

基于纳什竞价的空间众包任务定价算法

林韦达 董红斌 赵炳旭

哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001

(3103968942@qq.com)

摘要 任务定价是众包平台解决利润驱动的任务分配、最大化利润的重要步骤。然而关于工人期望的任务定价研究相对较少,现有大多数研究并不考虑工人与任务的动态需求。此外,出于工人隐私和传感器限制,获取完整的工人信息是困难的。为解决上述难题,提出了基于纳什竞价的空间众包任务定价算法。首先通过机器学习算法获取任务的价格范围,然后在价格区间上进行纳什竞价。为了解决动态供需造成的价格大幅波动问题,设计调整机制来稳定任务均价。最后为模拟纳什均衡点,采用了两种不同的梯度递减函数,来搜索匹配数最大的任务定价。分别在 gMission 数据集和合成数据集进行了实验,结果表明所提算法的匹配数量和任务均价分别是 MCMF 算法的 60% 和 1.57 倍,时间花费是 MCMF 算法的 9.6%,验证了所提算法的有效性。

关键词: 纳什均衡;任务定价;工人期望;动态供需;不完整信息

中图法分类号 TP391

Spatial Crowdsourcing Task Pricing Algorithm Based on Nash Bidding

LIN Weida, DONG Hongbin and ZHAO Bingxu

College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Abstract Task pricing is an important step for crowdsourcing platforms to solve profit-driven task allocation and maximize profits. However, there are relatively few studies on task pricing about worker expectations, and most existing studies do not consider the dynamic demands of workers and tasks. Furthermore, obtaining complete worker information is difficult due to worker privacy and sensor limitations. In order to solve the above problems, a pricing algorithm for spatial crowdsourcing tasks based on nash bidding is proposed. The algorithm first obtains the price range of the task through the machine learning algorithm, and then conducts nash bidding on the price range. In order to solve the problem of large price fluctuations caused by dynamic supply and demand, an adjustment mechanism is designed to stabilize the average price of tasks. Finally, in order to simulate the Nash equilibrium point, two different gradient functions are used to search for the task price with the largest number of matches. The proposed algorithm is tested on the gMission data set and the synthetic data set respectively. The results show that the algorithm is 60% and 1.57 times of the MCMF algorithm in terms of the number of matches and the average task price, and the time cost is 9.6% of the MCMF algorithm. Experimental results show the effectiveness of the proposed algorithm.

Keywords Nash equilibrium, Task pricing, Worker expectations, Dynamic supply and demand, Incomplete information

1 引言

移动互联网和共享经济的发展带动了空间众包的繁荣。目前大多数学者的研究集中在带约束的路径规划、工人行为预测等方面^[1-3],为众包领域做出了卓越的贡献。任务定价是空间众包至关重要的步骤,任务定价需要考虑多方面因素,其中工人期望这一因素不可忽视。若任务价格低于工人期望,工人会拒绝接受任务,即使接受任务,工人违约的概率也会大大增加。若任务价格远高于工人期望,平台的利润会相对减少,从而损害平台的利益。对工人期望价格的获取是困难的,根据工人期望给任务合理定价是一个重要问题。

现有大多数空间众包研究中,通常认为工人信息是可知的,进而通过计算工人期望报酬来进行单方面任务分配。但是现实中由于个人隐私以及传感器(GPS)的限制,获取完整的工人/任务信息是困难的。以出租车为例,2016年7月,纽约出租车管理委员会用区域ID替换经纬度敏感信息,给通过行驶距离来估算打车费造成了一定困难。同时,现实中任务和工人受众多不确定因素的影响通常会存在动态供需^[4-5]问题。Farber在2015年对纽约市2009—2013年出租车运营数据的研究^[6]显示,与晴天相比,雨天的出租车供给量明显减少了7.1%,同时相关数据(如图1所示)也表明,恶劣天气、节假日等因素会导致工人/任务数量减少,造成供需不平衡。

到稿日期:2022-09-14 返修日期:2023-02-26

基金项目:国家自然科学基金(61472095);黑龙江省自然科学基金(LH2020F023)

This work was supported by the National Natural Science Foundation of China(61472095) and Natural Science Foundation of Heilongjiang Province, China(LH2020F023).

通信作者:董红斌(donghongbin@hrbeu.edu.cn)

在工人信息缺失和难以预测众多因素的前提下,如何给任务进行合理的定价是空间众包研究中急需解决的重要问题。

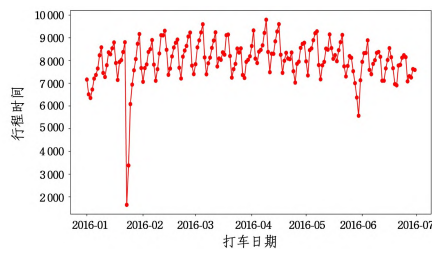


图1 纽约出租车每日累积运行时间

Fig.1 Cumulative daily running time of taxis in New York

综合上述问题,本文考虑任务定价过程中工人期望和市场动态供需两个重要因素,提出了基于纳什竞价的空间众包任务定价算法。首先使用预测、分类算法来获取任务价格范围,然后各任务在价格范围的基础上进行纳什竞价,以逼近工人的期望价格。本文的主要工作如下:

1)针对工人期望难以获取和动态供需难题提出了基于纳什竞价的空间众包任务定价算法。借助博弈论的思想,通过任务之间的竞价来逼近工人的期望。同时为应对不同场景下的动态供需,引入动态调整机制,来保持价格的稳定。在最大化工人与任务的匹配数的同时,最小化任务均价,为众包平台获取最大利益。

2)使用真实数据集 gMission^[7] 和合成数据集来验证算法的效果。相较于 MCMF 算法,本文提出的算法不仅解决了工人期望获取困难的问题,在时间花费上也远少于 MCMF 算法。证明了所提算法的有效性。

2 相关工作

2.1 研究现状

空间任务定价是空间众包至关重要的步骤。空间众包的步骤如下:首先由众包平台对任务进行合理的定价,每一个任务需要一个工人从其位置到达任务位置^[8-9];然后平台将定价后的任务展现给所有工人,工人在得知任务的内容和价格后,根据距离、个人喜好等因素选择任务。Li 等^[10] 基于 E-CAR-GO 模型提出了 K-M 算法,可以高效地匹配司机与乘客。Pan 等^[11] 提出了基于禁忌搜索的时空众包任务分配算法。传统的众包定价算法^[12-14] 并不从工人期望角度考虑,而本文将从工人期望的角度考虑任务定价。

博弈论^[15] 是研究空间众包的重要方法。博弈论主要由 3 个要素构成:参与者、博弈策略和收益。在博弈过程中,参与者之间会产生一种平衡状态,即任何参与者改变自己的策略,只会使得自己的收益受损。上述的平衡状态被称为纳什均衡。Xia 等^[16] 通过引入两阶博弈模型来解决请求者和平台之间的众包困境。Peng 等^[17] 设计了双边竞争博弈框架,工人通过调整报价争取有限的资源,来确定任务价格。Nie 等^[18] 提出了基于两阶段 Stackelberg 博弈的激励模型。Yang 等^[19] 基于斯塔伯格博弈和竞价拍卖理论,分别提出了针对众包平台和用户的两类激励方案。本文引入博弈思想来为任务定价,以逼近工人期望。

2.2 问题描述

定义 1(空间任务) 一个空间任务 t 定义为三元组形式

$\langle l_t(x, y), pr_t, id_t \rangle$ 。其中 $l_t(x, y)$ 表示空间任务的位置信息, x 和 y 表示任务的横纵坐标; pr_t 表示任务的价格区间,成功匹配的任务其价格应该在 pr_t 区间内; id_t 表示任务编号。

定义 2(任务价格区间) 任务价格区间 pr_t 主要通过 3 种方式获得:1)任务请求者直接提供;2)预测算法预测^[20-21];3)分类算法估测^[22]。

本文使用的 gMission 数据集和合成数据集的价格区间分别通过方法 1)和方法 2)获得:在合成数据集中,使用生成函数生成任务的历史数据,然后使用预测算法支持向量回归^[23] (SVR)来预测价格区间。

定义 3(众包工人) 一个工人通过一个四元组 $\langle l_w(x, y), d_w, a_w, t_w \rangle$ 定义。其中 $l_w(x, y)$ 表示工人位置信息,工人需要从 $l_w(x, y)$ 出发达到任务 t 的位置 $l_t(x, y)$ 以完成工作; d_w 表示路费折扣, d_w 和 a_w 是难以获得的参数; a_w 表示工人期望,在路费的基础上波动; t_w 表示工人选择的任务编号,当 $t_w = -1$ 时,工人不选择任务。工人活动范围不受限制。

定义 4(成功匹配) 定义一个成功匹配集合为 M 。匹配集合中的一个匹配 m 定义为一个三元组 $\langle w_i, t_j, p(w_i, t_j) \rangle$, 其中 w_i 表示一个工人, t_j 表示工人 w_i 所匹配的任务; $p(w_i, t_j)$ 表示工人 w_i 完成任务 t_j 后平台支付给工人的报酬,其应满足式(1)。

$$p(w_i, t_j) - d_i(l_i - l_j) - a_i \geq 0 \quad (1)$$

定义 5(匹配问题) 在二维场景中,给定一个工人集合和任务集合,在除了工人选择的任务编号以外,平台没有工人元组其他信息的前提下,需要求得一个匹配集。

本文中的任务场景可以被看成基于匹配的任务分配问题,在工人相关信息未知的情况下使用本文提出的算法。相关符号定义如下:

1)输入数据定义

(1)一个二维平面,作为任务和工人的分布空间,其横纵坐标的范围是 $[0, 100]$; (2)一个空间任务集合 $T = \{t_1, t_2, \dots, t_n\}$; (3)一个工人集合 $W = \{w_1, w_2, \dots, w_n\}$ 。

2)变量定义

(1) m_{ij} : 一个匹配元组 $\langle w_i, t_j, p(w_i, t_j) \rangle$; (2)优化目标:最大化匹配数 $\max M = \min(|W|, |T|)$, 最小化任务均价 $\min \frac{1}{n} \sum_k p_k(w_i, t_j)$ 。

3)限制

(1)对于工人信息,众包平台只知道工人选择的任务编号,其他属性如位置、期望报酬等未知; (2)一个工人只能选择一个任务,一个任务只能由一个工人完成; (3)匹配元组 $\langle w_i, t_j, p(w_i, t_j) \rangle$ 应该满足式(1)。

3 算法部分

算法主要由两部分组成:任务价格区间获取和纳什竞价。算法输入工人集合 W 和任务集合 T ,第一部分获取任务的价格范围 pr_t ,本文直接从请求者获取(gMission 数据集),或者使用支持向量回归来预测(合成数据集)。在获取价格区间后,算法将在价格区间内进行纳什竞价,最后输出匹配集合 M 。

3.1 基于纳什竞价随机梯度算法(NRG)

本文中,为了最大化匹配数,同时最小化任务均价,任务

定价时考虑使用纳什竞价策略来逼近工人群体对任务的期望报酬。在工人选择任务时,任务价格变动的幅度是至关重要的因素。价格幅度变动大,会导致算法不能很好地搜索局部优解,进而导致搜索到的局部解质量较差。价格幅度变动小,算法容易陷入质量较差的局部解。任务价格变动的幅度通过如下公式计算:

$$C_i = \frac{C_i}{W} \max M, (i=1, 2, 3) \quad (2)$$

$$inc(i) = \begin{cases} 3((i-1)^2 + 2) \times \left(\frac{|W|}{|W| + |T|} \right)^{1.1}, & |T| > |W| \\ 3((i-1)^2 + 2) \times \frac{|T|}{|W| + |T|}, & |T| \leq |W| \end{cases} \quad (3)$$

$$\epsilon = \frac{1}{\max M} \sum_{i=0}^3 C_i \times inc(i) \quad (4)$$

首先统计有 $i(i \leq 3)$ 个工人报名的任务,记为 C_i 。然后式(2)将 C_i 规格化到最大匹配数 $\max M$,为使尽可能多的工人参与匹配, C_0 不进行规格化。当多数任务最大报名人数 $\max_w \neq 1$ 时,意味着此刻工人产生了聚集或弃选现象,这说明大多数任务的价格是不合理的。反之大多数任务 $\max_w = 1$ 时,意味着多数任务被一个工人选择,这时大多数任务价格是相对合理的。式(4)考虑了上述情况,对不同值的 \max_w 分配不同的权重 $inc(i)$,其中 C_1 的权重最小, C_3 的权重最大。权重计算函数见式(3)。当工人分布不均匀时 ($\max_w \neq 1$),梯度 ϵ 会变大,价格调整幅度增大;工人分布均匀后 ($\max_w = 1$),式(4)值变小, ϵ 变小,价格变动的幅度变小,价格趋于稳定。

当工人的数量与任务不相等时,会产生任务均价不稳定的问题,进而导致算法的效用降低。该问题可通过式(3)解决,其中 $|W|$ 代表工人的数量, $|T|$ 代表任务的数量。当 $|T| > |W|$ 时,由于工人稀缺,任务之间会有通过涨价吸引工人的行为。这种行为会随着 $|T|$ 变大或者 $|W|$ 减小而加剧,而公式 $|T|/(|W| + |T|)$ 的值会减小,进而抑制价格的上涨。反之,当 $|T| < |W|$ 时,工人的数量增多,这时需要考虑降低任务均价,最大化平台利益。

当任务有工人选择时,任务会降价。降价幅度见式(6)。其中 p_{\max} 表示当前任务的最高价格; $iter$ 表示算法迭代次数; $random()$ 表示随机数,随机的范围为 $[0, 1]$ 。当任务没有工人选择时,会通过涨价来吸引工人匹配。涨价幅度见式(7),其符号表示与式(6)相同。

$$\text{sigmoid}(x) = \frac{1}{1 + e^{3-2x}} \quad (5)$$

$$\Delta p = p_{\max} \times \text{sigmoid}(\max_w) \times \frac{20}{iter} \times \text{random}() \quad (6)$$

$$\Delta p = p_{\max} \times \text{sigmoid}(\epsilon) \times \frac{20}{iter} \times \text{random}() \quad (7)$$

算法的伪代码如算法1—算法3所示,其中伪代码的基本符号如表1所列。

算法1 基于纳什竞价随机梯度算法

输入:工人集合 W 和任务集合 T

输出:工人-任务的匹配集合 M

1. 随机初始化任务的当前价格

2. repeat

3. 统计工人选择任务的数量分布

4. 使用式(2)规格化工人和任务到最大匹配数

5. 式(4)计算下降梯度 ϵ

6. for 对于每一个 $t \in T$ do

7. 调用任务价格更新函数

8. for 对于每一个 $w \in W$ do

9. 调用工人任务更新函数

10. 将当前匹配数与记录的最优匹配数进行比较

11. until 迭代次数 $iter$ 超过预设值

算法2 任务价格更新函数

输入:下降梯度 ϵ 和迭代次数 $iter$

输出:当前任务更新后的价格 p_{now}

1. 初始化梯度 Δp

2. if $\max_w > 0$ then

3. 根据式(6)更新 Δp

4. else

5. 根据式(7)更新 Δp

6. 更新任务价格 p_{now}

7. if $p_{\text{now}} > p_{\max}$

8. $p_{\text{now}} = p_{\max}$

算法3 工人任务更新函数

输入:任务集合 T

输出:当前工人更新后的任务目标

1. 计算当前工人对每个任务的收益 profit

2. 将 profit 从大到小进行排序

3. for 对于每一个 $t \in T$ do

4. if $\text{profit} > 0$ and $\max_w < 3$ and $\text{random}() > \max_w \times p$ then

5. 工人选择任务 t

6. break

7. if 工人没有选择任何任务 then

8. $t_w = -1$

表1 论文中使用的基本符号

Table 1 Basic symbols used in this paper

符号	定义
$ W , T $	工人数量,任务数量
t	一个任务
\max_w	任务最大报名人数
p_{\min}, p_{\max}	任务价格的上界与下界
w	一个工人
M	一个匹配集合
$m \in M$	匹配集合的一个元组
$\max M$	最大匹配数

为了使算法能更好地搜索局部最优解,递减幅度大小考虑使用递减函数。当算法迭代到一定次数时,梯度会无限趋近于0。在纳什竞价过程中寻找最大匹配数的组合并记录下来。选择的递减函数为 $1/iter$,其中 $iter$ 为算法迭代的次数。为了增大算法的搜索空间,可以跳出局部劣解,算法的梯度随机扩大 n 倍 ($0 \leq n \leq 20$),涨价和降价见式(6)、式(7)。

考虑工人选择任务的策略。工人的选择策略遵循如下两点:1)贪婪策略,在满足工人期望的前提下,工人会优先选取获利最多的任务,即式(1)的最大值;2)不聚集策略,任务的 \max_w 值越大,后继工人选择它的概率就越小。

3.2 基于纳什竞价算法(NASH)

不同于NRG算法的递减函数,NASH算法考虑递减

迭代的次数和大小对搜索高质量解产生的影响,NASH 算法的递减函数采用 $\alpha^{iter \times \beta}$ 。其中 α 控制算法的迭代次数, β 控制迭代梯度大小。为了验证规格化对算法的作用,去除算法 2 的第 4 行,同时使用式(8)和式(9)替换算法 2 的式(6)和式(7),修改为 NASH 算法。

$$\Delta p = p_{\max} \times \text{sigmoid}(\max_w) \times \alpha^{iter \times \beta} \quad (8)$$

$$\Delta p = p_{\max} \times \text{sigmoid}(\epsilon) \times \alpha^{iter \times \beta} \quad (9)$$

3.3 基于纳什竞价规格化算法(NN)

在工人和任务数量不对等的场景中,NASH 算法在均价上的表现十分不稳定,在工人稀缺时出现了过多以涨价吸引工人的行为。使用式(8)和式(9)替换式(6)和式(7),修改为 NN 算法。

举例:假设场景有 3 个工人和 3 个任务,其工人和任务的相关属性如表 2 和图 2 所示。

表 2 工人和任务的属性

Table 2 Attributes of workers and tasks

对象	坐标	价格范围	路费折扣	期望波动	目标任务
w_1	(30,55)		0.63515	0	-1
w_2	(98,24)		0.46922	0	-1
w_3	(68,10)		0.51949	0	-1
t_1	(32,70)	[0,41]			
t_2	(68,42)	[0,19]			
t_3	(45,57)	[0,16]			

工人	w_1	w_2	w_3
任务	t_1	-1	-1
价格	20.5	0	0

第一次迭代

任务	t_1	t_2	t_3
\max_w	1	0	0
Δp	0	0	0
价格	20.5	9.5	8

第二次迭代

工人	w_1	w_2	w_3
任务	t_1	t_2	t_2
价格	16.8	19	19

任务	t_1	t_2	t_3
\max_w	1	2	0
Δp	-3.7	10	10
价格	16.8	19	16

工人	w_1	w_2	w_3
任务	t_3	-1	-1
价格	16	0	0

任务	t_1	t_2	t_3
\max_w	0	0	1
Δp	-1.85	-3.65	0
价格	14.95	15.35	16

图 2 算法迭代

Fig. 2 Algorithm iteration

任务初始价格设定为价格范围的中位数,即为 [20.5, 9.5, 8],其他参数见表 2。初始化后只有 t_1 的价格满足 w_1 的期望,而 w_2 和 w_3 由于没有任务的价格满足期望,因此不选择任何任务,此时的匹配只有 $\langle w_1, t_1, 20.5 \rangle$ 。接下来任务开始根据工人选择分布情况使用式(6)或式(7)更新价格。在第一轮价格更新后,此时的匹配有两个: $\langle w_1, t_1, 16.8 \rangle$ 和 $\langle w_2, t_2, 19 \rangle$ 。任务 t_2 出现了聚集现象,所以 t_2 在第二轮价格更新,降价幅度相对变大。由于遵循降低平台成本的思路, t_1 虽然只有一个工人报名,但依然进行小幅度降价。而 t_3 超出了价格范围,价格不变。在算法经过多次迭代后,记录迭代过程中的最大匹配数,作为最终任务定价策略。

3.4 博弈模型的构建和分析

假设场景中所有工人对任务的期望报酬是合理的,且工人和任务的位置分布是均匀的,不存在一个任务的价格低于所有工人期望报酬。首先,将本文任务定价问题转换为 n 个

任务之间的策略博弈 $G = \langle T, ST, U \rangle$,包含任务(玩家)、策略和效用函数。具体定义如下:

1) $T = \{t_1, t_2, \dots, t_n\}$ 是任务集合,将其作为博弈中的玩家。

2) $ST = \{st_1, st_2, \dots, st_n\}$ 是所有玩家策略,即博弈的策略空间。 st_i 表示任务 t_i 的策略集合, st_i 中的策略主要有 3 种:(1)选择此任务的工人数量大于 1,任务做降价调整,策略记为 C;(2)选择此任务的工人数量等于 1,不调整价格,策略记为 N;(3)没有工人选择此任务,任务做涨价调整,策略记为 R。因此可以用 $st_i = \{C, N, R\}$ 表示任务的策略集合。

3) $U = \{U_1, U_2, \dots, U_n\}$ 是所有任务的收益函数, U_i 表示 t_i 的收益函数。收益计算式如下:

$$Y_i = \begin{cases} r_i - p_i \\ 0 \end{cases} \quad (10)$$

$$U_i(st_i) = Y_i \quad (11)$$

其中, r_i 表示任务初始价格, p_i 表示完成任务支付给工人的报酬。若有工人选择 t_i ,其收益为 $r_i - p_i$;若没有工人选择 t_i ,其收益为 0。

接下来证明定价问题存在纯纳什均衡,这表明每个任务会以确定性方式选择调整价格策略。首先需要证明定价问题是一个严格的位势博弈(Exact Potential Game, EPG),EPG 存在一个全局势函数,所有任务的定价策略都可以映射到这个函数上。对于包含有限策略的 EPG,采用最佳响应策略总是会收敛到纯纳什均衡。

严格位势博弈:当一个策略博弈 $G = \langle T, ST, U \rangle$ 存在一个势函数 Φ ,如式(12)所示:

$$U_i(st_i', st_{-i}) - U_i(st_i, st_{-i}) = \Phi(st_i', st_{-i}) - \Phi(st_i, st_{-i}) \quad (12)$$

其中, st_i 和 st_i' 是任务 t_i 的定价策略, st_{-i} 是其他任务的定价策略, Φ 是博弈 G 的严格位势函数。

定理 1 定价问题是一个严格位势博弈。

证明:定义势函数为式(13),它表示所有任务的收益,然后可以得出推导公式式(14):

$$\Phi(st) = \sum_{j \in T} Y_j \quad (13)$$

$$\begin{aligned} & \Phi(st_i', st_{-i}) - \Phi(st_i, st_{-i}) \\ &= (Y_i' + \sum_{j \in T-i} Y_j) - (Y_i + \sum_{j \in T-i} Y_j) \\ &= Y_i' - Y_i \\ &= U_i(st_i', st_{-i}) - U_i(st_i, st_{-i}) \end{aligned} \quad (14)$$

根据 EPG 的定义,定价问题是一个严格位势博弈。

定理 2 定价问题存在纯纳什均衡。

证明:定价问题是一个严格位势博弈。对于包含有限策略的严格位势博弈,采用最佳响应策略总是会收敛到纯纳什均衡^[24]。由于定价问题中任务调整价格策略是有限的,因此定价问题存在纯纳什均衡。

所有的任务定价策略都得到纯纳什均衡,即每个任务都至少有一个工人选择。若 $|T| > |W|$,则没有被选择的任务价格已经最高,无法进行加价策略。此时任何一个任务都无法单方面地更改自己的价格来获得更多收益。

定理 3 纳什定价算法能找到一个更优纳什均衡点。

证明:如式(13)所示,所有的工人收益都可以映射到势函数 Φ ,这表明每个任务的策略调整对其利润和势函数的影响是相同的。在势博弈中,每个任务能依据最佳响应策略依次更新自己的策略,最终势函数会达到一个纳什均衡状态,即局部最优解。文献[25]证明了在任何有限的势博弈中,依据最佳响应依次更新策略总是会收敛到一个纳什均衡状态。

为了跳出局部最优解,将纳什定价算法中的任务策略进行如下调整:保留策略(3),去掉策略(2),修改策略(1)为选择此任务的工人数量大于等于1个,任务做降价调整,记策略为C。考虑算法到达一个纳什均衡点,此时每个任务都至少有一个工人选择,任何一个任务都无法通过单方面更改价格来获得更多利润。但由于去掉了策略(2)并且修改了策略(1),此时所有任务都进行了降价调整,算法会试图搜索更好的纳什均衡点。

4 实验分析

4.1 实验设置

实验使用两个数据集:gMission数据集(记为GM)和合成数据集(记为SYN)。为验证本文算法的有效性,从众包平台gMission获取了6000条记录,并抽取了300条记录进行实验。gMission的数据集格式以元组形式表示为:任务<抵达时间, x 坐标, y 坐标,范围,报酬,截止时间>,工人<抵达时间, x 坐标, y 坐标,范围,服务质量>。由于gMission缺少相应数据,因此对gMission数据做如下更改:去除工人和任务的时间属性和范围属性,工人服务质量转化为路费折扣。

工人的数据格式变为< x 坐标, y 坐标,路费折扣,报酬波动,目标任务>,任务的数据格式< x 坐标, y 坐标,价格区间>,设置任务价格区间为[0,报酬]。为了保持实验的有效性,这里设置报酬波动为0。

合成数据生成的范围:工人的坐标位置均匀地在[0,100]的二维平面上生成,工人的路费折扣系数在区间[0.4,0.8]均匀分布,报酬波动在区间[-10,10]上均匀分布。为防止出现负数报酬,在计算报酬时,若报酬小于0,则记为0.1。任务的坐标位置均匀地在[0,100]的二维平面上生成,任务的历史数据由线性函数生成,生成后添加高斯噪声。

实验的参数设置以及参数范围如表3所列,没有特殊说明的情况下,下划线参数为默认参数。

表3 实验参数

Table 3 Experimental parameters

参数	参数范围
W(SYN)	100,200,300,400,500,600
W(GM)	50,100,150,200,250,300
T(SYN)	100,200,300,400,500,600
T(GM)	50,100,150,200,250,300
α	0.5,1.5,2.5,3.5,4.5,5.5
β	0.965,0.97,0.975,0.98,0.985,0.99
p	0,0.068,0.136,0.204,0.272,0.34

现有的研究大多直接计算工人与任务的欧氏距离,以计算工人期望,但是这并不合理,而且需要获取工人的完整信息,并不符合实验场景。因此本实验对如下几种算法进行比较,实验结果如图3和图4所示。

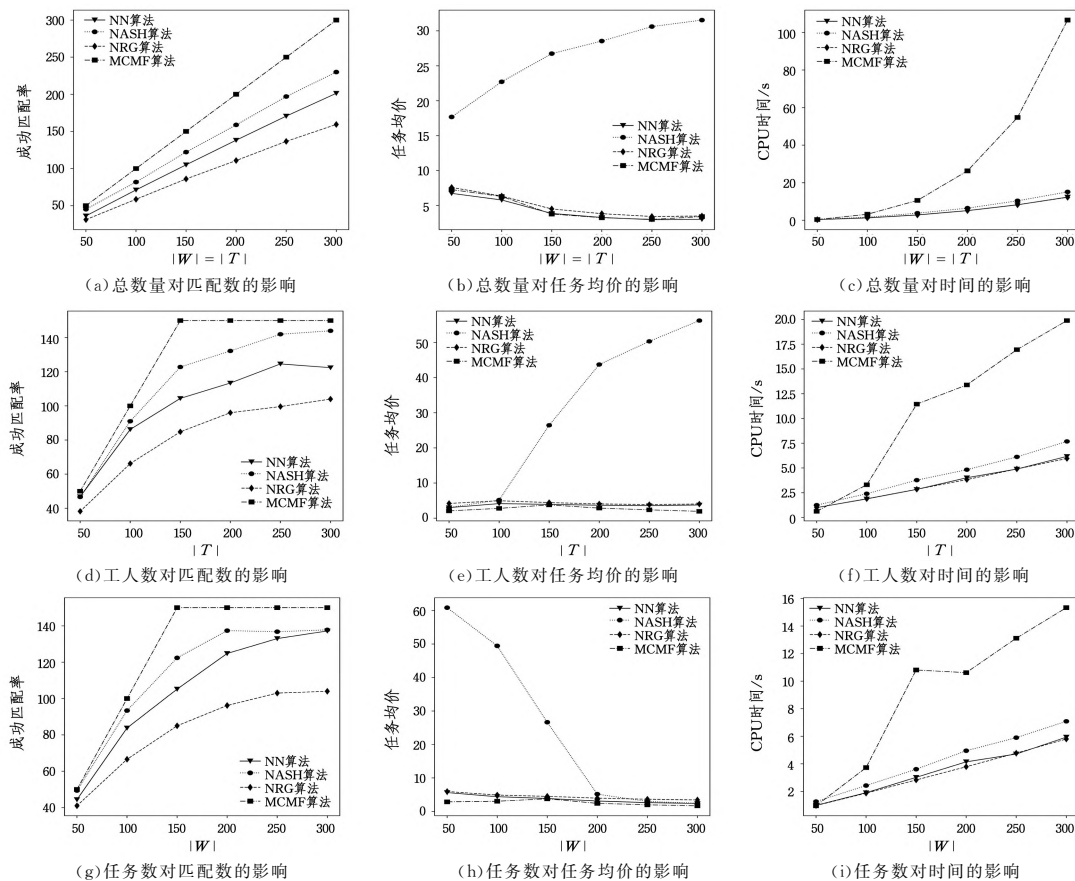


图3 GM数据集的实验结果

Fig. 3 Experimental results on GM data set

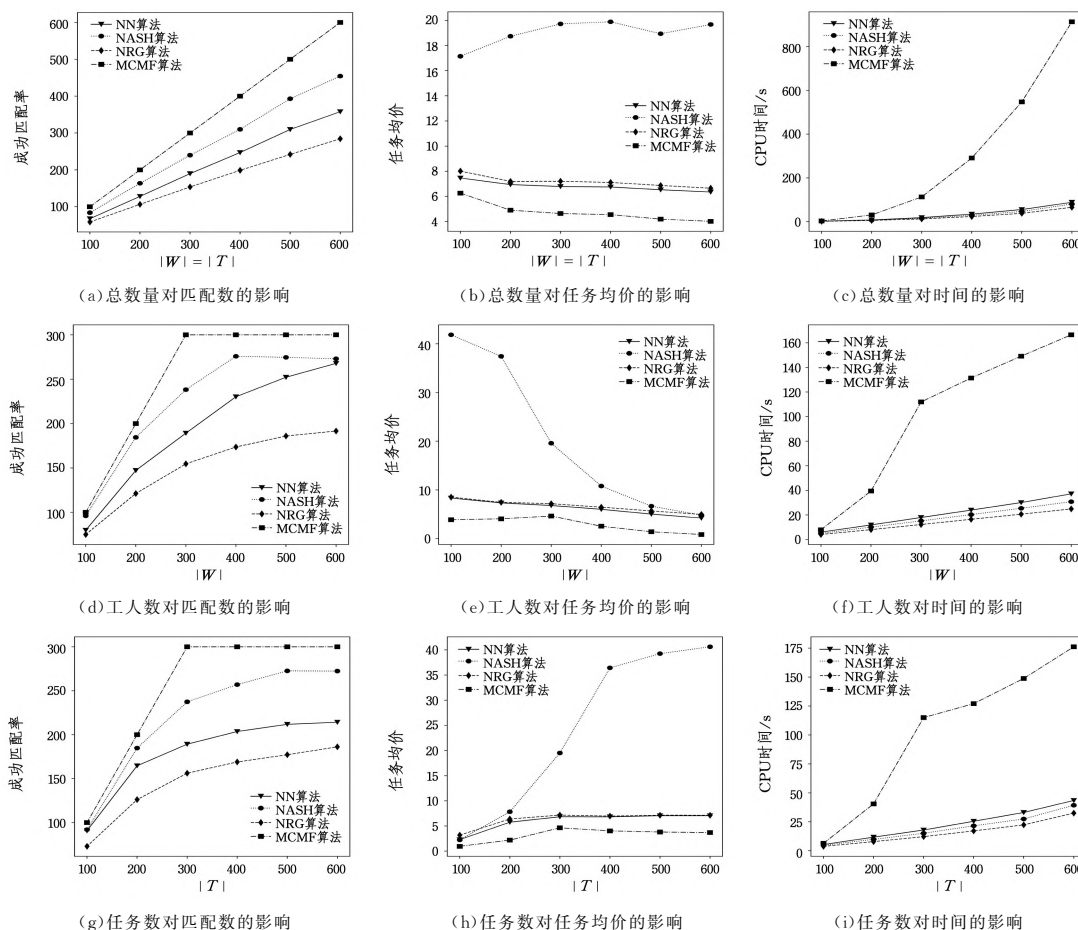


图4 SYN数据集的实验结果

Fig. 4 Experimental results on SYN data set

MCMF算法^[26]:最小费用最大流算法。MCMF算法先建立一个工人-任务的二分匹配图,然后在二分图上反复使用SPFA算法寻找最小费用路进行扩流,最终达到最优匹配。MCMF算法中工人的完整信息和对任务的期望价格对于众包平台是已知的。

NRG算法:基于纳什竞价随机梯度算法。

NASH算法:基于纳什竞价算法。

NN算法:基于纳什竞价规格化算法。

1)总数量对算法效用的影响。设置工人数等于任务数。从图3(a)和图4(a)可以看出MCMF算法效果最好,在两个数据集上达到最大匹配数的同时,匹配的均价也最低。但是随着数量级的线性增长,MCMF算法在时间花费上呈现指数增长,如图4(c)所示,在SYN数据集中,MCMF算法最大的时间花费是其他算法的14倍。NASH算法在匹配上达到了次优,是MCMF算法的75.5%~90%,在时间花费上有着较好的效果,但在任务均价上远远高过其他算法,是MCMF算法的2.4~9.26倍(见图3(b)和图4(b)),任务均价随着数量的增加而增加。这是由于NASH算法没有规格化的缘故,没有规格化可以在解空间进行更大范围的搜索,但是在迭代次数恒定的前提下,梯度过早减少无法搜索到良解,使得任务均价增加。NN算法和NRG算法在匹配数量上分别为第三优和最差,分别是MCMF算法的59.5%~68%和47%~62%。

NN算法在均价上是最优算法的0.92~1.41倍,NRG算法在均价上是最优算法的1.02~1.66倍,两种算法的速度总体最快。其中NRG算法使用了 $1/x$ 递减函数,导致梯度过早减少,没有找到更好的均衡点,相较于其他算法匹配数最少。

2)工人数量 $|W|$ 对算法效用的影响。如图3(d)和图4(d)所示,随着工人的数量增长,MCMF算法在 $|W|=150$ 时达到了最大匹配数。NASH算法匹配数在GM($|W|=200$)和SYN($|W|=400$)中达到一定的数量级后几乎没有明显的增长。而NN算法的匹配数增长几乎不受影响,在工人数是任务数两倍的情况下达到了与NASH相同的匹配数,为91.3%。NRG算法随着工人数的增加增长速度放缓。如图3(e)和图4(e)所示,NASH算法十分不稳定,在 $|W|<|T|$ 时,其均价是其他算法的10倍左右。可以观察到相较于MCMF算法,NN算法和NRG算法的任务均价在GM数据集上相差不大,在SYN数据集上有相对明显的差别,是MCMF算法的2倍左右。可能的原因有两点:(1)数据特征的不同对算法的效果产生了影响;(2)SYN数据集的预测结果有偏差,导致实验效果不一。时间性能与总数量的影响类似。

3)任务数量 $|T|$ 对算法效用的影响。在工人数不变的情况下,4种算法成功匹配的数量都随着任务数量的增长而增长。如图3(g)和图4(g)所示,MCMF算法达到了最优匹配

数, 均价也最低。虽然 NASH 算法任务均价达到了 MCMF 算法的 90.8%, 但是当 $|W| < |T|$ 时, 任务之间产生了竞争的情况, 导致其任务均价在最差情况下是 MCMF 算法的 28 倍, 算法均价受任务数量的影响变得不稳定。NN 算法表现稍好, 无论工人与任务的数量比值如何变化, 其均价远低于 NASH 算法。NRG 算法由于递减梯度过快减少陷入了局部解, 导致算法性能并没有太大变化。

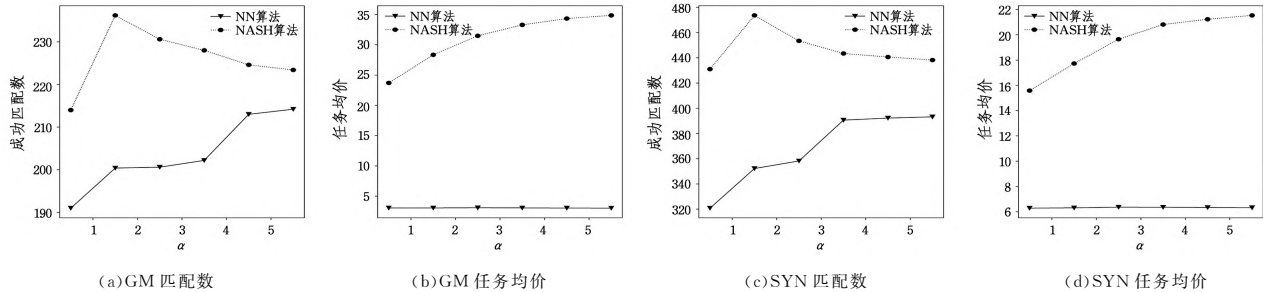


图 5 参数 α 对算法性能的影响

Fig. 5 Effect of parameter α on algorithm performance

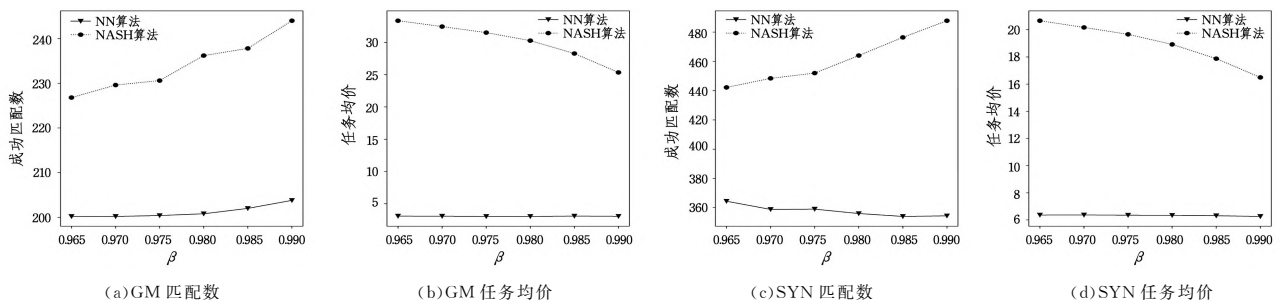


图 6 参数 β 对算法性能的影响

Fig. 6 Effect of parameter β on algorithm performance

α 影响梯度下降速度: α 的值越大, 梯度下降的速度越快。如图 5(a) 和图 5(c) 所示, $\alpha=1.5$ 时, NASH 算法达到了最大匹配数。当 α 继续增大时, 匹配数逐渐减小, 下降幅度为 6%~8%。如图 5(b) 和 5(d) 所示, α 增长的同时任务均价也在上升, 增幅为 28%~33%。原因是当梯度下降过快时, 算法容易陷入较差局部解。NN 算法的匹配数随着 α 的增大而增大, 增长幅度为 11%~19%。均价稳定, 几乎没有太大变化。

β 影响算法的迭代次数: β 越大, 算法迭代次数越多。如图 6(a) 和 6(c) 所示, 随着 β 的增长, NASH 算法的匹配数显著上升。 β 参数对 NN 算法影响较小, 在匹配数上最多只有

3% 的波动。如图 6(b) 和 6(d) 所示, NASH 算法的任务均价下降, 降幅在 20%~25%。结合对参数 α 的分析, 这是因为算法有更多的机会搜索更优的解。NN 算法几乎不受影响。根据图像观察到 NN 算法在两个数据集上表现略有不同, 原因见 $|W|$ 的影响中对任务均价的分析, 因此算法表现有微小的差异。

p 是工人放弃任务的概率。在任务报酬高于期望的前提下, 工人为了提高成功匹配的概率, 会更趋向于选择报名人数少的任务。为了更加直观地表现 p 的影响, 实验分别设置 $|T| > |W|$ 和 $|T| < |W|$ 两个实验组。由于 MCMF 算法没有参数 p , 因此实验只对比 NRG 算法、NASH 算法和 NN 算法。

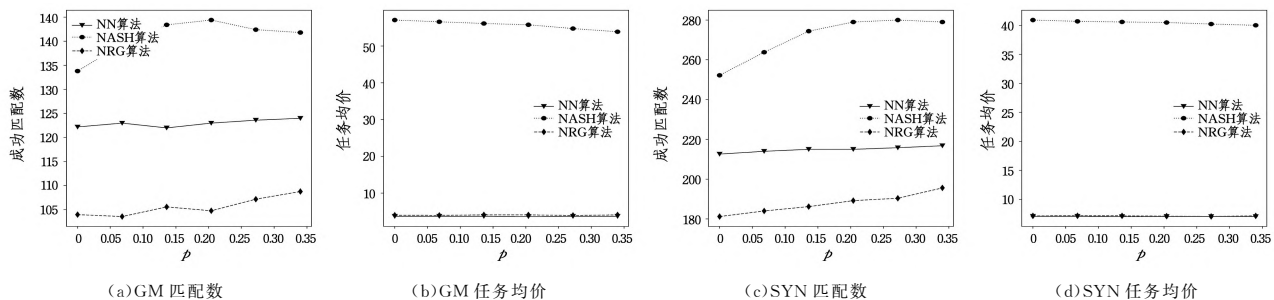
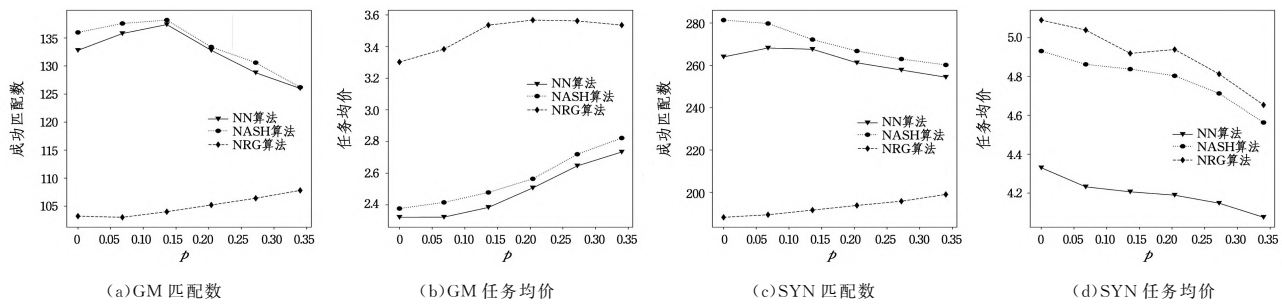


图 7 $|T| > |W|$ 时, 参数 p 对算法性能的影响

Fig. 7 Effect of parameter p on algorithm performance when $|T| > |W|$

图8 $|T| < |W|$ 时,参数 p 对算法性能的影响Fig. 8 Effect of parameter p on algorithm performance when $|T| < |W|$

当 $|T| > |W|$ 时,如图 7(a)和图 7(c)所示,随着 p 变大, NASH 算法的匹配数变多,但超过一定值后,匹配数不变或者略微下降。当 p 较小时,因工人稀缺的缘故,工人有充分的任务可以选择,所以匹配数会有一定程度的上升。 p 变大时,工人的选择会变得十分不稳定,进而影响到梯度的计算,所以匹配数会有一定下降。NN 算法和 NRG 算法随着 p 变大匹配数总体略微增加,NRG 相对于 NN 更为明显。由图 7(b)和图 7(d)可以看出, p 对 3 个算法的任务均价的影响很小,最多只有 6% 的波动(NASH 算法)。

当 $|T| < |W|$ 时,如图 8(a)和图 8(c)所示,随着 p 变大,在 $0 \sim 0.136$ 的范围内,NASH 算法和 NN 算法的匹配数略微上升或者下降。当 $p > 0.136$ 时,NASH 和 NN 算法匹配数都有明显下降,NRG 算法的匹配数在两个数据集上都有增长,幅度在 $4\% \sim 6\%$ 。这是由于 NRG 算法的匹配数远少于其他算法, p 变大使得工人的分布更加均匀。NASH 和 NN 算法的任务均价的变动幅度都不大,在 0.5 之内。由于数据特征不同,随着参数 p 变大,两种算法任务均价在 GM 数据集上略微上升,在 SYN 数据集上略微下降。

综合上述两个实验可以看出参数 p 影响着匹配数。成功匹配数越大, p 的影响就越明显。任务均价受 p 的影响较小。

结束语 本文提出了纳什竞价的算法,从工人期望和动态供需考虑任务定价问题;考虑到工人个人隐私和相关信息收集困难等一系列问题,从工人期望角度阐述任务定价与工人之间的关系,分析了价格对工人选择任务行为和平台利益的影响,在几乎不使用工人信息的前提下,提出了纳什竞价的动态任务定价算法。根据市场供需关系来自动地调节任务价格,为任务制定合理的价格。最后通过实验证明所提出的定价算法在匹配数和任务均价方面有着良好的表现。未来将引入工人联盟的定价策略,考虑使用更拟真的数据集,同时进一步提高算法性能。

参考文献

- [1] LI Y, XU W, YIU M L. Client-Side Service for Recommending Rewarding Routes to Mobile Crowdsourcing Workers[J]. IEEE Transactions on Services Computing, 2021, 14(6): 1995-2010.
- [2] TONG Y, WANG L, ZHOU Z, et al. Flexible online task assignment in real-time spatial data[J]. Proceedings of the VLDB Endowment, 2017, 10(11): 1334-1345.
- [3] ZHAO Y, ZHENG K, CUI Y, et al. Predictive Task Assignment in Spatial Crowdsourcing: A Data-driven Approach[C]// 2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020.
- [4] ZHOU Z, CHEN R, WANG C, et al. Dynamic pricing in profit-driven task assignment: a domain-of-influence based approach[J]. International Journal of Machine Learning and Cybernetics, 2021, 12(2021): 1015-1030.
- [5] GUO B, YAN L, WU W, et al. ActiveCrowd: A Framework for Optimized Multi-Task Allocation in Mobile Crowdsensing Systems[J]. IEEE Transactions on Human-Machine Systems, 2017, (3): 392-403.
- [6] FARBER H S. Why you can't find a taxi in the rain and other labor supply lessons from cab drivers[J]. The Quarterly Journal of Economics, 2015, 130(4): 1975-2026.
- [7] CHEN Z, FU R, ZHAO Z, et al. gMission: A general spatial crowdsourcing platform[J]. Proceedings of the VLDB Endowment, 2014, 7(13): 1629-1632.
- [8] TONG Y X, CHEN L, SHAHABI C. Spatial crowdsourcing: challenges, techniques, and applications[C]// Proceedings of the VLDB Endowment, 2017: 1988-1991.
- [9] TONG Y, CHEN L, SHAHABI C. Spatial crowdsourcing: challenges, techniques, and applications[C]// Very Large Data Bases. VLDB Endowment, 2017.
- [10] LI X H, DONG H B. Matching modeling and optimization method of ride-sharing trip based on E-CARGO model[J]. Computer Applications, 2022, 42(3): 778-782.
- [11] PAN Q X, YIN Z X, DONG H B, et al. Task Allocation Algorithm for Space-Time Crowdsourcing Based on Tabu Search[J]. Journal of Intelligent Systems, 2020, 15(6): 1040-1048.
- [12] GUAN S. Analysis of Optimal Pricing Model of Crowdsourcing Platform Based on Cluster and Proportional Sharing[C]// 2018 6th International Symposium on Computational and Business Intelligence(ISCBI). 2018.
- [13] WANG H, NGUYEN D N, HOANG D T, et al. Real-Time Crowdsourcing Incentive for Radio Environment Maps: A Dynamic Pricing Approach[C]// 2018 IEEE Global Communications Conference(GLOBECOM). IEEE, 2019.
- [14] TONG Y, WANG L, ZHOU Z, et al. Dynamic pricing in spatial crowdsourcing: A matching-based approach[C]// Proceedings

- ACM SIGMOD International Conference on Management of Data, 2018;773-788.
- [15] 孙雪峰,张成堂,朱林.考虑企业社会责任的双渠道闭环供应链定价决策研究[J].重庆工商大学学报(自然科学版),2022,39(4):51-59.
- [16] XIA H,ZHANG R,CHENG X,et al. Two-Stage Game Design of Payoff Decision-Making Scheme for Crowdsourcing Dilemmas [J]. IEEE/ACM Transactions on Networking, 2020, 28(6): 2741-2754.
- [17] PENG J,ZHU Y,SHU W,et al. When data contributors meet multiple crowdsourcers: Bilateral competition in mobile crowdsourcing[J]. Computer Networks,2016,95(11):1-14.
- [18] NIE J,LUO J,XIONG Z,et al. A Stackelberg Game Approach Toward Socially-Aware Incentive Mechanisms for Mobile Crowdsensing[J]. IEEE Transactions on Wireless Communications,2019,18(1):724-738.
- [19] YANG D,XUE G,XI F,et al. Incentive Mechanisms for Crowdsensing: Crowdsourcing With Smartphones [J]. IEEE/ACM Transactions on Networking,2016,24(3):1732-1744.
- [20] WANG B,LIU P,ZHANG C,et al. Research on Hybrid Model of Garlic Short-term Price Forecasting based on Big Data[J]. Computers,Materials and Continua,2018,57(2):283-296.
- [21] BONDE G,KHALED R. Stock price prediction using genetic algorithms and evolution strategies[C]//International Conference on Genetic and Evolutionary Methods, 2012.
- [22] KARGER D R,OH S,SHAH D. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems[J]. Operations Research,2014,62(1):1-24.
- [23] SCHUMAKER R P. Machine learning the harness track: Crowdsourcing and varying race history [J]. Decision Support Systems,2013,54(3):1370-1379.
- [24] MONDERER D,SHAPLEY L S. Potential games [J]. Games and economic behavior,1996,14(1):124-143.
- [25] NISAN N,ROUGHGARDEN T,TARDOS E,et al. Algorithmic Game Theory[J/OL]. <https://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=DE1D742A2A81DF238CF38C3F8125ED88?doi=10.1.1.720.4143&rep=rep1&type=pdf>.
- [26] CHEN L,KYNG R,LIU Y P,et al. Maximum Flow and Minimum-Cost Flow in Almost-Linear Time[J]. arXiv:2203.00671, 2022.



LIN Weida, born in 1998, postgraduate. His main research interests include spatial and crowdsourcing and intelligent computing.



DONG Hongbin, born in 1963, Ph. D, professor. His main research interests include computational intelligence, machine learning and data mining.

(责任编辑:何杨)