbf{x},\ c}$。

$$UCB_{\mathbf{x},c} = f_\theta(\mathbf{x}, c) + \alpha\sqrt{f'_\theta(\mathbf{x}, c)^T D^{-1} f'_\theta(\mathbf{x}, c)} \tag{3}$$

其中，$f_\theta(\mathbf{x}\ c)$是一个线性模型，用于将上下文$\mathbf{x}$映射为某个工作负载容量的期望奖励；$f'_\theta(\mathbf{x}\ c)$是其导数。α 是预设的系数参数。θ 是线性模型的参数，D 是协方差矩阵。θ 和 D 是需要训练的参数。

神经网络增强型UCB。标准UCB算法的一个局限在于，假设期望奖励与上下文之间存在线性关系，即公式（3）中的 $f_\theta(\mathbf{x}\ c)$。因此，在我们的场景中（见第II-A节），标准UCB无法描述经纪人的注册率（期望奖励）与工作状态（上下文）之间的非线性关系。为此，我们用神经网络替代线性模型。

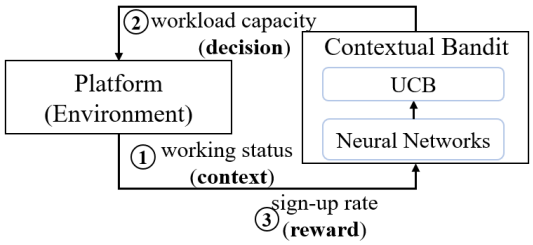


图6：神经网络增强型UCB的工作流程。

我们将相应的容量选择策略命名为神经网络增强型UCB。

图6展示了我们的神经网络增强型UCB的工作流程。我们用学习得到的奖励映射函数 $\mathcal{S}_\theta(\mathbf{x}\ c)$ 替换标准UCB中的线性模型 $f_\theta(\mathbf{x}\ c)$。为简单起见，我们采用具有 L 层的全连接多层感知器网络：

$$\mathcal{S}_\theta(\mathbf{x}, c) = W_L \cdot \sigma_{L-1}(\cdots \sigma_1(W_1[\mathbf{x}; c])) \tag{4}$$

其中，W_i $(1 \le i \le L)$ 是每层的学习参数，$W_1 \in \mathbb{R}^{m\times d}$、$W_i \in \mathbb{R}^{m\times m}$ $(2 \le i \le L-1)$、$W_L \in \mathbb{R}^{m\times 1}$。$\sigma_i$ 是ReLU激活函数。

设 $\theta \in \mathbb{R}^d$ 为神经网络的所有可学习参数，其梯度记为 $g_\theta(\mathbf{x}\ c) = \nabla_\theta \mathcal{S}_\theta \in \mathbb{R}^d$。那么，我们可以轻松地将式（3）中期望奖励的上置信界扩展如下。

$$UCB_{\mathbf{x},c} = \mathcal{S}_\theta(\mathbf{x}, c) + \alpha\sqrt{g_\theta(\mathbf{x}, c)^T \mathbf{D}^{-1} g_\theta(\mathbf{x}, c)} \tag{5}$$

也就是说，我们用神经网络 $\mathcal{S}_\theta(\mathbf{x}\ c)$ 及其梯度 $g_\theta(\mathbf{x}\ c)$ 替换线性模型 $f_\theta(\mathbf{x}\ c)$ 及其导数 $f'_\theta(\mathbf{x}\ c)$，其中 θ 和 $\mathbf{D}$ 按照标准 UCB 的方式进行训练。

多臂老虎机训练。算法 1 展示了如何随时间训练上下文老虎机 $\mathcal{B}_{\theta,\ \mathbf{D}}$，即，学习我们神经网络增强型 UCB 中式（5）定义的参数 θ 和 $\mathbf{D}$。我们首先观察工作状态 $\mathbf{x}_b$，然后根据式（5），优先探索期望奖励上置信界最大的容量（第 6 至 10 行）。接下来，我们使用与式（5）相同的扩展方式更新协方差矩阵 $\mathbf{D}$，即，将线性模型的导数 f'_θ（$\mathbf{x}\ c$）替换为神经网络的梯度 g_θ（$\mathbf{x}\ c$）（第 12 行）。之后，我们观察实际工作负载 w_b 以及奖励 s_b（第 13 行）。我们按照批大小 $batchSize$（预设为 16）进行探索和观察，并将三元组（$\mathbf{x}\ w$, s）存储在缓冲区ob中。收集 $batchSize$ 个批次的观察结果后，我们通过最小化以下损失来更新参数 θ。

$$\mathcal{L}(\theta) = \sum_{o\in ob} \|\mathcal{S}_\theta(\mathbf{x}_o, w_o) - s_o\|^2 + \lambda\|\theta\|_2^2 \tag{6}$$

其中 $(x_o,\ w_o,\ s_o)$ 是一次观测，λ 是用于避免过拟合的正则化参数。

使用老虎机估计容量。训练好多臂老虎机$\mathcal{B}_{\theta后,\ \mathbf{D}}$即可像算法1那样，用它估计后续批次的工作负载容量。也就是说，给定工作状态$\mathbf{x}$，我们选择根据公式（5）计算出的上置信界最大的工作负载容量 c。为便于表述，我们用 $\mathcal{B}.estimate(\mathbf{x})$表示工作负载容量

算法1： 神经网络增强型UCB

输入： 正则化参数 λ，上置信度
上置信界系数 α，以及批大小 $batchSize$

输出： 上下文老虎机$\mathcal{B}_{\theta,\ \mathbf{D}}$

1 初始化偏差矩阵 $\mathbf{D} \leftarrow \lambda\mathbf{I}$;
2 初始化观测缓冲区 $ob \leftarrow []$;
3 使用高斯分布初始化参数 θ ;
4 **对**每次试验 $t \in \mathcal{T}$**执行**
5 //探索工作负载容量;
6 观察经纪人的工作状态$\mathbf{x}$;
7 **对于**每个候选容量 $c \in \mathcal{C}$ **执行**
8 $U_{\mathbf{x},\ c} \leftarrow \mathcal{S}_\theta(\mathbf{x}\ c) + \alpha\sqrt{g_\theta(\mathbf{x}\ c)^T\mathbf{D}^{-1}g_\theta(\mathbf{x}\ c)}$;
9 选择 b 的容量 $c^* \leftarrow \arg\max_c U_{\mathbf{x},c}$;
10 **end**
11 //更新上置信度和奖励函数;
12 $\mathbf{D} \leftarrow \mathbf{D} + g_\theta(\mathbf{x}\ c) \cdot g_\theta(\mathbf{x}\ c)^T$;
13 观察经纪人的工作负载 w 和奖励 s;
14 将观测到的三元组（$\mathbf{x}\ w, s$）存储在缓冲区 ob 中;
15 **如果** $ob.size = batchSize$ **则**
16 将损失函数 $\mathcal{L}(\theta)$定义为 $\sum_{o\in ob}\|\mathcal{S}_\theta(\mathbf{x}_o,\ c_o) - s_o\|^2 + \lambda\|\theta\|_2^2$;
17 $\theta \leftarrow \theta - \nabla_\theta\mathcal{L}(\theta)$;
18 清空观测缓冲区 $ob \leftarrow []$;
19 **end**
20 **结束**
21 **返回**上下文老虎机 $\mathcal{B}_{\theta,\ \mathbf{D}}$;

估计，即给定工作状态$\mathbf{x}$后选择合适的工作负载容量。

D. 个性化工作负载容量估计器

如前所述，上下文老虎机仅为所有经纪人学习一个通用容量估计器，但工作负载容量估计可能因经纪人而异。我们通过在经纪人特定数据上微调式（5）中的神经网络 $\mathcal{S}_\theta(\mathbf{x}\ c)$，实现个性化工作负荷容量估计。

具体而言，我们首先在所有经纪人的观察结果 $\cup_{b\in B}\mathcal{T}_b$ 上训练基础奖励映射函数θ^{base}，即式（4）中定义的神经网络。然后，将 θ^{base} 的前$L-1$ 层复制到经纪人 b的代理专属奖励映射函数 θ_b 中。之后，冻结 θ_b的前 $L-1$层，并按照算法 1，根据该经纪人的观察结果 $\mathcal{T}_b$ 微调最后一个全连接层。通过这种方式，我们获得个性化奖励映射函数。

E. 神经网络增强型置信上界的有效性

现在，我们从理论上分析用于基于上下文老虎机的工作负载容量估计的神经网络增强型UCB策略的有效性。我们通过遗憾来评估其有效性；遗憾是老虎机的一项标准性能指标 [14]。

遗憾定义为最优策略与学习得到的决策策略之间的总奖励差异。对于我们的工作负载容量估计，遗憾可定义为理想容量下注册率总和与

估计容量下注册率总和之间的差异，形式化表示如下：

$$\mathcal{R}egret = \sum_{t=1}^{|\mathcal{T}|} \max_{c\in\mathcal{C}} s(\mathbf{x}_b, c) - \sum_{t=1}^{|\mathcal{T}|} s_t \tag{7}$$

其中， $\mathcal{T}$为试验三元组，$\max_{c\in\mathcal{C}} s(\mathbf{x}_b, c)$是在工作状态$\mathbf{x}_b$下的理想奖励，而 s_t 是由我们的神经网络增强型UCB算法产生的奖励。

关于我们的神经网络增强型UCB算法的遗憾，我们有如下结论。

定理1。对于采用神经网络增强型*UCB*算法、具有 L层多层感知机网络的上下文老虎机，若存在 $|\mathcal{C}|$个候选容量，则 n 个批次的遗憾不超过$\frac{n|\mathcal{C}|\xi^L}{\pi^{L-1}}$，其中 $\pi \approx 3.14$ 为圆周率常数，ξ 为多层感知机模型中单个参数值的最大值。

证明。令 $c_t^* = \arg\max S_{\theta^*}(\mathbf{x}_t, c)$表示上下文$\mathbf{x}_t$下的最优候选容量，且$\hat{c}_t = \arg\max U_{\mathbf{x}_t,c}$表示批次$t$中的最优选择。首先，我们分析批次 t中的瞬时遗憾 r_t，其定义如下：

$$r_t = S_{\theta^*}(\mathbf{x}_t, c^*) - S_{\theta^*}(\mathbf{x}_t, \hat{c}) \tag{8}$$

利用利普希茨连续 [16]来界定多臂老虎机的瞬时遗憾值，我们有，

$$\|S_{\theta^*}(\mathbf{x}_t, c^*) - S_{\theta^*}(\mathbf{x}_t, \hat{c})\|_2 \leq Lip(S_{\theta^*})\|c^* - \hat{c}\|_2 \tag{9}$$

其中， $\|\cdot\|_2$ 表示 L_2范数， $Lip(S_{\theta^*})$称为 S_{θ^*}的利普希茨常数。

接下来，我们分析奖励函数 S_{θ^*}的利普希茨常数$Lip(S_{\theta^*})$的上界。根据 [16],中的引理2，我们可以构造其上界，

$$Lip(S_{\theta^*}) \leq \frac{\|W_L \mathrm{diag}(\sigma_{L-1}) W_{L-1} \cdots \mathrm{diag}(\sigma_1) W_1\|_2}{\pi^{L-1}} \tag{10}$$

其中， $\pi \approx 3.14$ 为圆周率常数。若将*ReLU*作为激活函数 σ，则有 $\|\sigma\|_2 \leq \|\mathbf{I}\|_2$，其中I为单位矩阵。因此，利普希茨常数 $Lip(S_{\theta^*})$的上界可改写为，

$$Lip(S_{\theta^*}) \leq \frac{\Pi_{i=1}^{L}\|W_i\|_{op} \cdot \Pi_{i=1}^{L-1}\|\mathbf{I}\|_2}{\pi^{L-1}} \leq \frac{\xi^L}{\pi^{L-1}} \tag{11}$$

其中， $\|W_i\|_{op}$ 是 W_i的算子范数，即， W_i ($\|W_i\|_{op} \leq \xi$) 的最大单值。由于 c^* 和$\hat{c}$ 均属于 $\mathcal{C}$，因此有 $\|c^* - \hat{c}\|_2 \leq |\mathcal{C}|$。于是，对于瞬时遗憾值$r_t$，我们可以给出其上界，

$$r_t = S_{\theta^*}(\mathbf{x}_t, c^*) - S_{\theta^*}(\mathbf{x}_t, \hat{c}) \leq Lip(S_{\theta^*})\|c^* - \hat{c}\|_2 \leq \frac{|\mathcal{C}|\xi^L}{\pi^{L-1}} \tag{12}$$

最后，我们证明了我们的神经网络增强型UCB策略在 n 批次上的遗憾值上界为：

$$\mathcal{R}egret = \sum_{t=1}^{n} r_t \leq \frac{n|\mathcal{C}|\xi^L}{\pi^{L-1}} \tag{13}$$

□

讨论。我们从定理1中得出两点实践启示。

- 设置合适数量的候选容量有助于选出最优决策。我们通过实证根据第 II 节的历史观测确定候选集的范围，并且不探索注册率明显偏低的工作负载容量。
- 虽然更深的网络可以对经纪人的绩效与工作状态之间更复杂的关系进行建模，但也可能妨碍多臂老虎机选择最优的工作负载容量。我们在神经网络增强型UCB算法中实证采用3层多层感知机网络，以平衡神经网络的复杂度和工作负载估计的有效性。

VI. 基于容量的分配

现在，我们介绍LACB的分配模块。该模块以估计的容量为输入，并同时考虑容量约束以及跨批次分配的依赖关系来进行分配。

A. 将批量分配建模为马尔可夫决策过程

不同于以往研究 [10], [17] 对每个批次都独立地进行分配，我们通过将一段时间内的分配建模为马尔可夫决策过程（*MDP*） [14]，提出以更全面的视角匹配经纪人。该建模考虑了不同批次间分配的依赖关系（即，经纪人随时间变化的剩余容量），并有可能带来更高的总效用。

标准的马尔可夫决策过程模型由四个要素构成：状态、动作、状态转移和奖励。下面我们将在经纪人分配的背景下说明这些要素。

- **状态**。由于我们的分配决策考虑容量，我们将每个经纪人的状态定义为 $cr_b \in [0, c_b]$，其中cr_b 和 c_b 分别表示经纪人的剩余容量和工作负载容量。
- **动作**。在批次 i中，动作是一个分配策略$\mathcal{M}_{(i)\,r,b}$，适用于每个请求 r 和每个经纪人 b。如果 $\mathcal{M}_{(i)\,r,b} = 1$，则经纪人 b 分配给请求 r；如果 $\mathcal{M}_{(i)\,r,b} = 0$，则不分配。在本文中，我们采用价值函数引导分配策略（见第VI-B节）。
- **状态转移**。经纪人的状态会因动作 $\mathcal{M}_{(i)\,r,b}$ 而改变。如果 $\sum \mathcal{M}_{(i)\,r,b} = 0$，经纪人$b$的状态保持不变。否则，状态 cr_b 将转移至 $cr_b - \sum \mathcal{M}_{(i)\,r,b}$。也就是说，经纪人的剩余容量会根据分配给该经纪人的请求数量而减少。
- **奖励**。动作的奖励定义为批次 i 中所有经纪人的效用（又称批次效用），即，$r(\mathcal{M}_{(i)}) = \sum_{r,b} u_{r,b} M_{(i)\,r,b}$。请注意，马尔可夫决策过程模型的奖励不同于多臂老虎机的奖励（见第 V 节）；后者是单个经纪人的累积效用，即，$s_b = \sum_i \sum_r u_{r,b}\mathcal{M}_{(i)\,r,b}$。

请注意，我们将批量分配建模为马尔可夫决策过程模型（并采用基于强化学习的解决方案），而非采用老虎机算法，因为后者不适用于具有状态转移的长期规划 [14]。

B. 容量感知分配

基于上述马尔可夫决策过程模型，我们利用容量感知价值函数来指导分配。

容量感知价值函数。 在马尔可夫决策过程中，通过学习状态[10]，[13]的价值函数来做出决策是常见的做法。在本文中，我们定义了一个容量感知价值函数 $\mathcal{V}(i,\ cr)$，它表示经纪人在批次 i之后的期望效用，其中 cr 是经纪人的剩余容量。这种容量感知价值函数能够反映剩余容量不同的经纪人的长期效用。随后，我们使用 Q-learning [10]，这一时间开销相对较低的经典方法，根据以下时序差分（TD）方程训练容量感知价值函数。

$$\mathcal{V}(cr) \leftarrow \mathcal{V}(cr) + \beta[u + \gamma\mathcal{V}(cr') - \mathcal{V}(cr)] \tag{14}$$

其中， cr/cr' 是采取动作后当前/转移到的状态，u 是动作的奖励， β 和 γ 分别是 TD 的学习率和折扣因子。

算法 2： 价值函数引导分配

输入： 经纪人 $B = \{b_1, b_2, ..., b_{|B|}\}$、请求 $R = \{R^{(1)}, R^{(2)}, ..., R^{(|I|)}\}$ 在 I个时间间隔内，经纪人的专属带宽 $\{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_{|B|}\}$

输出： 匹配结果 $\mathcal{M}$

1 **对于** 每个经纪人 b在 B **执行**

2 $c_b \leftarrow \mathcal{B}_b.estimate(\mathrm{x}_b)$;

3 **对于** 每个区间 $i \in I$ **执行**

4 获取可用经纪人 $B_+ = \{b | w_b < c_b \wedge b \in B\}$；

5 **对于** 每个候选对 $(r, b) \in R^{(i)} \times B_+$ **执行**

6 $$u'_{r,b} \leftarrow \begin{cases} u_{r,b} + 0, & \text{如果 } f_b \le \delta \\ u_{r,\ b} + \gamma\mathcal{V}(cr') - \mathcal{V}(cr), & \text{如果} f_b > \delta \end{cases}$$

7 基于以下内容执行匈牙利算法精化效用 $\mathcal{M}^{(i)} = KM(u', R^{(i)}, B_+)$;

8 **对于**每个经纪人 b 在 B_+ **执行**

9 $w_b \leftarrow w_b + \sum_r \mathcal{M}^{(i)}_{r,b}$;

10 基于容量感知价值函数更新时序差分方程 Eq. (14)；

11 **对于**每个经纪人 b在 B **执行**

12 $s_b \leftarrow \sum_i \sum_r u_{r,\ b}\mathcal{M}^{(i)}_{r,b}$;

13 $\mathcal{B}_b.update(\mathrm{x}_b,\ w_b,\ s_b)$;

14 **返回** $\mathcal{M} = \cup_{i \in I}\mathcal{M}^{(i)}$

价值函数引导分配（VFGA）。 现在，我们可以利用上述容量感知价值函数，从全局视角分配经纪人，以最大化总效用。算法 2 展示了整体分配算法。首先，我们根据上下文老虎机$\mathcal{B}_b$ 确定经纪人的个性化容量（见第 V 节）。在第 4–14 行中，我们在每个批次内进行分配。具体而言，在第 5 行中，我们首先选出一组可用经纪人 B_+，其工作负载 w_b 低于其容量c_b。然后，我们按如下方式更新每个候选匹配对的效用。

$$u'_{r,b} = \begin{cases} u_{r,b} + 0, & \text{if } f_b \le \delta \\ u_{r,b} + \gamma\mathcal{V}(cr') - \mathcal{V}(cr), & \text{if } f_b > \delta \end{cases} \tag{15}$$

其中，

$u_{r\ \ b}$ $cr = ,\ b/u'_{r,}$

$c_b - w_b$ $cr' = cr - 1$

分别表示原始效用和精化效用，且 。请注意，只有顶级经纪人可能达到其工作负载容量；我们仅对达到容量的频率 f_b 超过 δ的顶级经纪人使用价值函数，其中 δ 是接近 1 的正数。在第 9 行中，我们在一个二分图上运行经典的 Kuhn–Munkres（KM） [18] 算法，该二分图的边权为精化效用$u'_{r,\ b}$，并返回分配策略 $\mathcal{M}^{(i)}$。在第 10–12 行中，我们根据公式（14）更新已分配经纪人的工作负载和容量感知价值函数。在第 15–17 行中，当我们完成所有批次中的请求分配后，收集经纪人的工作负载和绩效作为反馈，以更新每位经纪人的老虎机。如前所述，每当观测缓冲区已满时，我们就更新老虎机的参数；否则，仅将新观察结果添加到缓冲区中。最后，VFGA 返回分配结果 $\mathcal{M}$。

讨论。 我们对分配模块作两点说明。

- 由于请求数量 $|R|$ 通常少于经纪人数量 $|B|$，一种常见做法是向较小的一侧添加一些虚拟顶点，以构造平衡二分图 [19]–[21]。通过添加 $|B| - |R|$ 个虚拟顶点，我们得到一个两侧各有 $|B|$ 个顶点的平衡二分图，并可以执行经典的KM算法。
- 一旦客户对分配的经纪人感到不满意，就可以向平台申诉，要求更换经纪人。平台会将该客户与已分配经纪人之间的效用设为 0，恢复该经纪人的工作负载，并在下一个时间区间选择另一名经纪人。

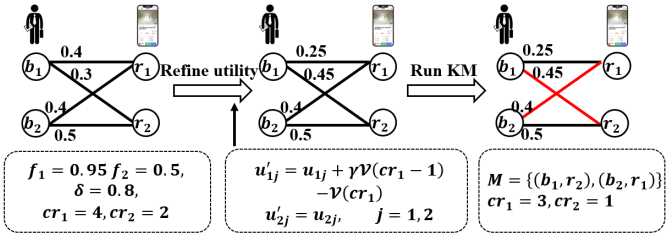


图 7：VFGA算法示例。

图 7展示了VFGA算法核心步骤的一个示例。现有两名经纪人 b_1、b_2 和两个请求r_1、r_2。(b_1, r_1)、(b_1, r_2)、(b_2, r_1)和(b_2, r_2)的效用分别为0.4、0.3、0.4和0.5。根据公式（15），b_1 达到阈值 $\delta = 0.8$，我们将 b_1的效用优化为 $u_{11} = 0.25$和 $u_{12} = 0.45$，而 b_2 的效用保持不变。然后，我们运行KM算法，在优化后的权重上获得最大分配，即 $\{(b_1, r_2)$、$(b_2, r_1)\}$。最后， b_1 和 b_2 的剩余容量分别更新为3和1。

复杂度分析。 对于每个批次，VFGA算法的时间复杂度由KM算法 [18]决定。由于KM算法是在两侧各有 $|B|$ 个顶点的平衡二分图上运行的，因此VFGA算法的时间复杂度为 $\mathcal{O}(|B|^3)$。

C. 通过经纪人选择加速分配

房地产平台需要尽可能高效地为客户分配经纪人，这是影响

算法 3： 候选经纪人选择（CBS）

输入： 候选规模 k、请求 r 和经纪人 B

输出： 候选经纪人 Top_k^r

```
如果 |B| ≤ k 则
    返回 B;
结束
选择一个随机值 p，范围为 u_{r,b} (b ∈ B);
LC ← {b ∈ B|u_{r,b} ≥ p};
RC ← {b ∈ B|u_{r,b} < p};
如果 |LC| ≥ k，则
    Top_k^r ← CBS(k, r, LC)
结束
否则
    Top_k^r ← LC ∪ CBS(k − |LC|, r, RC);
结束
返回Top_k^r
```

用户体验的一个重要因素。为了确保良好的用户体验，通常需要在 2 秒内响应 [22]。由于在线房地产平台上的经纪人数量与请求数量极不平衡，因此有机会加快 VFGA算法的运行速度。在每个批次中，平台通常需要将数千名经纪人分配给仅有几十个请求。如果我们剪除不太可能匹配成功的经纪人，就能显著减少请求一侧添加的虚拟顶点数量，从而降低时间复杂度。

理论证据。我们声称，只需一小组经纪人即可实现有效分配，定理2和推论 1对此进行了说明。

定理2。 给定一个二分图 $G =< U, V, E >(|U| \le |V|)$，并令 $opt_{\mathcal{M}}(u)$ 为最优分配 $\mathcal{M}$中与 u 匹配的顶点。存在一个最优分配$\mathcal{M}$，使得对于任意顶点 $u \in U$，都有 $opt_{\mathcal{M}}(u) \in \boldsymbol{Top}^u_{|U|}$，其中 $\boldsymbol{Top}^u_{|U|}$ 是与 u相连的所有顶点中边权重最大的 $|U|$ 个顶点的集合。

证明。 采用反证法。我们先假设结论不成立。令 u 为 U 的任意顶点，且与 u 匹配的顶点为 v^*。假设不存在最优分配 $\mathcal{M}$，使得 $v^* = opt_{\mathcal{M}}(u) \in \text{Top}^u_{|U|}$。考虑一种极端情况：在最优分配 $\mathcal{M}$中， U 的所有其他顶点都与 $\text{Top}^u_{|U|}$ 中的顶点匹配。此时仍至少存在一个未匹配的顶点 $v' \in \text{Top}^u_{|U|}$。根据 $\text{Top}^u_{|U|}$的定义，有 $w(u, v') \ge w(u, v^*)$。若将 (u, v') 替换为 (u, v^*)，并保持 $\mathcal{M}$ 的其他匹配不变，我们就能构造出另一个最优分配 $\mathcal{M}'$，这与“不存在最优分配”这一假设矛盾。 □

推论 1. 给定一个不平衡二分图 $G =< U, V, E >$，为了找到最优分配，我们至多需要为每个顶点 $u \in U$ 选取 $|U|$ 个候选顶点，即，对任意 $u \in U$，将 $\boldsymbol{Top}^u_{|U|}$ 作为候选集。

候选经纪人选择算法。 为了高效地为每个请求 r选择候选集合 $\text{Top}^r_{|R|}$，我们设计了候选经纪人选择算法（Candidate Broker Selection，CBS）算法（见算法3），其灵感来自经典选择问题 [18]的解决方案。我们主要介绍如何将这一优化集成到我们的 VFGA 中。在每个批次中，获取可用经纪人 B_+ 后（算法 2 第 5 行），我们执行 CBS 算法，选择所需的经纪人 $\cup_{r \in R^{(i)}} \text{Top}^r_{|R^{(i)}|}$。随后，我们便能在规模小得多的二分图上进行分配。

复杂度分析。 候选经纪人选择的期望时间复杂度为 $\mathcal{O}(|R||B|)$。基于上述优化，我们可以在包含 $|R|$ 个顶点和 $|R|^2$ 条边的二分图上分配经纪人，因此，执行 KM算法的时间复杂度从 $\mathcal{O}(|B|^3)$ 降至$\mathcal{O}(|R|^3)$，其中 $|R|$ 通常远小于 $|B|$。因此，采用 CBS 的 VFGA算法的总体时间复杂度为 $\mathcal{O}(|R|^3 + |R||B|)$。

表 III：合成数据集。

因素	设置
经纪人数量 $\|B\|$	500、1000、**2000**、5000、10000
请求数量 $\|R\|$	10K、20K、**50K**、100K、200K
覆盖天数 Day	7, 10, **14**, 17, 21
不平衡程度 σ	0.005, 0.01,**0.015**, 0.02, 0.05

表 IV：真实世界数据集。

City	日期	经纪人	请求
A市	8月 1 ~2021年8月21日	5515	103106
B市	7月 1 ~2021年7月21日	8155	387339
C市	6月 8 ~2021年6月28日	3689	74831

VII. 实验研究

本节介绍 LACB 的评估结果 .

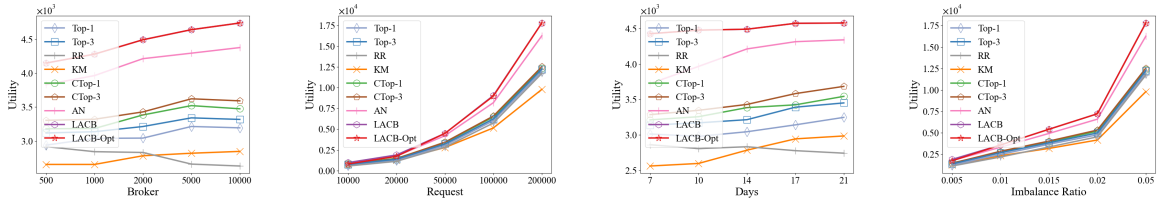
A. 实验设置

数据集。我们在合成数据集和真实数据集上测试容量感知分配算法。

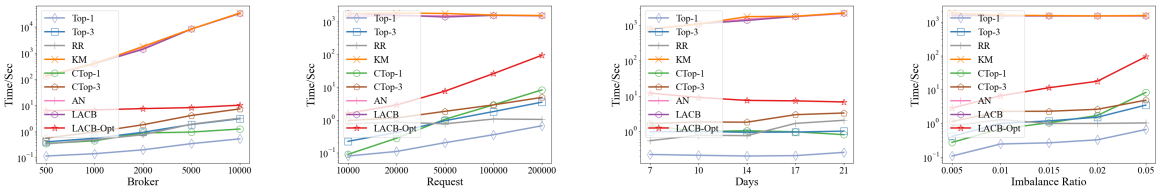
- 合成数据集。我们总共生成了 2000 名经纪人和 50000 个请求。随后，我们改变经纪人数量、请求数量、覆盖天数以及不平衡程度。不平衡程度 $\sigma = |R|/|B|$ 是每个批次中请求数与经纪人数的比值。为改变 σ，我们保持经纪人数不变，并调整请求数量。表 III 总结了合成数据集的配置。默认设置以粗体标出。

- 真实世界数据集。我们从中国最大的在线房地产平台贝壳收集了覆盖 21 天、来自三个城市的数据。表 IV 总结了真实数据集的统计信息。

实现。 我们的实验在贝壳的模拟器上进行，该模拟器采用相同的效用函数，并输出请求与经纪人之间的效用，因此我们可以将此类效用作为输入，并在真实世界的匹配实例上评估算法。在神经网络增强型UCB中，我们采用了三层多层感知机网络，并将输入层、隐藏层和输出层的大小分别设为 128、64 和 16。我们采用*ReLU* 作为激活函数。在算法 1 中，我们将 α 设为 0.001，将批大小 $batchSize$ 设为 16，并将正则化参数 λ 设为 0.001。在算法 2 中，我们设定



(a) 变化 $|B|$时的效用 (b) 变化 $|R|$时的效用 (c) 变化 Day时的效用 (d) 变化 σ时的效用



(e) 变化 $|B|$时的时间 (f) 变化 $|R|$时的时间 (g) 变化时的时间 Day (h) 变化时的时间 σ

图 8：合成数据集上的结果。

学习率 β 为 0.25，折扣因子 γ 为 0.9，阈值 δ 为 0.8。所有实验均在配备 32GB 主存的 Intel(R) Xeon(R) Gold 6230R 中央处理器 @ 2.10GHz 上进行。算法使用 Python 3 实现。

对比算法。 我们将 LACB即，第 VI-B 节，以及采用 CBS 的 LACB即，第 VI-C 节（记为 LACB-Opt）与两类基线方法进行比较。第一类（Top-K、RR 和 KM）不为经纪人设置明确的容量，而第二类（CTop-K 和 AN）先选择经纪人的容量，再将请求分配给他们。

- Top-K 推荐（Top-K） [2]：按效用从高到低排序并返回 K 个经纪人。我们评估 Top-1 和 Top-3 推荐。
- 随机推荐（RR）：将经纪人的服务质量作为采样权重，并随机推荐一位经纪人。RR 扩展了先前的 公平匹配 算法 [23] ，并将服务质量视为公平性指标。通过将在线请求分配给所有经纪人，它可以避免过载现象。
- Kuhn–Munkre（KM）算法 [18]：该算法在每个批次中运行 KM 算法，将经纪人分配给请求。
- 约束 Top-K（CTop-K） [24]：这是 Top-K 的扩展，CTop-K 会观察城市层面的工作负载与注册率之间的关系（图2），并通过经验选择所有经纪人的容量。根据经验，A市、B市和C市经纪人的工作负载容量分别设为 45、55 和 40。我们测试了 K=1 和 K=3 两种情况下的 CTop-K。
- 基于 NeuralUCB 的分配（AN）：该方法利用 NeuralUCB [9]探索工作负载容量，并通过 KM 算法 [18]将经纪人分配给请求。

B. 合成数据集上的绩效

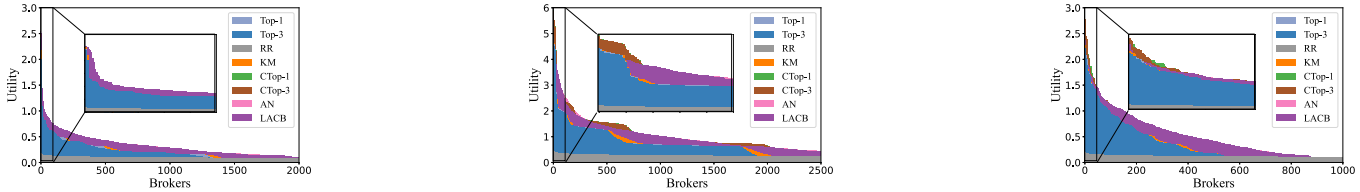
在这组实验中，我们测试了不同参数对不同算法绩效的影响。

经纪人数量的影响。图 8 的第一列展示了改变 $|B|$ 的结果。就总效用而言，我们的 LACB 和 LACB-Opt 优于其他所有基线方法，包括 Top-K、CTop-K、KM、RR 和 AN。我们还观察到，随着 $|B|$ 增加，Top-K 推荐的效用会下降，这表明由于过载现象，增加经纪人数量并不会提高总效用。最后，LACB-Opt 达到的效用与 LACB 相同，这与推论 1 中的理论分析一致，即，候选经纪人选择不会牺牲总效用。就运行时间而言，随着 $|B|$ 增加，KM、AN 和 LACB 因其立方时间复杂度而变得低效，而 Top-k、RR 和 CTop-k 的运行时间仅略有增加。LABT-Opt 的运行时间保持稳定，因为其时间复杂度主要取决于请求数量，并且比其他基于 KM 的算法（KM、AN、LACB）更快。

请求数量的影响。图8第二列展示了改变 $|R|$ 所得的结果。总效用通常会随着 $|R|$ 增加而提高。与前一项实验类似，LACB 和 LACB-Opt 达到相同的效用，且表现优于其他算法。至于运行时间，基于 KM 的算法比其他算法慢。LACB-Opt 比 KM、AL 和 LACB 最多快 $16.4\times \sim 1091.9\times$ 。与 Top-K 和 CTop-K 相比，LABT-Opt 的速度也颇具竞争力，尤其是在请求数量较少时。

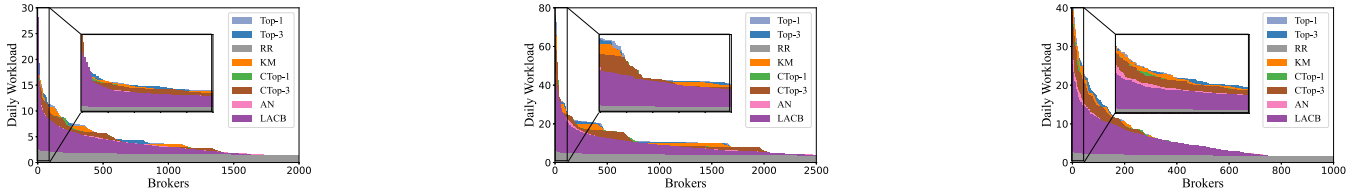
覆盖天数的影响。图 8 的第三列展示了变化覆盖天数 Day 的实验结果。就总效用而言，我们的方法仍优于其他基线方法。特别是，AN 在覆盖七天时获得的效用较低，这表明它可能面临冷启动问题；而 LACB 在覆盖天数变化时始终表现出色。同样，LACB 和 LACB-Opt 的效用表现相同。至于运行时间，变化 $|B|$ 时也呈现出类似趋势。我们的 LACB-Opt 比其他基于 KM 的算法快$65.5\times \sim 329.4\times$ 。

不平衡程度的影响。图8的第四列绘制了不同的不平衡比率 σ。由于我们固定 $|B|$并改变 $|R|$ 来测试不同的不平衡比率，随着 σ 增加，所有算法的效用变化趋势都相似。LACB-Opt 相对于 LACB 的加速比与不平衡比率 σ密切相关。例如，当 $\sigma = 0.005$ 和 $\sigma = 0.05$时，LACB-Opt 分别比 LACB 快 $641.7\times$ 和 $16.4\times$ 。



(a) A市 (b) B市 (c) C市

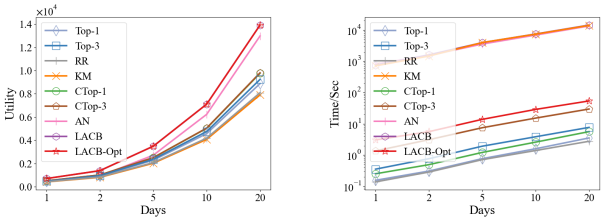
图 9：所有被比较算法的效用分布。



(a) A市 (b) B市 (c) C市

图10：所有对比算法的工作负载分布，其中我们可以将 Top-K 视为过载结果。

C. 真实数据集上的性能



(a) A市的效用 (b) A市的时间

(c) B市的效用 (d) B市的时间

(e) C市的效用 (f) C市的时间

图11：真实世界数据集上的结果。

总体绩效。图11展示了在三个真实世界数据集上的结果。以A市为例，我们首先可以从总效用方面得出以下观察结果。正如预期，Top-K 在所有三个数据集上的表现都很差。Top-3 略优于 Top-1，因为 Top-1 更容易使推荐的经纪人超载。CTop-K 的总效用高于 Top-K，表明容量感知的必要性。由于采用了上下文老虎机，AN 的表现优于大多数基线方法，而我们的 LACB 和 LACB-Opt 又优于 AN。运行时间随天数线性增长。与合成数据集上的结果类似，由于时间复杂度为立方级，KM、AN 和 LACB 的速度最慢。LACB-Opt 的效率具有竞争力，仅比 Top-K 和 CTop-K 慢 $1.7 \sim 24.2$ 秒，并比其他基于 KM 的算法快 $233.4\times \sim 284.9\times$。我们在来自三个城市的数据上也观察到了类似结果。

对经纪人的深入分析。为了了解我们的算法相较于基线方法的优势，我们进一步分析基线方法的效用和工作负载分布。如前所述，LACB-Opt 与 LACB 的区别仅在于效率，因此以下分析无需纳入 LACB-Opt。由于我们关注的是顶级经纪人的效用/工作负载，因此仅展示效用或工作负载较高的经纪人。其他经纪人的效用/工作负载也呈现与图 9 和图 10 类似的长尾分布，故此处省略。

- 效用分布。图 9 绘出了所有算法的效用分布。以 A市为例。基于容量的分配算法（CTop-K、AN 和 LACB）对大多数经纪人实现了比 Top-K 更高的效用，即，避免过载现象对个体层面也有好处。特别是，与 Top-K 相比，LACB 中有 80.8% 的经纪人效用有所提升。由于 RR 会将请求随机分配给所有经纪人，因此大多数经纪人的效用较为接近。然而，与 Top-K 相比，RR 降低了 25.7% 经纪人的效用。在 B市和 C市也观察到了类似结果。总而言之，LACB 能够提升顶级经纪人和其余经纪人的效用。

- 工作负载分布。图10绘制了所有算法的工作负载分布。正如预期，Top-K 使顶级经纪人承担了最高的工作负载。RR 随机分配请求，因此顶级经纪人的工作负载最低。然而，即使顶级经纪人拥有剩余容量，RR 也会阻止他们处理更多请求，从而限制了顶级经纪人的潜力。除 RR 外，LACB 中顶级经纪人的工作负载最低，这意味着 LACB 中的顶级经纪人面临较低的过载风险。

总而言之，得益于适合顶级经纪人的工作负载容量，LACB 的表现优于基线方法。

D. 实验结果总结

我们将主要实验发现总结如下。

- CTop-K 在所有数据集上的表现均优于 Top-K，表明设置工作负载容量是必要的。
- 我们的解决方案 LACB 和 LACB-Opt 的总效用通常高于其他基线方法。与 Top-K 相比，它们还提升了 72.0%∼82.2% 经纪人的效用。
- 在不损失效用的情况下，LACB-Opt 在真实世界数据集上的速度最高可达 284.9× 其他基于 KM 的算法的数倍，这与我们的理论分析一致。

VIII. 相关工作

我们的工作主要与三条研究脉络相关：在线任务分配、房地产数据科学和上下文老虎机。我们将综述具有代表性的研究如下。

在线任务分配。 任务分配是众包的核心操作 [25]–[33]。我们聚焦于在线任务分配，并将以往研究归纳为两类：基于顶点的模式和基于批次的模式。 （*i*）基于顶点的模式。在 [34],Kazemi等人 研究了空间众包中的二分图匹配，并采用基于贪心的分配算法来优化总效用。Tong等人 [35] 对在线二分图匹配开展了实验研究，结果表明，贪心算法在许多实际设置中具有竞争力。后续研究探索了具有不同目标或约束的任务分配问题，例如公平性 [23], [36], [37], 、隐私 [38]–[40]和激励 [41]–[43]。特别是， [23] 中的公平感知任务分配算法可扩展为基线算法，以避免我们场景中的过载现象。 （*ii*）基于批次的模式。这种模式在网约车领域十分普遍 [10], [11], [13]。Wang等人 [10] 采用强化学习，自适应地调整每个批次分配中的时间窗口。Zhang等人 [11] 通过启发式爬山法解决批次任务分配问题。 [13] 和 [44] 中的作者利用价值函数优化长时间范围内的序贯决策，这启发了我们方法的核心思想之一。现有解决方案主要关注分配策略，并假定容量已知；然而，我们的场景更具挑战性，因为在进行分配之前，我们需要估计个性化容量。

房地产数据科学。大数据的普及激发了人们对在房地产行业 [3]采用数据驱动方法的兴趣。以往的研究主要聚焦于两个主题：住房评估 [45] 和住房查找 [46]。在 [45],中，Fu 等人通过建模邻近房产及其关联商业区域之间的依赖关系，改进了房地产估价。Zhang 等人 [47] 提出了一种基于图神经网络的方法，用于处理异步时空依赖关系。Grbovic 等人 [46] 设计了嵌入技术，以实现实时个性化住房搜索。在 [48], 中，Weng 等人通过分析住宅位置与相关兴趣点之间的可达性，改进了找房服务。

在这项工作中，我们研究经纪人匹配这一在线房地产平台的重要问题，但这一问题鲜少从数据驱动的角度进行优化。

上下文老虎机。 上下文老虎机利用额外信息做出决策，已应用于推荐系统 [8], 查询优化 [49]和动态定价 [41]。早期研究主要关注遗憾值上界的理论分析 [50], [51]。Li 等人 [8] 于 2010 年提出 LinUCB 算法，将线性预测器与置信上界算法相结合；后续研究者扩展了 LinUCB 的基本框架，以实现高效学习 [52] 和潜在关系表示 [53]。近期，一些研究利用神经网络优化上下文老虎机 [9], [49], [54]。Zhou等人 [9] 设计了一种不对奖励函数作任何假设的神经上下文老虎机算法，可将其扩展为适用于我们场景的基线。Marcus 等人 [49]提出了结合树卷积神经网络的上下文老虎机，用于学习数据库中的查询执行策略。Ban等人 [54] 提出了多方面上下文老虎机，其中每个方面都用于刻画用户的一种需求。然而，这些算法并不直接适用于个性化估计，而这正是我们容量感知经纪人匹配场景所需的。我们的 LACB 建立在这一系列研究的基础上，并通过采用层迁移实现更具数据效率的老虎机学习，从而改进了以往的研究。

IX. 结论

本文研究了在线房地产平台中经纪人匹配的过载现象。具体而言，我们发现，主流平台采用的经纪人匹配 top-k 推荐机制往往会使顶级经纪人不堪重负，从而降低注册率和总效用。其根本原因在于忽视了经纪人的工作负载容量。为此，我们提出了 LACB，通过可迁移神经上下文老虎机有效估算个性化容量，并从全局视角为客户分配经纪人。我们还提出了 LACB-Opt，可显著提升 LACB 在不平衡二分图上的效率。基于真实在线房地产平台，在三个城市开展的大量评估证明了我们方法的有效性和效率。我们希望本研究能为众多工作负载容量有限的服务行业提供有益参考。

参考文献

[1] H. Zuo，“平台型网络的崛起：房地产经纪行业展望”，载于了解中国房地产市场。Springer，2021年，第299–305页。[2] P. Cremonesi、Y. Koren和R. Turrin，“推荐算法在top-n推荐任务中的绩效”，载于*RecSys*，2010年，第39–46页。[3] R. Bekkerman、V. Josifovski和F. J. Provost，“房地产行业的数据科学”，载于*KDD*。ACM，2020年，第3559–3560页。[4] D. Rigney，《style id='11'>马太效应：优势如何带来更多优势》。Columbia University Press，2010年。[5] Y. Zhao、K. Zheng、Y. Cui、H. Su 等人，“空间众包中的预测性任务分配：一种数据驱动的方法”，载于*ICDE*。IEEE，2020年，第13–24页。[6] Y. Zheng、J. Wang、G. Li、R. Cheng 等人，“QASCA：面向众包应用的质量感知任务分配系统”，载于*SIGMOD*。ACM，2015年，第1031–1046页。

[7] Y. Tong、J. She、B. Ding、L. Wang等人，“空间众包中的在线移动微任务分配”，载于ICDE。IEEE，2016年。[8] L. Li、W. Chu、J. Langford和R. E. Schapire，“个性化新闻文章推荐的上下文老虎机方法”，载于WWW。ACM，2010年，第661–670页。[9] D. Zhou、L. Li和Q. Gu，“采用基于UCB的探索的神经上下文老虎机”，载于ICML，第119卷。PMLR，2020年，第11 492–11 502页。[10] Y. Wang、Y. Tong、C. Long、P. Xu等人，“自适应动态二分图匹配：一种强化学习方法”，载于ICDE。IEEE，2019年，第1478–1489页。[11] L. Zhang、T. Hu、Y. Min、G. Wu等人，“基于组合优化的出租车订单调度模型”，载于KDD。ACM，2017年，第2151–2159页。[12] T. Chen和C. Guestrin，“XGBoost：一种可扩展的树提升系统”，载于KDD论文集。ACM，2016年，第785–794页。[13] Z. Xu、Z. Li、Q. Guan、D. Zhang等人，“按需网约车平台中的大规模订单调度：一种学习与规划方法”，载于KDD。ACM，2018年，第905–913页。[14] T. Lattimore和C. Szepesvari，《老虎机算法》。CU Press，2020年。[15] R. S. Sutton和A. G. Barto，《强化学习：导论》。MIT Press，2018年。[16] A. Virmaux和K. Scaman，“深度神经网络的Lipschitz正则性：分析与高效估计”，载于NeurIPS，2018年，第3839–3848页。[17] Z. Chen、P. Cheng、Y. Zeng和L. Chen，“最小化空间众包中任务分配的最大延迟”，载于ICDE。IEEE，2019年，第1454–1465页。[18] T. H. Cormen、C. E. Leiserson、R. L. Rivest和C. Stein，《算法导论》。MIT Press，2009年。[19] T. Abeywickrama、V. Liang和K. Tan，“通过增量成本计算优化现实应用中的二分图匹配”，VLDB，第14卷，第7期，第1150–1158页，2021年。[20] P. Shi、M. Zhao、W. Wang、Y. Zhou等人，“兼得两者：在无预算条件下缓解众包工作者策略选择的不均衡”，Knowl. Based Syst.，第163卷，第1020–1031页，2019年。[21] L. Ramshaw和R. Tarjan，“关于不平衡二分图中的最小成本分配”，2012年。[22] F. F. Nah，“关于可容忍等待时间的研究：网络用户愿意等待多久？”，Behav. Inf. Technol.，第23卷，第3期，第153–163页，2004年。[23] F. Basik、B. Gedik、H. Ferhatosmanoglu和K.-L. Wu，“众包配送中的公平任务分配”，IEEE Trans. Serv. Comput.，第14卷，第4期，第1040–1053页，2021年。[24] K. Christakopoulou、J. Kawale和A. Banerjee，“带容量约束的推荐”，载于CIKM。ACM，2017年，第1439–1448页。[25] Y. Tong、Z. Zhou、Y. Zeng、L. Chen和C. Shahabi，“空间众包：综述”，The VLDB Journal，第29卷，第217–250页，2020年。[26] X. Yu、G. Li、Y. Zheng、Y. Huang等人，“Crowdota：众包中的在线任务分配系统”，载于ICDE。IEEE，2018年，第1629–1632页。[27] G. Li、J. Wang、Y. Zheng和M. J. Franklin，“众包数据管理：综述”，载于ICDE。IEEE，2017年，第39–40页。[28] S. Amer-Yahia和S. B. Roy，“众包中的人为因素”，VLDB，第9卷，第13期，第1615–1618页，2016年。[29] S. B. Roy、I. Lykourentzou、S. Thirumuruganathan等人，“知识密集型众包中的任务分配优化”，The VLDB Journal，第24卷，第4期，第467–491页，2015年。[30] Y. Tong、Y. Zeng、B. Ding等人，“空间众包中的双边在线微任务分配”，IEEE Trans. Knowl. Data Eng.，第33卷，第5期，第2295–2309页，2021年。[31] B. Du、Y. Tong、Z. Zhou等人，“电动汽车共享的需求感知型充电桩规划”，载于KDD。ACM，2018年，第1330–1338页。[32] J. She、Y. Tong、L. Chen等人，“反馈感知的社交活动参与者安排”，载于SIGMOD。ACM，2017年，第851–865页。[33] Y. Tong、Y. Yuan、Y. Cheng、L. Chen和G. Wang，“时空众包数据管理技术综述”，J. Softw.，第28卷，第1期，第35–58页，2017年。[34] L. Kazemi和C. Shahabi，“Geocrowd：利用空间众包实现查询应答”，载于GIS。ACM，2012年，第189–198页。[35] Y. Tong、J. She、B. Ding、L. Chen等人，“实时空间数据中的在线最小匹配：实验与分析”，VLDB，第9卷，第12期，第1053–1064页，2016年。[36] D. Shi、Y. Tong、Z. Zhou等人，“学习分配：迈向大规模网约车中的公平任务分配”，载于KDD。ACM，2021年，第3549–3557页。

[37] Y. Zhao, K. Zheng, J. Guo, B. Yang 等人，“空间众包中的公平性感知任务分配：博弈论方法”，载于 ICDE。IEEE，2021年，第 265–276 页。 [38] Q. Tao, Y. Tong, Z. Zhou, Y. Shi 等人，“空间众包中的差分隐私在线任务分配：一种基于树的方法”，载于 ICDE。IEEE，2020年，第 517–528 页。 [39] H. To、C. Shahabi 和 L. Xiong，“不可信服务器环境下空间众包中的隐私保护在线任务分配”，载于 ICDE。IEEE，2018年，第 833–844 页。 [40] Y. Xu, S. Wei 和 Y. Wang，“拼车平台上的隐私保护在线匹配”，Neurocomputing，第 406 卷，第 371–377 页，2020年。 [41] Y. Tong, L. Wang, Z. Zhou, L. Chen 等人，“空间众包中的动态定价：一种基于匹配的方法”，载于 SIGMOD。ACM，2018年，第 773–788 页。 [42] H. Cheng, S. Wei, L. Zhang, Z. Zhou 等人，“通过竞争吸引网约车司机：竞技场案例研究”，载于 MDM。IEEE，2021年，第 19–28 页。 [43] D. Gao, Y. Tong, J. She 等人，“空间众包中的 Top-k 团队推荐”，载于 WAIM，第 9658 卷。Springer，2016年，第 191–204 页。 [44] D. Shi, Y. Tong, Z. Zhou 等人，“大型机器人仓库的自适应任务规划”，载于 ICDE 2022。IEEE，2022年，第 3327–3339 页。 [45] Y. Fu, H. Xiong, Y. Ge, Z. Yao 等人，“利用地理依赖关系进行房地产估价：排序与聚类的相互视角”，载于 KDD。ACM，2014年，第 1047–1056 页。 [46] M. Grbovic 和 H. Cheng，“利用嵌入实现 Airbnb 搜索排序的实时个性化”，载于 KDD。ACM，2018年，第 311–320 页。 [47] W. Zhang, H. Liu, L. Zha, H. Zhu 等人，“Mugrep：一种用于房地产估价的多任务层次化图表示学习框架”，载于 KDD。ACM，2021年，第 3937–3947 页。 [48] D. Weng, H. Zhu, J. Bao, Y. Zheng 等人，“Homefinder 再探：运用以可达性为中心的多准则决策寻找理想住宅”，载于 CHI。ACM，2018年，第 247 页。 [49] R. Marcus, P. Negi, H. Mao 和 N. T. and，“Bao：让学习式查询优化切实可行”，载于 SIGMOD。ACM，2021年，第 1275–1288 页。 [50] D. Bouneffouf, I. Rish 和 C. C. Aggarwal，“多臂老虎机与上下文老虎机应用综述”，载于 CEC。IEEE，2020年，第 1–8 页。 [51] J. Langford 和 T. Zhang，“适用于具有侧信息的多臂老虎机的 epoch-greedy 算法”，载于 NeurIPS，2007年，第 817–824 页。 [52] D. N. Hill, H. Nassif 等人，“一种用于实时多变量优化的高效老虎机算法”，载于 KDD。ACM，2017年，第 1813–1821 页。 [53] H. Wang, Q. Wu 和 H. Wang，“为上下文老虎机学习隐特征”，载于 CIKM。ACM，2016年，第 1633–1642 页。 [54] Y. Ban, J. He 和 C. B. Cook，“多方面上下文老虎机：神经网络视角”，载于 KDD。ACM，2021年，第 35–45 页。