arXiv:2303.04627v1 [cs.DB] 8 Mar 2023

空间众包中的公平性驱动的技能型任务分配与额外预算

周云军[1], 万书涵[1], 张德田[1*]和文诗婷[2]

[1*]苏州大学人工智能研究院、计算机科学与技术学院，江苏苏州，中国。[2]浙江大学宁波理工学院，浙江宁波，中国。

摘要

随着移动设备的普及和无线网络的无处不在，空间众包受到了学术界和工业界的广泛关注。在空间众包平台上，任务请求者可以发布空间任务，而工作者需要前往目的地执行这些任务。本文正式定义了具有额外预算的技能型任务分配（STAEB）问题，其目标是最大化平台总收益，并实现对工作者和任务请求者的公平性。在STAEB问题中，复杂任务需要一名以上的工作者来满足其技能要求，并且拥有额外预算，用于补贴工作者的额外出行成本，从而吸引更多工作者。我们证明了STAEB问题是NP完全问题。因此，我们提出了两种近似算法来解决该问题，包括一种贪心方法和一种博弈论方法。在真实和合成数据集上进行的大量实验表明，我们提出的方法具有较高的效率和有效性。

关键词：空间众包、任务分配、额外预算、公平性驱动

*通讯作者。电子邮箱：detian@suda.edu.cn。作者：
20214227073@stu.suda.edu.cn；20195227042@stu.suda.edu.cn；
wensht@nit.zju.edu.cn；

1 引言

随着智能设备和高速无线网络的发展，将移动工作者分配给基于位置的任务的空间众包（SC）近年来备受关注。具体而言，工作者需要亲自前往指定地点，完成由任务请求者发布的任务。此类空间众包可应用于许多场景，例如网约车服务（如滴滴和 Uber）、送餐服务（如 Grubhub、饿了么和美团）、交通监测（如 Waze）以及地理数据生成（如 OpenStreetMap）。然而，一些复杂任务不仅需要特定技能，还要求工作者处于一定距离范围内，以确保公平收益。

空间众包中的一个基本问题是任务分配。现有的大多数任务分配研究 [1–6] 主要假设所有任务都很简单，可以由单个工作者轻松完成，例如包裹配送、拍照或热点报告。然而，个别工作者无法完成一些复杂任务，例如筹备派对、房屋装修和全面清洁。比如，房屋装修需要设计图纸、墙面粉刷、铺瓷砖、家具搬运以及水电安装等。因此，平台需要安排具有不同技能的多名工作者，以满足任务的不同需求。此外，分配的公平性也很重要。因为对于同一任务，每位工作者不仅需要提供技能服务，还需承担不同的出行成本。

以往关于空间众包的研究 [7, 8]主要致力于将复杂任务分配给多个工作者，使这些工作者能够满足任务的技能需求，同时忽略了工作者产生的额外出行成本。然而，在现实中，如果工作者希望完成所分配的任务，还需要前往任务地点，这将产生出行成本。因此，工作者只愿意承担预算范围内的出行成本，从而确保其实际收入公平。以图1为例，其中有三名具有不同技能的工作者（以不同颜色表示），以及两名需要相应技能才能完成任务的任务请求者（所需技能同样以不同颜色表示）。任务的固定范围约束用实线表示，额外范围约束用虚线表示。可以看到，工作者 w_2 能够很好地满足任务请求者 t_2 的技能要求，且未超过距离限制。然而，如果任务请求者t_1 希望寻找一名工作者来完成其任务，则没有工作者能够响应，因为满足技能要求的工作者（即 w_1）超出了其距离限制，而在距离限制内的工作者（即 w_3）的技能又无法满足要求。这导致任务请求者 t_1 可能始终无法完成任务分配。在实践中，任务请求者 t_1 可能拥有额外预算，用于补贴工作者 w_1 的额外出行成本，从而使任务得以完成，也有助于实现公平的在线推荐。

因此，本文研究此类问题设置下的空间众包任务分配，即带额外预算的技能型任务分配

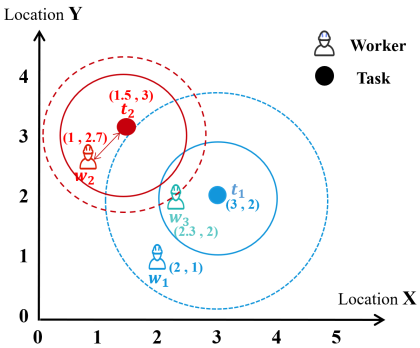


图 1 STAEB问题中距离约束的示例。

（STAEB）。具体而言，给定一组工作者和一组任务，其目标是将多技能工人分配给复杂任务，并提供额外预算。

总之，我们对本文的贡献总结如下：

•我们正式定义了带额外预算的技能型任务分配（STAEB）问题，并证明该问题是NP完全的。

• 我们提出两种基于批次的近似算法来解决 STAEB问题，即贪心算法和博弈论方法。

• 我们在真实和合成数据集上进行了大量实验，以证明所提算法的有效性和效率。

本文其余部分的组织结构如下。第 2 节回顾相关工作。第3节正式定义STAEB问题。第 4 节介绍贪心算法。第 5节提出博弈论算法。第6节展示在真实和合成数据集上进行的大量实验。最后，第 7节总结本文。

2 相关工作

空间众包是匹配型服务中的一个重要主题，已受到移动互联网和共享经济领域学者的广泛关注。其中，任务分配是空间众包中最重要的领域之一，可分为两类 [9]：匹配和规划。在匹配模型中，任务分配通常被表述为一个基于二分图的问题。工人和任务可以用二分图中的顶点表示，而工人与任务之间的效用或成本可以用边的权重表示。因此，该问题旨在获得二分图中的最优匹配。在规划模型（又称调度模型）中，任务分配旨在为每位工人规划一条路线，以执行一系列任务。本节将介绍基于上述两类的空间众包相关研究。

2.1 匹配模型

通常，根据现有研究中的不同目标，匹配问题可以分为效用最大化[9–11],成本最小化 [12–14], 和稳定匹配 [15, 16] 。

效用最大化的目标等价于最大化分配总数或总收益 [9]。由于任务匹配可以表述为二分图匹配问题，因此，二分图匹配问题中的精确算法（例如匈牙利算法 [12]）可以最优地解决该问题。Kazemi 等人 [10]通过将图归约为最大流问题 [11], 的一个实例，并使用 Ford-Fulkerson 算法 [17]，获得了精确结果。此外，还提出了各种贪心算法，以降低精确算法的计算量（例如匈牙利[12]）。To 等人 [2] 考虑了基于该问题中每个任务优先级的贪心匹配。具体而言，他们将位置熵 [10] 的思想扩展到区域熵，即应优先分配给内部工作者较少的任务。

由于每个工人-任务分配都有预算，因此在任务分配中，找到成本最低的匹配也很重要。匈牙利算法 [12]和逐次最短路径算法（SSPA） [13] 可以得出该问题的精确解。此外，Hou 等人 [14] 利用索引（R 树索引）和 I/O 优化技术来提高效率。

稳定匹配试图将工人或任务的偏好纳入优化目标。直观的解决方案是从剩余任务和工作者中迭代选择距离最近的一对 [15]。为了提高效率，Wong 等人 [16] 将“互为最近邻”的概念转化为双色互为最近邻搜索问题，并提出了一种基于最近邻搜索的链式算法。

如上所述，匹配模型更像是一个基于二分图的问题，其中每个任务只分配一名工人。然而，在我们的问题中，一个任务可能需要多名具备不同技能的工人才能完成，因此这些研究并不适用于我们的问题。

2.2 规划模型

任务分配中的规划模型旨在为每位工人规划一条路线（即任务序列）。送餐和拼车在实际应用中都是规划问题。根据规划模型中的任务复杂度（即任务是否需要多位工人协作），我们可以进一步将该模型分为两类：单工人规划 [1–4]和多工作者规划 [7, 18, 19]。

2.2.1 单工人规划

以往大多数研究主要关注将任务分配给单个工作者 [1–6]。Kazemi 等人 [10] 将匹配问题归约为最大流

问题 [20], 并使用 Ford-Fulkerson 算法 [17] 最大化已分配任务的数量。Den等人 [21] 首次研究如何在出行预算和截止期限约束下最大化已执行任务的数量，并证明了该问题具有NP难性。为解决这一问题，他们提出了一种基于动态规划的精确算法，以及若干基于贪心启发式的近似算法。基于贪心启发式的算法包括最近邻启发式（NNH）、最有前景启发式（MPH）和最短过期时间启发式（LEH）。为更好地权衡效率与有效性，他们提出了束搜索启发式（BSH）， [22],该方法将 NNH 中候选集的基础集合扩展至给定阈值。Tong等人 [23] 首次考虑任务分配的在线场景，并提出具有理论保证的基于阈值的算法，以最大化分配的总效用。Song等人 [24] 提出了实时空间众包中的三色在线匹配，其中包含工人、任务和工作地点三类实体。Cheng等人 [25] 提出了一种跨平台在线匹配，使平台能够从其他平台借用部分空闲工人。Zhao等人 [26]首次考虑减少用户的平均等待时间，并完成更多任务，以提升用户体验。

2.2.2 多工作者规划

在实践中，有一些复杂任务需要由工作者组来完成，因为通常单个工作者并不具备全部技能。因此，一些研究者将重点放在多工作者规划上（即多个工作者协作完成一项任务）。规划模式旨在最大化总体效用（例如，满意度评分 [27, 28], 收益 [18, 29–31], 距离 [32–34]）。

具体而言，Shin等人 [18] 提出了一种局部比率算法，以最大化已完成任务的奖励。To 等人 [19] 正式定义了空间众包中的最大任务分配（MTA）问题，并提出了多种解决方案。随后，考虑到不同类型的任务可能需要具备不同技能集合的工人，他们首先根据工人的技能扩展了MTA，提出最高分分配（MSA）问题。Cheng等人 [7] 提出了多技能空间众包（MS-SC）问题，旨在找到最优的工作者与任务分配策略。MS-SC已被证明是NP难问题。因此，他们提出了三种有效的启发式方法来解决该问题，包括贪心算法、g-分治算法和基于成本模型的自适应算法。She等人 [28] 还提出了一种基于RatioGreedy的贪心算法，该算法考虑每个工作者-任务对的效用成本比，并将比率最大的工作者-任务对加入规划中。Gao等人 [27]同时考虑技能和时间约束，以最大化总满意度。他们首先构建一个工人集合，使其基础集合最低，同时满足任务的技能要求，然后贪心地将满意度最高的工人分配给该任务。Gao等人 [35] 为任务推荐排名前k的工人群组，并在每个群组中选择一名工人担任组长。Cheng等人 [8] 考虑多个工人之间的协作关系，以最大化任务的总协作质量收益。实时空间众包的在线算法由

Song等人首次提出 [36]。Online-Exact算法始终为新出现的任务或工人计算最优分配，而Online-Greedy算法则用于快速计算任务分配。Li等人 [37]提出了考虑任务群组偏好的群组任务分配。Ni等人 [38] 考虑到任务之间可能存在依赖关系。也就是说，只有在其依赖任务已被分配后，某个任务才能被调度。

然而，这些方法并未考虑带有额外预算的任务的紧急需求（如图 1 所示），因此有些任务可能需要等待很长时间，甚至可能永远得不到服务。这对这些任务而言确实不公平，也会给平台客户带来糟糕的体验。总之，所有这些现有论文主要关注任务-工人匹配，却忽视了其中存在的公平性问题（例如，距离任务较远的工人）。本文将任务的额外预算纳入考量，并提出公平性驱动的任务分配算法，以高效、有效地解决这些问题。

3 问题定义

本节将正式定义具有额外预算的技能型任务分配（STAEB）问题。

假设 $S = \{s_1, s_2 ... s_k\}$ 是一组 k 技能。每个工人都拥有 S 中的一种或多种技能，而每个任务都需要 S 中的一种或多种技能。不同的技能要求任务请求者支付不同的费用 p_s。

定义 1（任务）任务记为 $t =< l_t, a_t, r_t, b_t, S_t >$，在时间 a_t 于二维空间中的位置 l_t 发布到平台上。r_t 是 t 的半径，也是 t 的固定范围约束；b_t 是其提供的额外预算，t 需要技能 $S_t \subseteq S$。

定义 2（工人）工人记为 $w =< l_w, a_w, S_w >$，在时间 a_w 出现在平台上，位置为二维空间中的 l_w。w 拥有技能 $S_w \subseteq S$。

定义 3（出行成本）出行成本记为 $cost(t, w)$，由从 l_w 到 l_t 的旅行距离决定。

旅行距离可以用任何类型的距离来衡量，例如欧几里得距离或道路网络距离。为简化起见，本文使用欧几里得距离作为旅行距离，并直接将其视为出行成本。

定义 4（额外出行成本）额外出行成本记为 e_t，是指超过固定范围约束的实际出行成本，即 $e_t = cost(t, w) - r_t$。

定义 5（有效工人集合）每项任务都需要由多名工人共同完成。能够完成该任务的多名工人组成的集合称为有效工人集合，其中工人的技能集可以覆盖任务所要求的技能。任务的每项技能要求只需由一名

工人满足，但不同工人之间可以有重复技能。因此，有效工人集合中的每名工人都必须至少拥有一项任务所需、而其他工人不具备的技能。有效工人集合记为 $W_v(t)$，是满足以下三个条件的工人集合：

（1）任务的每项技能都可以由集合中的技能覆盖，即，$s \in \bigcup_{w \in W_v(t)} S_w$, $\forall s \in S_t$,

（2）集合中工作者的额外出行成本总和应小于任务的额外预算，即，$\sum_{w \in W_v(t)} e(t, w) \leq b_t$。

定义 6（平台收益）给定一个任务 t 和一个工作者 w，平台收益表示为：

$$p_{(t,w)} = \begin{cases} \alpha \sum_{s \in S_w \cap S_t} p_s, & cost(t,w) < r_t, \\ \alpha \sum_{s \in S_w \cap S_t} p_s - \beta e(t,w), & cost(t,w) \geq r_t. \end{cases} \tag{1}$$

其中，$\sum_{s \in S_w \cap S_t} p_s$ 表示工人为任务提供所需技能所获得的总收益。α（$0 < \alpha < 1$）是控制平台从技能费用中获得的收入的参数。我们假设平台技能收入与技能费用成正比。工人只愿意接受一定范围内的出行成本。因此，任务和平台将补贴工人的额外出行成本。β（$0 < \beta < 1$）是控制平台为工人的额外出行成本提供补贴的参数。

当工人前往任务地点的出行成本小于任务的固定范围约束时，平台收益等于技能收入的奖励。当工人前往任务地点的出行成本大于或等于任务的固定范围约束时，平台需要为工人超出任务固定范围约束部分的出行成本提供补贴。因此，平台收益等于技能收入减去额外出行成本。平台从任务中获得的收益等于分配到该任务的工人的收益，记为：

$$P_{<t, W_v(t)>} = \sum_{w \in W_v(t)} p_{(t,w)} \tag{2}$$

定义 7（带额外预算的技能任务分配（STAEB）问题）给定任务集合 T 和工作者集合 W。可行匹配结果记为 M，由一组 $< task, valid\ worker\ set >$ 组成，其形式为 $< t_1, W_v(t_1) >, <t_2, W_v(t_2) >, ..., < t_{|T|}, W_v(t_{|T|}) >$，其中 $\cap_{i=1}^{|T|} W_v(t_i) = \emptyset$。我们的问题是寻找一个可行匹配结果 M，以公平地实现最大化平台总收益的目标 $P_M = \sum_{<t, W_v(t)> \in M} P_{<t, W_v(t)>}$。

定理 1 *The Skilled Task Assignment with Extra Budget problem is an NP-complete problem.*

Proof 我们通过将带额外预算的技能任务分配问题归约为最大权独立集问题，证明该问题是 NP完全的。在我们的问题中，首先可以将所有 $t \in T$ 中每个可用的 $W_v(t)$初始化为图中的一个顶点。然后，将属于同一 t 的所有 $W_v(t)$顶点依次连接起来。最后，将存在交集（即拥有相同工人）的 $W_v(t)$顶点连接起来。此时，任何可以分配给 T 的 $W_v(t)$顶点集合都必须是独立集（否则至少存在一个 t 拥有多个 $W_v(t)$，或有两个 $W_v(t)$由同一工人承担）。由于每个 $W_v(t)$都有收益，平台的收益也等于独立子集中各顶点权重之和，因此成功将 STAEB 问题归约为最大权独立集问题。由于最大权独立集问题是 NP完全问题，STAEB 问题也属于 NP完全问题。□

4 贪心算法

本节提出一种贪心方法 [38]。该算法的主要思路是按照技能平均费用降序排列任务，然后找出能够覆盖任务所需技能的最少工人。不同技能的费用各不相同，因此，工人完成费用较高的技能可获得更高收益。如果一名工人能够满足某项任务所需的多种技能，就能节省出行成本。因此，对于每项任务，优先选择规模最小的有效工人集合。

算法1展示了贪心算法的详细步骤。第1行将匹配对集合初始化为空，并将初始未匹配工人集合设为输入的工人集合。第2行按照技能平均费用降序排列任务。第3至4行将每项任务的有效工人集合设为空。当任务的未匹配技能集不为空时，我们按如下方式分配工人：第6至7行中，任务的总范围约束由固定范围约束和额外范围约束组成。 W^* 是能够完成该任务的工人集合。如果能够完成该任务的工人集合为空，则表示没有工人能够完成该任务剩余的技能，因此该任务无法完成，并将有效工人集合 W_v（t）中的工人加入工人集合 W'。接着，在第13至17行中，我们从 W^* 中选出拥有最多任务所需技能的工人 w ，并将 w 加入该任务的有效工人集合。我们更新额外范围约束和任务所需的技能集，并从工人集合中移除已分配的工人。第19至20行中，当有效工人集合不为空且技能集为空时，表示有效工人集合能够覆盖该任务所需的技能。我们将该任务及其有效工人集合加入匹配对集合，并从任务集合中移除该任务。最后，在第23行返回匹配对集合。

复杂度分析。 任务排序的时间复杂度为 $O(|S_t|\ \{T|+|T|\cdot\ln\ |T|)$，为每个任务寻找有效工人集合的时间复杂度为 $O(|T|\cdot|S_t|^2\cdot|W|)$，因此总时间复杂度为 $O(|S_t|\cdot|T|+|T|\cdot\ln\ |T|+|T|\cdot|S_t|^2\cdot|W|)$。

公平性驱 动的空间众包带额外预算的技能任务分配 g 9

算法1 贪心算法

输入： 任务集合 T，工作者集合 W
输出： 匹配对集合 M

$M \leftarrow \emptyset$, $W' \leftarrow W$;
排序 the 任务 in 降序 排列 根据 $\frac{\sum_{s \in S_t} p_s}{|number\ of\ task\ skills|}$ to the size of;
对于每个任务 $t \in T$ **执行**
  $W_v(t) \leftarrow \emptyset$;
  当 $S_t \neq \emptyset$ **时执行**
    $R_t = r_t + b_t$;
    $W^* = \{w | S_w \cap S_t \neq \emptyset and cost(t, w) \leq R_t\}$;
    如果 $W^* == \emptyset$ **则**
      $W \leftarrow W + W_v(t)$;
      $W_v(t) \leftarrow \emptyset$;
      中断;
    结束如果
    $w = argmax_{w \in W^*} |S_w \cap S_t|$;
    $W_v(t) = W_v(t) \cup \{w\}$;
    $b_t = b_t - e(t, w)$;
    $S_t = S_t - \{S_w \cap S_t\}$;
    $W = W - \{w\}$;
  结束 while 循环
  如果 $W_v(t) \neq \emptyset$ **则**
    $M \leftarrow M \cup \{< t, W_v(t) >\}$;
  结束如果
结束循环
返回 M;

5 博弈论算法

贪心算法能够有效地找到结果。然而，该算法假设工人会按照平台的分配完成任务，并忽略了不同工人之间的竞争。STAEB问题的本质是，每个任务都需要由多名具备所需技能的工人共同完成，这意味着工人的选择会受到其他工人决策的影响。每个工人都希望选择收益较高的任务。这种相互依赖的决策可以通过博弈论建模，其中工人可视为参与博弈的独立参与者。因此，STAEB问题可以形式化为一个多人博弈。已有许多关于博弈论模型的研究 [39–41]。基于现有的博弈论模型研究，本节提出一种博弈论方法，寻找任务的有效工人集合，直至达到纳什均衡。

5.1 博弈建模

我们的 STAEB问题可以建模为一个由博弈参与者、整体策略和效用函数组成的 $n-$人策略博弈$\mathcal{G} =< W,\ \mathbb{Y},\ \mathbb{U} >$。具体建模如下:

(1) $W = \{w_1, w_2, ..., w_n\}(n \geq 2)$ 是由有限数量的工人构成的集合，这些工人作为博弈参与者。在本文其余部分中，我们将交替使用“博弈参与者”和“工人”这两个术语。

(2) $\mathbb{Y} = \cup_{i=1}^{n} Y_i$ 是博弈参与者的整体策略。 Y_i 是工人 w_i 可以选择的有限策略集合。

(3) $\mathbb{U} = \cup_{i=1}^{n} U_i$ 表示所有博弈参与者的效用函数。 U_i 的值取决于博弈参与者 w_i 以及其他博弈参与者的策略。 $U_i:\ \mathbb{Y} \to \mathbb{P}$ 是博弈参与者 w_i的效用函数。对于每个联合策略 $\vec{y} \in \mathbb{Y}$， $U_i(\vec{y}) \in \mathbb{P}$表示博弈参与者 w_i的效用，可按如下方式计算:

$$U_i(\overrightarrow{y}) = p_{(t,w_i)} - p_{(t_0,w_i)} \tag{3}$$

其中， $p_{(t,w_i)}$ 表示将博弈参与者 w_i 分配至任务 t所带来的平台收益。$p_{(t_0,w_i)}$ 表示将博弈参与者 w_i 分配至任务 t_0所带来的平台收益。

设 y_i 为联合策略 Y_i 中博弈参与者 w_i 的策略， y_{-i} 为除博弈参与者 w_i 之外所有其他博弈参与者的联合策略。当且仅当对于每个满足以下条件的博弈参与者 $w_i \in W$，策略博弈存在纯纳什均衡 $Y^* \in \mathbb{Y}$：

$$U_i({y_i}^*, {y_{-i}}^*) \geq U_i(y_i, {y_{-i}}^*), \forall y_i \in Y_i, y_i^* \in Y^* \tag{4}$$

在我们的问题中，由于每位工人都需要采用确定性策略，即选择一项任务或什么都不做，因此我们只考虑确定性策略。这意味着，多人参与的工人 w_i 可从 Y_i 中选择的概率为 Y_i 1，而 [42]中其余策略的概率均为0。接下来，我们证明 STAEB 博弈是一个精确势博弈（EPG），且至少存在一个纯纳什均衡。

在纳什均衡中，当其他博弈参与者坚持当前策略时，任何参与者都无法通过单方面改变自己的策略来提高效用。我们首先介绍精确势博弈理论。

定义 8 （精确势博弈）当且仅当存在一个势函数 $\phi: \mathbb{Y} \to \mathbb{P}$，使得对于所有 $\vec{y} \in \mathbb{Y}$ 和 $w_i \in W$，策略博弈 $\mathcal{G} =< W, \mathbb{Y}, \mathbb{U} >$满足以下条件:

$$U_i(y_i, y_{-i}) - U_i({y_i}', y_{-i}) = \phi(y_i, y_{-i}) - \phi({y_i}', y_{-i}), \forall y_i, {y_i}' \in Y_i \tag{5}$$

其中， y_i 和 ${y_i}'$ 是工人 w_i 可以选择的策略， y_{-i} 是除工人 w_i之外其他工人的联合策略。

定理 2 *Our STAEB problem is an Exact Potential Game(EPG).*

公平性驱 动的带额外预算的技能任务分配，适用于空间众包 g 1

Proof T中所有任务的平台总收益由势函数 $\phi(y)=\sum_{t\in T} P_{<t,\ W_v(t)>}$表示。工人 w_i 可以在 y_i 和 y_i' 两种策略之间进行选择，而 y_{-i} 表示除工人 w_i以外所有工人的策略。策略 y_i' 中选择的任务记为 t_j 和 t_k。于是有：

$$\begin{aligned}
&\phi(y_i, y_{-i}) - \phi(y_i', y_{-i}) \\
&= p_{<t_j, W_v(t_j)>} + p_{<t_k, W_v(t_k)-w_i>} + \sum_{t\in T-t,j-t_k} p_{<t, W_v(t)>} - (p_{<t_k, W_v(t_k)>} \\
&+ p_{<t_j, W_v(t_j)-w_i>} + \sum_{t\in T-t_j-t_k} p_{<t, W_v(t)>}) \\
&= p_{<t_j, W_v(t_j)>} + p_{<t_k, W_v(t_k)-w_i>} - (p_{<t_k, W_v(t_k)>} + p_{<t_j, W_v(t_j)-w_i>}) \\
&= p_{<t_j, W_v(t_j)>} - p_{<t_j, W_v(t_j)-w_i>} - (p_{<t_k, W_v(t_k)>} - p_{<t_k, W_v(t_k)-w_i>}) \\
&= p_{(t_j, w_i)} - p_{(t_k, w_i)} \\
&= (p_{(t_j, w_i)} - p_{(t_0, w_i)}) - (p_{(t_k, w_i)} - p_{(t_0, w_i)}) \\
&= U_i(y_i, y_{-i}) - U_i(y_i', y_{-i})
\end{aligned} \tag{6}$$

因此，根据定义，STAEB问题的策略博弈是一个精确势博弈。□

5.2 博弈论方法

由于我们的 STAEB 博弈存在纯纳什均衡，我们提出了一种基于最佳响应框架的额外预算感知博弈论方法（EBGT），以求得该策略博弈的纳什均衡联合策略$\mathcal{G}$。在该算法中，每位工人都会被分配到对其而言“最佳”的任务，从而获得更高的平台总收益。博弈论方法的详情分为两个步骤，在算法 2 中进行说明。

步骤 *1*：初始化策略。在第 2 行，将匹配集合初始化为空，并分别将任务和工人的输入设为初始的未匹配任务与工人集合。在第 3 至 4 行，将每个任务的有效工人集合设为空。在第 5 至 12 行，我们遍历任务的每项技能，并查找能够完成该任务的工人。如果能够完成技能任务的工人集合为空，则意味着没有工人能够完成该任务剩余的技能。因此，该任务无法完成，我们将未匹配工人集合恢复原状。在第 13 至 14 行，我们选择在 W^* 中平台收益最高的工人，并更新任务所需的技能集及其额外范围约束。同时，我们从工人集合中移除已分配的工人。当有效工人集合能够覆盖任务所需的技能时，我们在第 17 行将该任务及其有效工人集合加入匹配对集合。

步骤 *2*：寻找纳什均衡。该算法根据其他工人当前的联合策略，迭代调整每个工人的策略，以获得最佳响应策略，直至找到纳什均衡，即无人会改变自己的策略。此时，平台收益将达到最大值。在每次迭代中，只允许一名工人选择最佳博弈，并且应按顺序进行博弈。我们记录匹配结果

12 公平性 驱动的带额外预算的技能任务分配：空间众包中的 n_{g}

算法 2 考虑额外预算的博弈论方法（EBGT） ch

输入：任务集合 T，工作者集合 W输出：匹配对集合 M1: // 步骤 1：初始化策略。 2: $M \leftarrow \emptyset$, $T' \leftarrow T$, $W' \leftarrow W$; 3: 对每个任务 $t \in T$ 执行 4: $W_v(t) \leftarrow \emptyset$; 5: 对每个任务技能 $s \in S_t$ 执行 6: $R_t = b_t$; 7: $W^* = \{w | s \in S_w$ 和 $cost(t, w) \leq R_t\}$; 8: 如果 $W^* = \emptyset$ 则 9: $W' \leftarrow W' + W_v(t)$; 10: $W_v(t) \leftarrow \emptyset$; 11: 跳出； 12: 结束条件 13: $w = argmax_{w \in W^*} P_{(t,w)}$; 14: $W_v(t) = W_v(t) \cup \{w\}$, $b_t = b_t - e$ (t, w) $W' = W' - \{w\}$; 15: 结束循环 16: 如果 $W_v(t) \neq \emptyset$ 则 17: $M \leftarrow M \cup \{< t, W_v(t) >\}$, $T' = T' - \{t\}$; 18: 结束条件 19: 结束循环 20: // 步骤 2：求纳什均衡。 21: 重复 22: $M' \leftarrow M$; 23: 对每个工人 $w \in W$ 执行 24: 为 w_i查找最佳响应任务 t^*； 25: 如果 t^* 不存在则 26: 继续； 27: 否则 28: 如果 $t^* \in T'$ 则 29: 获取包含 w_i 的 $W_v(t^*)$; 30: $M \leftarrow M \cup \{< t^*, W_v(t^*) >\}$, $T' = T' - \{t^*\}$; 31: 否则 32: $W' \leftarrow W' + W_v(t^*)$; 33: $M \leftarrow M - \{< t^*, W_v(t^*) >\}$; 34: W_v $(t^*) \leftarrow \emptyset$; 35: 获取包含 w_i的 $W_v(t^*)$; 36: $M \leftarrow M \cup \{< t^*, W_v(t^*)$ $>\}$, $T' = T' - \{t^*\}$; 37: 结束条件 38: 结束条件 39: 比较 M 和 M'; 40: 结束循环 41: 直到达到纳什均衡 42: 更新 M; 43: 返回 M;

公平性驱 动的带额外预算的技能任务分配（空间众包） g 13

第 22 行中的上一轮。对于每个工人 $w_i \in W$，我们首先在第 24 行查找最佳响应任务 t^*，其计算方式如下:

$$t^* = argmax_{t \in \{t|S_{w_i} \cap S_t \neq \emptyset \ and \ cost(t,w_i) \leq R_t\}} (p_{<t,W_v(t)>} - p_{<t,W_v(t)-\{w_i\}>} - (p_{<t_0,W_v(t_0)>} - p_{<t_0,W_v(t_0)-\{w_i\}>})) \quad (7)$$

如果根据当前任务分配，工人没有最佳响应任务， w_i 就不会改变其策略。在第 28–30 行中，当 w 有最佳响应任务且该最佳响应任务的有效工人集合为空时，我们会得到包含 w_i的有效工人集合。我们将 t^* 和一个有效工人集合加入匹配集合。在第 31–38 行中，当最佳响应任务的有效工人集合不为空时，我们清空该任务当前的有效工人集合，并得到包含 w_i的有效工人集合。然后，将 t^* 和一个有效工人集合加入匹配集合。最后，根据纳什均衡更新 M。

5.3 博弈论方法分析

由于 STAEB问题是一个精确势博弈，且策略集合 S有限，因此工人经过有限轮策略变更后即可达到纳什均衡。为简单起见，我们通过考虑该问题的一个缩放版本，证明达到纯纳什均衡所需总轮数的上界。为验证达到纯纳什均衡所需总轮数的上界，我们探讨了该问题的一个缩放版本。该问题的目标函数取整数值。我们假设存在一个具有势函数的可比博弈： $\phi_{\mathbb{Z}}(S) = d \cdot \phi(S)$，其中 d 是一个正乘法因子，使得 $\phi_{\mathbb{Z}}(S) \in \mathbb{Z}, \forall S \in \mathbb{S}$。我们证明，GT技术使用该缩放势函数至多执行 $\phi\mathbb{Z}(S^*)$轮，其中 S^* 是工人在该势STAEB博弈中可选的最优策略。 $\phi_{\mathbb{Z}}(S^*)$是正乘法因子 d 与目标函数最优值 P_M的乘积。

引理 1 *The upper bound on the number of rounds to converge to a pure Nashequilibrium is* $\phi_{\mathbb{Z}}(S^*)$ *in each batch with the EBGT technique, where* S^* *is the optimal joint strategy that workers can select in the potential STAEB game, and* $\phi\mathbb{Z}(S^*) = d \cdot \phi\ S^*$ *is a scaled potential function with only integer values.* ()

Proof 当没有工人偏离其当前策略时，EBGT 方法收敛，这意味着每一轮中至少有一名工人偏离其当前策略。由于 $\phi_{\mathbb{Z}}(S) \in \mathbb{Z}$，每名工人 w_i 都从当前策略 $s_i{}'$ 转向更优策略 s_i，使缩放后的势函数至少增加 1，即 $U_i(s_i, s_{-i}) - U_i(s_i{}', s_{-i}) \geq 1$。因此，收敛到纯纳什均衡所需轮数的上界为最大值 $\phi_{\mathbb{Z}}(S^*)$。

表1 实验设置

参数	设置
固定范围约束 r	600、800、**1000**、1200、1400
额外范围约束 b	(400,600)、600、800)、(**800,1000**)、(1000,1200)、(1200,1400)
技能数量 S	10, 11, **12**, 13, 14
任务数量 T	800, 900, **1000**, 1100, 1200
工人数量 W	2400, 2700, **3000**, 3300, 3600

我们最佳联合策略中任务 t_j 的平台收益上界为:

$$\overline{p}_{t_j} = \sum_{w_i \in W_v(t_j)} p_{(t_j, w_i)} \tag{8}$$

其中， $W_v(t_j)$ 是分配给任务 t_j 的工人， p_i 是完成任务的工人所带来的最大平台收益。因此，上界为 $\phi_{\mathbb{Z}}(S^*) = d \cdot \sum_{t_j \in T} p_{<t_j\ W_v\ t_j\ ^*>} = d \cdot \overline{p_{t_j}}$ □，()。

复杂度分析。 时间复杂度为 $O(|T| \cdot |S_t| \cdot |W|^2 + |W|^2 \nmid S_t| \cdot d \cdot p_{t_j})$，其中 $|T|$ 是任务数量， $|W|$ 是工人数量，$|S_t|$ 是技能的最大数量， $d \cdot p_{t_j}$ 是调整每个工人最佳响应策略直至达到纳什均衡所需的迭代次数。

6 实验研究

6.1 实验设置

我们在实验中使用两个数据集。对于真实数据集，我们使用滴滴出行的出租车数据 [43]，，其中包含2016年11月1日至11月30日成都的订单数据。订单数据包含上车点和下车点、计费时间的起止时间等信息。我们使用上车地点和计费开始时间分别作为任务的位置信息和到达时间。由于乘客下车后工人会再次有空，我们将下车地点作为工人的位置，并将计费结束时间作为工人的到达时间。对于合成数据集，我们在成都的一个矩形区域内随机生成任务请求和工人。工人和任务请求的到达时间在一天内均服从均匀分布。表 1 展示了我们的实验设置，其中参数的默认值以粗体显示。

对比算法。我们评估了代表性算法的性能，即随机算法（RAN）、贪心算法（GRY）和额外预算感知博弈论算法（EBGT）。所有算法均使用 Java 实现，并在配备 Intel(R) Core (TM) i7-7700 CPU @ 3.60GHz 和 16 GB RAM 的机器上运行。

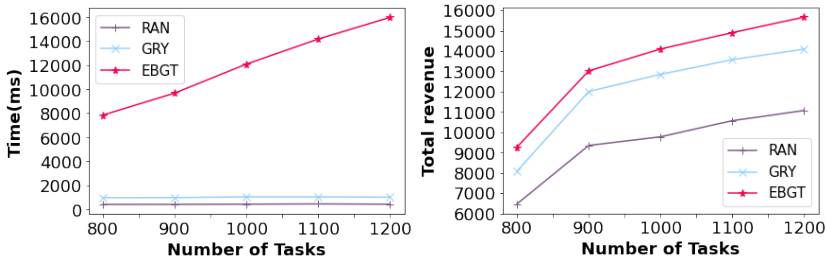
6.2 真实数据集上的结果

6.2.1 任务数量的影响。

本节中，我们探讨任务数量对算法的影响。如图2(a)所示，算法的运行时间随任务数量的增加而增加。EBGT 的运行时间比 RAN 和 GRY 更长，因为它需要经过多次迭代，持续地找到每一轮的最优解。然而，即使任务数量达到 1200，运行时间也是完全可以接受的，即约 16 秒。在图2(b)中，显然，平台收益随任务数量的增加而增加。这是因为可以完成更多任务，从而产生更多的平台收益。EBGT 获得了最高的平台收益，因为它为每名工人找到了最佳响应任务。

6.2.2 工人数量的影响。

如图 3（a）所示，算法的运行时间随着工人数量的增加而增加。由于需要经过多轮迭代为每个工人寻找最优匹配，EBGT 的运行时间比 RAN 和 GRY 更长。从图 3（b）中可以看出，平台收益随着工人数量的增加而增加。这是因为更多的工人意味着平台可以选择能够为平台带来更多收益的工人来完成任务。由于能够将更合适的工人与任务进行匹配，EBGT 仍然获得了最高的平台收益。

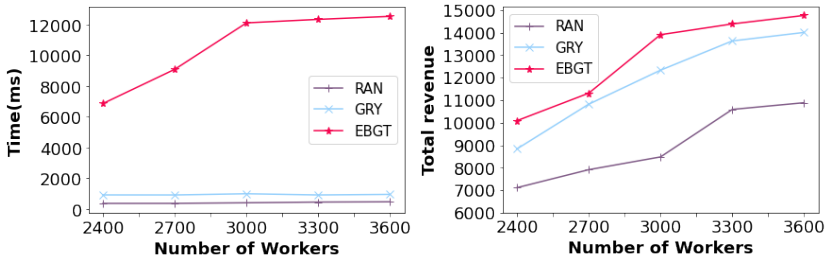


（a）运行时间　　（b）总收益

图 2 真实数据集上任务数量的影响

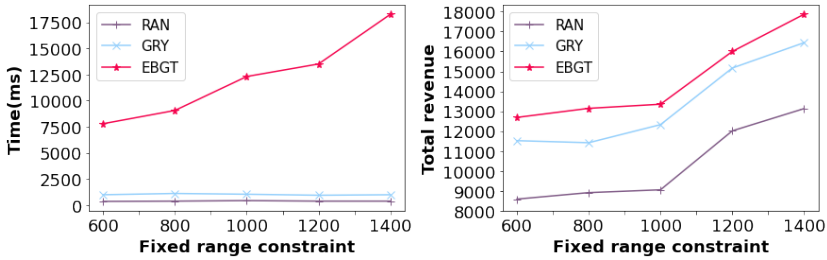
6.2.3 固定范围约束的影响。

如图所示4(a)，随着固定范围约束增大，算法的运行时间也随之增加。这是因为位于每个任务固定范围约束内的工人更多，因此需要更长的运行时间来处理。EBGT 的运行速度比 RAN 和 GRY 慢，因为它需要经过多轮迭代，才能不断找到每一轮的最优解。如图所示的平台收益结果4(b)，随着固定



(a) 运行时间 (b) 总收益

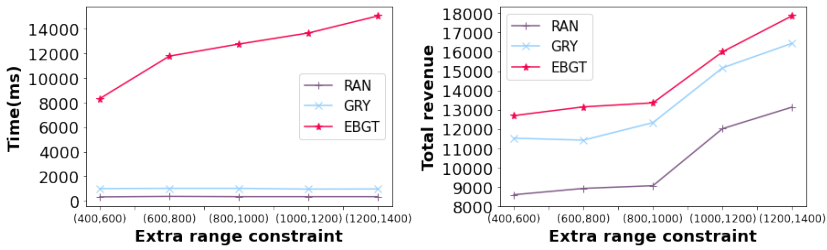
图 3 真实数据集中工人数量的影响



(a) 运行时间 (b) 总收益

图4 固定范围约束对真实数据集的影响

范围约束增大，平台收益也随之增加。EBGT 的平台收益最高，因为它为每位工人找到了最佳响应任务。

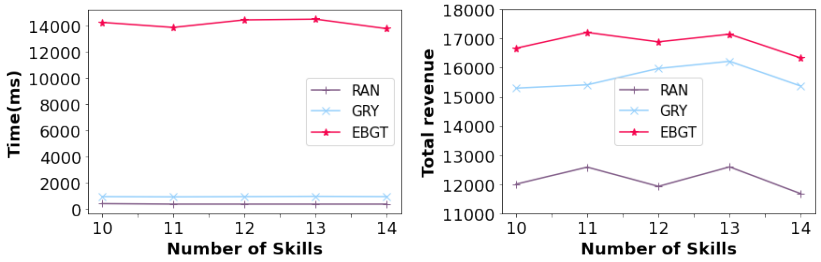


(a) 运行时间 (b) 总收益

图5 额外范围约束对真实数据集的影响

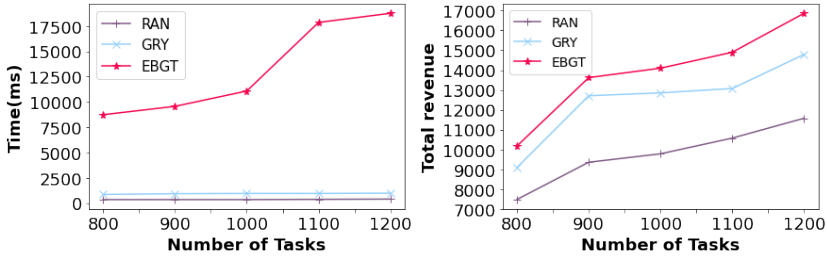
6.2.4 额外范围约束的影响。

如图所示 5(a)，随着额外范围约束增大，算法的运行时间也会增加。这是因为每个任务的额外范围约束内会有更多工人，因此需要更长的运行



(a) 运行时间 (b) 总收益

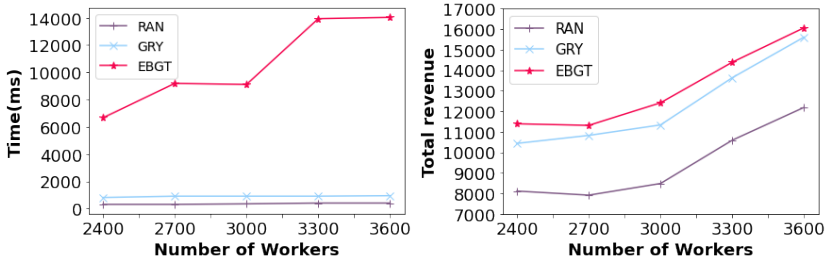
图6 真实数据集上技能数量的影响



(a) 运行时间 (b) 总收益

图7 任务数量对合成数据集的影响

时间。由图 5(b) 可以看出，随着额外范围约束增大，所有算法的平台收益都会增加，这是因为匹配的配对更多。EBGT 持续获得最高的平台收益，因为它会为每个工作者寻找最佳响应任务。



(a) 运行时间 (b) 总收益

图8 工人数量对合成数据集的影响

6.2.5 技能数量的影响。

如图6 (a)所示，随着技能数量变化，各算法的运行时间变化不大。这是因为随着任务所需技能数量的增加，能够满足要求的任务数量会减少，从而减少迭代次数，进而使运行时间保持稳定。EBGT 的运行时间仍然比另外两种算法长，因为它需要经过多轮迭代，才能不断找到每一轮的最优解。从图8(b)中可以看出，三种算法的总收益变化不大。EBGT 获得的平台收益最高，而 RAN 获得的平台收益最低。

6.3 合成数据集上的结果

6.3.1 任务数量的影响。

如图 7（a）所示，随着任务数量的增加，算法的运行时间也随之增加。同样，EBGT 的运行时间比 RAN 和 GRY 更长。如图 7（b）所示，随着任务数量的增加，平台收益也随之增加。这是因为能够完成更多任务，从而产生更多平台收益。EBGT 仍然获得最高的平台收益。总的来说，实验结果与上述真实数据集上的结果相似。

6.3.2 工人数量的影响。

如图 8（a）所示，随着工人数量的增加，算法的运行时间也随之增加。EBGT 的运行时间比 RAN 和 GRY 更长，因为它需要经过多轮迭代才能找到最优解。如图 8（b）所示，随着工人数量的增加，平台收益也随之增加。这是因为更多工人可以完成任务，从而产生更多平台收益。EBGT 获得最高的平台收益。结论是，该实验结果也与上述真实数据集上的结果相似。

7 结论

本文研究空间众包中的具有额外预算的技能型任务分配（STAEB）问题，其中每个有额外预算的任务可能需要多名具备多种技能的工作者才能完成。我们提出了两种近似算法，分别采用贪心方法和博弈论方法。具体而言，贪心方法按照技能平均费用对任务排序，并以贪心方式分配尽可能少的工作者来满足任务所需的技能。此外，我们还提出了一种博弈论方法，以进一步提高平台总收益。在真实和合成数据集上进行的大量实验表明，我们提出的方法具有良好的效率和可扩展性。

公平性驱 动的空间众包带额外预算的技能任务分配 *g* 19

补充材料声明：
STAEB 的源代码已随投稿附上 —，我们的出租车数据可在 [43] 获取，我们的合成数据集可通过**gMission**[44]生成。

参考文献

[1] Zhao, Y., Zheng, K., Cui, Y., Su, H., Zhu, F., Zhou, X.：空间众包中的预测性任务分配：一种数据驱动方法。载于：ICDE，第13–24页（2020年）。IEEE[2] To, H., Fan, L., Tran, L., Shahabi, C.：预算约束下超本地空间众包中的实时任务分配。载于：PerCom，第1–8页（2016年）。IEEE[3] Chen, Z., Cheng, P., Chen, L., Lin, X., Shahabi, C.：空间众包中的公平任务分配。VLDB **13(12)**，2479–2492（2020年）[4]Liu, J.-X., Xu, K.：空间众包中的预算感知在线任务分配。WWW **23(1)**，289–311（2020年）[5] Tong, Y., Zeng, Y., Ding, B., Wang, L., Chen, L.：空间众包中的双边在线微任务分配。TKDE（2019年）[6] Zheng, B., Huang, C., Jensen, C.S., Chen, L., Zheng, K.：空间众包中的在线三色取送调度。载于：2020年 IEEE 第36届数据工程国际会议（ICDE）（2020年）[7] Cheng, P., Lian, X., Chen, L., Han, J., Zhao, J.：面向多技能的空间众包任务分配。TKDE**28(8)**，2201–2215（2016年）[8] Cheng, P., Chen, L., Ye, J.：空间众包中的协作感知任务分配。载于：2019年 IEEE 第35届数据工程国际会议（ICDE），第1442–1453页（2019年）。IEEE[9] Tong, Y., Zhou, Z., Zeng, Y., Chen, L., Shahabi, C.：空间众包：综述。VLDBJ **29(1)**，217–250（2020年）[10] Kazemi, L., Shahabi,C.：Geocrowd：通过空间众包实现查询应答。载于：SIGSPATIAL，第189–198页（2012年）[11]Waissi, G.R.：网络流：理论、算法与应用。JSTOR（1994年）[12] Burkard, R., Dell'Amico, M., Martello, S.：分配问题：修订重印版。工业与应用数学学会，???（2012年）

[13]Derigs, U.：解决最小完美匹配问题的最短增广路径方法。Networks **11**(4)，379–390（1981）[14]LEONG, H.U.、Yiu, M.L.、Mouratidis, K.、Mamoulis, N.：空间数据库中的容量约束分配（2008）[15] Corral, A.、Manolopoulos, Y.、Theodoridis, Y.、Vassilakopoulos, M.：空间数据库中的最近点对查询。ACM SIGMOD Record **29**(2)，189–200（2000）[16] Wong, R.C.-W.、Tao, Y.、Fu, A.W.-C.、Xiao, X.：关于高效空间匹配。载于：第33届超大型数据库国际会议论文集，第579–590页（2007）[17] Ford, L.R.、Fulkerson, D.R.：网络中的最大流。Canadian journal of Mathematics**8**，399–404（1956）[18] He, S.、Shin, D.-H.、Zhang, J.、Chen, J.：面向众感知中位置相关任务的最优分配。载于：IEEE INFOCOM 2014-IEEE计算机通信会议，第745–753页（2014）。IEEE[19] To, H.、Shahabi, C.、Kazemi, L.：一种服务器分配的空间众包框架。ACM Transactions on Spatial Algorithms and Systems (TSAS) **1**(1)，1–28（2015）[20] Ahuja, R.K.、Magnanti, T.L.、Orlin, J.B.：Network flows（1988）[21]Deng, D.、Shahabi, C.、Demiryurek, U.：最大化空间众包中工人自选任务的数量。载于：第21届ACM SIGSPATIAL地理信息系统进展国际会议论文集，第324–333页（2013）[22] Deng, D.、Shahabi, C.、Demiryurek, U.、Zhu, L.：从工作者视角进行空间众包中的任务选择。GeoInformatica **20**(3)，529–568（2016）[23] Tong, Y.、She, J.、Ding, B.、Wang, L.、Chen, L.：空间众包中的在线移动微任务分配。载于：ICDE，第49–60页（2016）。IEEE[24] Song, T.、Tong, Y.、Wang, L.、She, J.、Yao, B.、Chen, L.、Xu, K.：实时空间众包中的三色在线匹配。载于：ICDE，第1009–1020页（2017）。IEEE[25] Cheng, Y.、Li, B.、Zhou, X.、Yuan, Y.、Wang, G.、Chen, L.：空间众包中的实时交叉在线匹配。载于：ICDE，第1–12页（2020）。IEEE

[26] Peng, W.、Liu, A.、Li, Z.、Liu, G.、Li, Q.：用户体验驱动

空间众包中的安全任务分配。WWW **23**(3), 2131–2151 (2020)[27] Gao, T. Dawei：空间众包中的面向团队的任务规划。载于：Web 与大数据 (2017)[28] She, J., Tong, Y., Chen, L.：效用感知的社交活动参与者规划。载于：2015年 ACM SIGMOD 数据管理国际会议论文集，第 1629–1643 页 (2015)[29] Asghari, M., Deng, D., Shahabi, C., Demiryurek, U., Li, Y.：大规模价格感知实时拼车：一种基于拍卖的方法。载于：第24届 ACM SIGSPATIAL 地理信息系统进展国际会议论文集，第 1–10 页 (2016)[30] Tao, Q., Zeng, Y., Zhou, Z., Tong, Y., Chen, L., Xu, K.：实时空间众包中的多工人感知任务规划。载于：数据库系统高级应用国际会议，第 301–317 页 (2018)。Springer[31] Zheng, L., Chen, L., Ye, J.：价格感知拼车中的订单派发。VLDB基金会会刊**11**(8), 853–865 (2018)[32]Huang, Y., Jin, R., Bastani, F., Wang, X.S.：道路网络上具有服务保障的大规模实时拼车。arXiv 预印本 arXiv:1302.6666 (2013)[33]Ma, S., Zheng, Y., Wolfson, O.：T-share：大规模动态出租车拼车服务。载于：2013年 IEEE 第29届数据工程国际会议 (ICDE)，第 410–421 页 (2013)。IEEE[34] Asghari, M., Shahabi, C.：空间众包中的在线任务分配。载于：2017年 IEEE 大数据国际会议 (Big Data)，第 395–404 页 (2017)。IEEE[35] Gao, D., Tong, Y., She, J., Song. . . , T.：空间众包中的 Top-k 团队推荐及其变体。数据科学与工程**2**(2), 136–150 (2017)[36] Song, T., Xu, K., Li, J., Li, Y., Tong, Y.：实时空间众包中的多技能感知任务分配。GeoInformatica (2019)[37] Li, X., Zhao, Y., Guo, J., Zheng, K.：空间众包中具有基于社会影响的偏好的群组任务分配。Springer，??? (2020)[38] Wangze, N., Peng, C., Lei, C., Xuemin, L.：依赖感知空间众包中的任务分配。载于：2020年 IEEE 第36届国际会议

数据工程（ICDE）（2020年）[39]Fudenberg, D., Tirole, J.：《博弈论》。Economica 60(238), 841–846（1992年）[40]Myerson, R.B.：《博弈论：冲突分析》。博弈论：冲突分析，???（1997年）[41] Rasmusen, E.：博弈与信息：博弈论导论。st.ewi.tudelft.nl（2006年）[42] Dov, M., S., S.L., Lloyd, S.：势博弈（1996年）[43] 滴滴出行数据。https://gaia.didichuxing.com 访问日期：2020年10月22日[44] Chen Zhao, R.F. Cheng Peng: https://github.com/gmission/gmission