

融合 LLM 认知建模的多智能体双层 仿真框架下骑手议价博弈力学研 究

赛道: 3

队名: Cooking Cookies Mice

小组成员: 吴林辉、陈俊佐、任毅、彭俊威、陈哲

导师: 陈本皓、曹忠

日期: 2025 年 6 月 30 日

一、背景介绍

平台经济的崛起，以外卖平台为典型代表，创造了数以百万计的“新就业形态”岗位，深刻地重塑了当代都市的劳动景观与服务模式(Vallas & Schor, 2020)。然而，外卖骑手这一庞大群体，其劳动过程深受算法的精密控制与支配(Wood et al., 2019)。高强度的劳动、不稳定的收入预期、社会保障的缺失以及无处不在的数字化监控，构成了他们日常工作的底色，引发了学界与公众对其劳动权益与福祉的广泛关注(Rosenblat, 2018; 陈龙, 2020)。

在传统的劳资关系中，议价权 (Bargaining Power) 是决定劳动报酬和工作条件分配的核心要素，通常指劳动者或其代表（如工会）与雇主在协商过程中影响对方决策的能力(Flanders, 1968)。然而，在以个体化、灵活用工为特征的平台经济中，传统工会的作用受到限制，骑手往往以“原子化”的个体身份直接面对拥有数据、算法和市场规则制定权的强大平台。这种结构性的不平等导致了骑手议价能力的显著削弱(常凯 & 郑小静, 2019)。

尽管如此，劳动者并非完全被动的接受者。斯科特(Scott, 1985)的“弱者的武器”理论为我们理解非正式抵抗提供了经典框架。大量质性研究发现，骑手会通过“摸鱼”、“磨洋工”、利用系统漏洞的“逆算法”实践以及通过社交网络共享信息等方式，展现出在结构限制下的能动性(Sun & Chen, 2021; 罗斯琦 & 陈佳慧, 2024)。这些看似微观的个体行为，在特定条件下可能汇聚成一种集体的市场力量。

基于此，本研究引入并关注一种“非正式议价权”。我们将其定义为：在缺乏正式组织（如工会）的情况下，骑手通过个体或小范围协调的策略性行为（如选择性接单、集体性下线），利用市场供需关系的变化，对自身劳动报酬和工作条件施加影响的能力。这种议价权并非通过明确的谈判达成，而是通过市场行为和

与平台算法的隐性博弈而实现。理解这种非正式议价权的形成机制、影响因素及其动态演化过程，对于揭示数字时代劳动过程的复杂性、探讨算法治理下的劳动者权益保障路径具有至关重要的理论与现实意义。

二、研究问题

传统研究方法，如质性访谈虽能深刻揭示骑手的能动性，却难以量化其集体效应；而计量经济学模型则往往简化了人的复杂决策过程。面对数字劳动这一复杂适应系统，既有研究尚无法系统性地回答以下核心问题：

在数字平台算法控制和激励机制下，零工骑手的个体决策如何汇聚成群体行为，从而影响其与平台间的议价能力？特定平台策略或市场环境如何改变这种议价能力？

为了深入探究这一核心议题，本研究将其分解为以下三个具体问题：

涌现机制问题： 在算法主导的平台环境中，外卖骑手个体的理性决策（如接单策略、工作时长选择）如何通过相互作用，在群体层面涌现出非正式的议价权？

关键因素问题： 哪些平台算法的关键参数（如动态定价机制、奖惩规则的严苛程度）和骑手间的社会网络特征（如信息共享的效率与广度）对此议价权的形成与效果产生显著影响？

情景适应性问题： 在不同的外部情境（如天气突变、订单需求峰谷波动、突发性运力短缺）以及政策干预下，外卖骑手的非正式议价权会呈现怎样的动态变化与适应性表现？

通过对这些问题的探究，本研究期望不仅能深化对数字时代劳动过程复杂性的理解，揭示算法治理下劳动者能动性的新形式，更能为优化平台治理策略、提升骑手群体福祉、构建更具韧性的数字劳动生态提供有益的理论洞见与实践启示。

三、研究方法

为系统性地探究骑手集体议价的动力学过程，本研究采用了多智能体建模

(Agent-Based Modeling, ABM) 这一“自下而上”的计算社会科学方法。ABM 通过模拟微观个体 (Agent) 的行为及其相互作用, 来观察宏观系统现象的涌现, 特别适用于研究像平台经济这样的复杂适应系统(Gilbert, 2008; Epstein & Axtell, 1996)。

本研究的核心方法论创新在于构建了一个融合大语言模型 (LLM) 认知建模的多智能体双层仿真框架。此方法旨在克服传统 ABM 中 Agent 认知能力不足的缺陷, 从而更真实地模拟人类在复杂社会经济环境中的决策。该框架由两个协同工作的层面构成:

计算层 (Computational Layer): 此层面负责处理大规模、高频率、基于规则的运算。它依据预设的参数和规则, 高效执行骑手移动、订单计时、基础收益结算等确定性行为。这确保了仿真的宏观效率和基础物理世界的运行逻辑。

认知层 (Cognitive Layer): 这是本研究方法的核心创新。该层面嵌入了大型语言模型 (LLM), 在关键决策节点被触发, 用于模拟人类的复杂认知过程。具体而言, LLM 被用于:

综合满意度评估: 模拟骑手在一天工作后, 综合考虑收入、时长、天气、奖惩等因素, 形成一个带有情感色彩和主观判断的满意度分数。

复杂订单与风险权衡: 模拟骑手面对高收益但高风险 (如恶劣天气、远距离) 订单时的风险偏好和决策过程。

社交信息解读与策略响应: 模拟骑手在接收到来自社交网络的非结构化信息 (如“集体下线”号召) 时, 如何理解信息、形成集体认同并调整自身行为策略。

通过此双层架构, 模型中的骑手 Agent 不再是简单的“计算代理”, 而升级为能够进行复杂推理和策略学习的“认知代理” (Cognitive Agent)。这使得我们的研究能够捕捉到传统模型难以刻画的、由社会心理因素驱动的行为模式, 极大地提升了模型的解释力与现实贴近性(Park et al., 2023)。

四、与 AgentSociety 赛事平台的结合

本研究的仿真实验是在清华大学举办的 AgentSociety 赛事平台上设计、实现并运行的。该平台为构建大规模、社会化的多智能体系统提供了强大的技术框架和标准化的开发环境，使得本研究的复杂构想得以落地。我们将研究模型命名为 TriadDelivery，它深度结合了 AgentSociety 平台的核心特性。

(一) 系统与代理架构：

TriadDelivery 系统严格遵循 AgentSociety 平台的代理体系规范。我们构建了一个包含生产者代理 (ProducerAgent)、配送代理 (DeliveryAgent, 即骑手) 和客户代理 (CustomerAgent) 的完整生态。所有代理类型均继承自平台提供的 SocietyAgent 基类，并在此基础上构建了具备经济交互能力的 OrderAgent 中间层，最终形成三级继承的代理架构体系。这种设计确保了代理间的统一接口和标准化交互协议，为复杂的跨代理通信和协作提供了坚实基础。

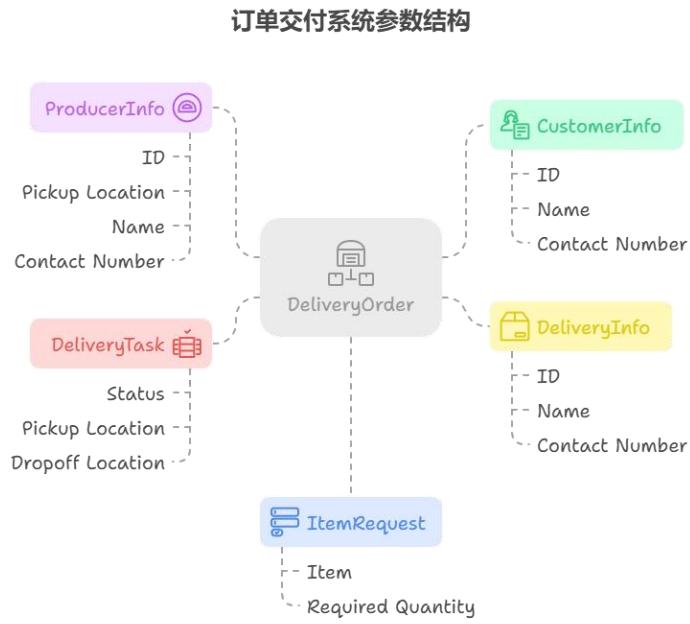


图 1 TriadDelivery 系统架构图

(二) 功能块与通信机制:

我们充分利用了 AgentSociety 平台的功能块 (Block) 化设计理念。例如, 生产者代理内部集成了库存管理块 (InventoryManagementBlock) 和订单分派块 (OrderDispatchBlock), 实现了功能的模块化和高内聚。跨代理的通信则采用平台约定的标准化消息格式, 如以 `order@` 为前缀的订单消息和以 `status_update@` 为前缀的状态更新消息。这种规范化的设计大大简化了复杂交互逻辑的实现, 保证了系统的稳定性和可扩展性。

(三) LLM 的集成与调用:

AgentSociety 平台对 LLM 的集成提供了便利的支持。在本研究的认知层, 我们通过平台接口在关键决策点 (如骑手决定是否接受配送任务) 调用 LLM。LLM 接收到包含骑手人设 (如“求生型”)、订单特征和当前环境状态的结构

化提示 (Prompt) , 并生成符合逻辑的决策及其理由。这种与平台框架的无缝结合, 是实现“认知代理”的关键技术保障。

五、外卖系统参数设定

本模型是一个双层混合智能仿真模型, 旨在模拟外卖骑手在平台经济中的决策行为, 特别是其非正式议价行为的萌芽与演化。

计算层(Computational Layer): 基于参数和规则, 高效处理大规模、高频率的仿真运算 (如移动、计时、基础收益计算) 。

认知层(Cognitive Layer): 引入大语言模型 (LLM) , 在关键决策点模拟人类骑手的复杂认知、情感判断和策略选择。

(一) 骑手主体(Rider Agent)

骑手 Agent 是模型的核心, 其初始化和行为逻辑深度融合了真实世界的调查数据。

1. A.参数化属性(Parameterized Attributes)

在仿真开始时, 通过 AgentSociety 模块根据以下数据分布进行初始化。

参数(Parameter)	类型	数据源与初始化方法
rider_id	Integer	唯一标识符。
Activated	Boolean	骑手不可能 24 小时待机, 故每个骑手都有上下线状态, 用于: 1、每天在工作时长达到后进入下线状态; 2、在暴雨或者休假天气可能下线;

		<p>3、在平台加大单价补助后从下线状态激活。</p> <p>4、集体行动中进入大批量下线状态激活平台补助。</p> <p>这时候需要使用该参数让骑手处于上线或下线状态。</p>
experience_level	Float(1–10)	<p>经验值，影响送餐速度。可以设定成功下单速度自动提升 0.1%，最高提升 5%。 初始设定为： (5 经验值送货速度为 25km/hour, 以此类推)</p> <p>“老骑手”：经验值从正态分布 $N(6.0, 1.5)$ 抽取；（占总数 70%）</p> <p>新骑手：经验值从均匀分布 $U(1.0, 3.0)$ 抽取。（占总数 30%）</p>
base_hourly_income_target	Float	<p>数据源：“月收入”与“日工作时长”分布。初始化：</p> <ol style="list-style-type: none"> 根据月收入分布为 Agent 设定 monthly_income_target。 根据日工作时长分布设定 typical_daily_hours。 计算： $target = monthly_target / (typical_daily_hours * 22)$ (按 22 个工作日算)。 <p>数据</p>

		<p>源 : "MonthlyIncome(forimpoverishedriders, 2019)" 和 "DailyWorkingHours"。</p> <p>模型应用 : 这是一个可以推算出的关键参数。</p> <p>首先, 根据报告中的月收入分布, 为每个 Agent 设定一个 monthly_income_target。</p> <ul style="list-style-type: none"> 9.8% 的 Agent 目标 \leq 1000 元 23.1% 的 Agent 目标在 1000–3000 元 30.2% 的 Agent 目标在 3000–5000 元 29.2% 的 Agent 目标在 5000–8000 元 7.7% 的 Agent 目标 $>$ 8000 元 <p>然后, 根据报告中的工作时长分布, 为每个 Agent 设定一个 typical_daily_hours。</p> <p>算 : $\text{base_hourly_income_target} = \text{monthly_income_target} / (\text{typical_daily_hours} \times 30 \text{ 天})$</p>
max_daily_working_hours	Integer	<p>数据源 : "Daily Working Hours(Delivery Time)" 的详细分布图。</p> <p>模型应用 : 直接使用该分布来初始化每个 Agent 的此项参数。</p> <ul style="list-style-type: none"> ~7% 的 Agent, max_daily_working_hours=1 ~19% 的 Agent, max_daily_working_hours=2 ~16% 的 Agent, max_daily_working_hours=3 <p>Overall(2019) : 58.8% work less than 4hours per day.</p>

		<p>具体分布如下</p> <p>0—1hour : ~7%</p> <p>1—2hours : ~19%</p> <p>2—3hours : ~16.8%</p> <p>3—4hours : ~16%</p> <p>4—5hours : ~12.2%</p> <p>5—6hours : ~9%</p> <p>6—7hours : ~7%</p> <p>7—8hours : ~5%</p> <p>8—9hours : ~3%</p> <p>9—10hours : ~2%</p> <p>10—11hours : ~1%</p> <p>11—12hours : <1%</p> <p>12+hours : <1%</p> <p>这反映骑手群体中，大部分人追求“灵活工作”（工作时长短），少数人是“全职核心”（工作时长长）的现实。</p>
current_shift_hours	ArrayOfRangeObjects	<p>根据 max_daily_working_hours 衍生的上下线时间</p> <p>此处设定从 6 点到晚 20 点下线，若 0.5 小时没有订单自动下线，在规定时间又重新上线。</p> <p>上线时间，早上 6—7 点</p>

		<p>中午 10—12 点</p> <p>晚上 17—20 点</p> <p>直到最大工作时长耗尽为止。</p>
agent_archetype	Enum	<p>数据源：“前职业”与“加入原因”(TGI 指数)。</p> <p>初始化：创建不同的骑手原型：</p> <ul style="list-style-type: none"> ① 求生型(Survival-Type)：前工厂/餐饮工人，因失业加入，目标是稳定收入。risk_preference 较低。 ② 机会型(Opportunity-Type)：前销售/个体户，因“闲着”加入，追求灵活和高回报。risk_preference 较高。 ③ 灵活型(Flexibility-Type)：以“时间自由”为首要动机（64%），对超时工作极其敏感。
risk_preference	String	和 agent_archetype 对应的风险偏好：低，高，敏感（因为用 LLM 进行判断，需要明确信息）
current_satisfaction_score	Integer	<p>满意度，用于判断是否上下线，挑单等行为。</p> <p>提高期望/暂时减少供给（参数 1—20），低于 6 触发挑单，低于 3 触发下线；</p> <p>当满意度过低时，骑手可能通过“挑单”（提高接单标准）或暂时下线来表达不满或等</p>

		待更好机会。
积分与星级管理系统		平台用于考核骑手的积分系统
monthly_base_points	Integer	每月基础分。固定值，如 150(100 基础分 +50 培训分)。
current_points_score	Float	当前实时积分。每月初重置为 monthly_base_points。
current_star_level	Integer(1-6)	当前星级。根据 current_points_score 由平台动态计算得出。
point_sensitivity	Float(0.5-2.0)	积分敏感度。这是个体的心理学属性。agent_archetype 为“求生型”或有稳定收入需求的骑手，此值更高。高星级的骑手此值也会动态增高。

2. B.核心决策逻辑(CoreDecisionLogic)

(1) 基于规则的决策(Rule-BasedDecisions)-计算层

上线/下线: 基于 current_satisfaction_score、current_shift_hours、阈值的简单比较。

优先程度: 阈值>current_shift_hours>current_satisfaction_score.

基础订单评估:

计算预期净收益(ENE)=offered_price-base_hourly_income_target X 路程 (distance_km) /速度 (experience_level) 。

time_cost 与骑手的 base_hourly_income_target 挂钩。IF ENE>0 THEN 初

步考虑接受。

配送行为:接受订单后，在地图上移动，更新位置、收入和工时。

(2) 基于 LLM 的决策(LLM-DrivenDecisions)-认知层

综合满意度评估(HolisticSatisfactionAssessment):

触发: 每日工作结束或固定时间间隔。

LLM 输入:{agent_archetype}人设+一天的工作总结(收入、时长、天气、奖惩、抢单次数等)。

LLM 任务:“你是一个最看重{时间自由}的{机会型}（这部分看 agent_archetype 上面类型设定）骑手，根据你今天{...}的工作情况，生成你当前的满意度分数（1-10 分）并说明理由。”

输出:更新 current_satisfaction_score。

(3) 复杂订单决策与风险权衡 (ComplexOrder&RiskTrade-off):

触发:当面临高不确定性订单时（如恶劣天气、远距离高价单）。

LLM 输入:{agent_archetype}人设+订单特征+risk_preference。

LLM 任务:“你是一个风险偏好为{低}的{求生型}骑手，现在有一个暴雨天的高价单，虽然比平时多赚 20 元，但有超时罚款的风险。你接还是不接？”

输出:接受/拒绝决策，可以覆盖简单的 ENE 规则。

(4) 社交信息解读与策略响应(SocialInfo&StrategyResponse):

触发:接收到来自模拟社交网络的非结构化信息。

LLM 输入:{agent_archetype} 人设+social_network_degree+信息文本 (如: “兄弟们 8 点后集体下线等涨价”、“XX 商场保安找茬”)。

LLM 任务:“你的社交很广, 在老乡群里看到‘集体下线等涨价’的号召。你当前的满意度不高, 你会如何行动? ”

输出:调整 current_strategy (如切换到 SEEKING_HIGH_PRICE 或 TEMPORARY_OFFLINE), 实现个体或群体的“软抵制”。

(二) 平台/环境主体(Platform/EnvironmentAgent)

1. A. 核心机制(CoreMechanisms)

(1) 接收订单(ReceivingOrders):

平台现在有一个“订单池”(OrderPool), 不断接收由所有 Customer_Agent 生成并提交的 Order_Object。

(2) 动态定价(DynamicPricing)-反馈循环更真实:

输入:平台的“订单池”大小 (代表实时需求) 和“在线空闲骑手”数量 (代表实时供给)。

核心逻辑:供需比计算(在线空闲骑手数)/(订单池中的订单数)依然是核心。

新的反馈循环:

骑 手 不 足 -> 供 需 比 低 -> 平 台 提 高

dynamic_pricing_multiplier.

dynamic_pricing_multiplier 升高-> 用户 Agent 的“下单意愿”降低->新生成的订单减少。

同时，高价吸引更多骑手上线/接受订单。

最终，市场通过价格机制自动寻找新的平衡点。

订单定价(OrderPricing):

当一个订单进入“订单池”后，平台为其计算 offered_price.

$$\text{offered_price} = (\text{base_order_price_per_km} \times \text{distance_km}) \times \text{dynamic_pricing_multiplier}.$$

定价后，订单状态变为 PENDING_ASSIGNMENT，并向骑手推送。

2. B. 积分与星级管理系统(Points&Star-Level System)

(1) 积分更新规则(PointsUpdateRules):

订单完成:rider.current_points_score+=0.1

收到好评:rider.current_points_score+=0.5

轻微超时:rider.current_points_score-=2

普通超时:rider.current_points_score-=8

严重超时:rider.current_points_score-=18

差评/投诉(首次):rider.current_points_score-=5

差评/投诉(再次):rider.current_points_score-=20

(2) 星级与奖金计算规则(Star-Level&BonusCalculation):

平台在每个时间步或每日结算时，根据每个骑手的 current_points_score 更新其 current_star_level。

星级划分(示例):

Score<160: 1 星(每单奖金 0.05 元)

160<=Score<190: 2 星(每单奖金 0.09 元)

190<=Score<220: 3 星(每单奖金 0.14 元)

220<=Score<250: 4 星(每单奖金 0.22 元)

250<=Score<320: 5 星(每单奖金 0.35 元)

Score>=320: 6 星(每单奖金 0.70 元)

奖金计算：骑手完成订单后，其实际收入变为 offered_price+star_level_bonus。

(三) 用户 Agent(Customer_Agent)

用户是需求的源头，他们的集体行为决定了市场的订单总量和分布。

1. 参数化属性(ParameterizedAttributes)

参数(Parameter)	类型	描述与初始化方法
customer_id	Integer	唯一标识符。
location	Tuple(x,y)	用户（住宅/办公）位置。在地图上根据人口密度分布进行初始化。
price_sensitivity	Float(0.1-2.0)	价格敏感度。1.0 为标准，>1.0 表示对高价更敏感，<1.0 表示对高价容忍度更高。可从正态分布 N(1.0,0.3) 抽取。
order_propensity_baselin	Float(0-1)	基础点餐倾向。代表该用户在“正常”情况下（非高峰、

e		价格正常) 点外卖的基准概率。
time_preference_factor	Dict	时间偏好系数。定义用户在不同时间段的点餐倾向。例如: {'lunch_peak':1.8,'dinner_peak':2.0,'late_night':0.5,'off_peak':0.2}。
last_order_time	Integer	上次下单的时间步, 用于模拟“短时间内不重复下单”的行为。
tolerance_level	Float(0.5-1.5)	个人容忍度/挑剔度。1.0 为标准, <1.0 表示更挑剔, >1.0 表示更宽容。可从正态分布 N(1.0, 0.2)抽取。
expected_delivery_time	Integer	期望送达时间。用户下单时, 平台根据距离和当时路况预估并显示给用户的時間。 expected_delivery_time=platform.estimate_delivery_time(order). B.核心决策逻辑(CoreDecisionLogic)

(1) 下单决策(OrderDecision)-每个时间步触发

逻辑: 用户在每个时间步都会根据当前情况计算一个“下单意愿分”(OrderWillingnessScore)。

计算公式(简化版):

Score=order_propensity_baselinetime_preference_factor[current_time_of_day]/(dynamic_pricing_multiplierprice_sensitivity)

决策规则:

IF(current_time-last_order_time)<min_reorder_interval THEN 不下单。

生成一个随机数 R(0-1)。

IF R<Score THEN 决定下单。

生成订单(OrderGeneration)

触发:当用户决定下单后。

逻辑:

创建一个 Order_Object 实例。

order.delivery_location=用户的 location。

order.pickup_location=从用户附近（例如，一定半径内）的虚拟“餐厅”中随机选择一个。

order.creation_time_step=current_time_step。

将此 Order_Object 提交给平台 **Agent** 进行定价和分配。

(2) 评价决策模块

这个模块将在骑手完成配送，订单状态变为 DELIVERED 后被触发。

A. 影响用户评价的关键因素

用户的评价主要基于其“期望”与“实际体验”之间的差距。关键因素包括：

配送时效(DeliveryTime):这是最重要的因素。

骑手服务态度(Rider'sAttitude):在模型中可以简化或用随机事件代表。

餐品完好度(FoodCondition):同上，可简化。

用户个人“容忍度”(Customer'sToleranceLevel):有些用户天生就更宽容，有些则更挑剔。

C. 评价决策逻辑(RatingDecisionLogic)

当一个订单配送完成后，触发对应用户的评价决策。

(1) 基于规则的决策(Rule-BasedDecision)

这部分可以处理大部分常规情况，确保模型的计算效率。

输入：

actual_delivery_time: 骑手实际送达的时间步。

order_creation_time: 订单创建的时间步。

expected_delivery_time: 平台预估的送达时间。

用户的 tolerance_level。

核心计算：计算“体验偏差分”(**ExperienceDeviationScore**)

time_taken=actual_delivery_time-order_creation_time

lateness_factor=time_taken/expected_delivery_time(例如, 1.0 代表准时, 1.2 代表超时 20%, 0.8 代表提前 20%)

deviation_score=lateness_factor(1/tolerance_level)

评价规则：

IF deviation_score<=1.05: 给予好评(**PositiveRating**)。 (即在容忍度范围内, 甚至稍微超时一点也可以接受)

ELSEIF 1.05<deviation_score<=1.3: 不评价 (**NoRating**) 或 中评 (**NeutralRating**)。 (明显超时, 但不至于给差评)

ELSEIF deviation_score>1.3: 给予差评(**NegativeRating**)。 (严重超时, 超出容忍极限)

随机事件：

为了模拟其他因素 (如餐品洒漏、骑手态度问题), 可以加入一个很小的随机概率 (如 1%) 直接导致差评, 无论是否超时。

(2) 差评传递

用户 Customer_Agent: “我根据我的体验, 决定要对骑手 456 执行

一次 Negative_Rating_Penalty 操作。”-> 生成指令 {'rider_id': 456, 'action_type': 'Negative_Rating_Penalty'}。

传递: 用户将这个指令通过直接调用或事件总线的方式发送给平台。

平台 Platform_Agent: “收到一个指令, 内容是 {'rider_id': 456, 'action_type': 'Negative_Rating_Penalty'}。好的, 我这就执行。”-> 调用内部积分系统的相应方法。

(3) 基于 LLM 的决策(LLM-DrivenDecision)-用于特殊情景

在某些需要复杂情景判断时, 可以调用 LLM 来模拟用户的心理活动。

触发:当出现一些特殊交互时, 例如骑手曾主动联系用户。

LLM 输入:

1. 用户人设:“你是一个对时间比较挑剔 (tolerance_level=0.8) 的用户。”

2. 事件描述:“你的外卖订单预计送达时间是 30 分钟, 但在第 25 分钟时, 骑手主动打电话给你, 非常诚恳地解释说因为商家出餐慢了, 可能会晚到 10 分钟左右, 并请求你的谅解。最终, 订单在第 38 分钟送达 (超时 8 分钟)。”

LLM 任务:“基于以上情况, 你会选择: A.理解骑手, 给予好评; B.虽然理解, 但不给评价; C.依然感到不满, 给予差评。请做出选择并说明你的心路历程。”

输出:一个更符合人类情感逻辑的评价结果。LLM 可能会判断“虽然超时了, 但骑手提前沟通的态度很好, 所以还是给个好评吧”, 这在现实中非常常见, 但纯规则系统很难模拟。

六、核心仿真流程与场景应用

(一) 步骤 0: 世界初始化(World Initialization)

根据报告的统计分布，生成一个包含数千甚至数万个异质性骑手 Agent 的 AgentSociety。每个 Agent 都带有独特的参数和 agent_archetype。

(二) 步骤 1-N: 仿真循环(Simulation Loop)

环境更新:推进 current_time_step，根据预设场景（如高峰期、恶劣天气）调整环境参数。

平台行动:平台生成订单，并根据供需比更新基础动态定价。

骑手决策 (混合模式) :

所有在线骑手并行执行基于规则的决策（如快速筛选掉 ENE 为负的订单）。

对于符合特定条件的骑手或订单，触发 **LLM** 进行认知层决策。

行动执行与状态更新:所有 Agent 执行决策（移动、配送、下线），并更新各自的动态属性。

信息传播:在社交网络中传播信息（可能包含 LLM 生成的策略性信息）。

周期性评估:每日结束时，触发 LLM 对每个 Agent 进行满意度评估。

1. 场景应用(Scenario Application)

运力不足场景: 手动或按事件使大量骑手下线。观察平台 `dynamic_pricing_multiplier` 的飙升, 以及剩余骑手 (特别是机会型 Agent) 利用议价优势的行为。

七、结果

八、框架演示与过程分析

为了验证 TriadDelivery 框架的功能完备性和分析潜力, 我们进行了一次小规模的试点运行。该运行环境包含 10 个骑手 Agent、10 个商家 Agent 和 10 个顾客 Agent, 仿真周期为 3 天。本章旨在展示该框架如何捕捉和再现骑手议价博弈中的关键动态过程, 而非旨在提供具有统计意义的普适性结论。

(一) 骑手异质性与初始态度设定

仿真开始时, 骑手 Agent 依据第四章所述的参数设定, 被赋予了异质性属性, 包括经验水平 (70%老骑手, 30%新骑手)、月收入目标和日工作时长偏好。同时, 依据研究设定了“求生型”、“机会型”和“灵活型”三种原型。

关键在于, 为测试模型在社会冲突情景下的动态表现, 我们将骑手对集体议价的初始态度设定为一个两极分化的实验条件: 50%的骑手被设定为支持 (态度分值 7), 另外 50%则持反对态度 (态度分值 3)。商家群体的态度则被设定为更为保守 (60%反对)。重申, 这并非一项经验调查结果, 而是一项用于激发模型动态的程序化设置。

(二) 社交网络重构与集体行动基础

在试点运行中，我们引入了社交网络重构机制以模拟集体议价联盟的形成。系统将骑手 Agent 原有的社交关系清空，并根据其被预设的集体议价态度，重新分配到“同意朋友”和“反对朋友”两个隔离的社交网络中。

这一机制的关键意义在于，它模拟了现实中劳动者通过线上社群或线下联络，基于共同诉求形成信息共享和行动协调网络的过程。如 Silver (2003) 所强调的“关联性权力”，这种基于共同立场的网络连接，是零工工人在缺乏正式组织时形成集体力量的基础。仿真结果表明，该机制能够成功运行，为后续策略信息的有效传播和同步行动（如集体下线）创造了必要的组织条件。

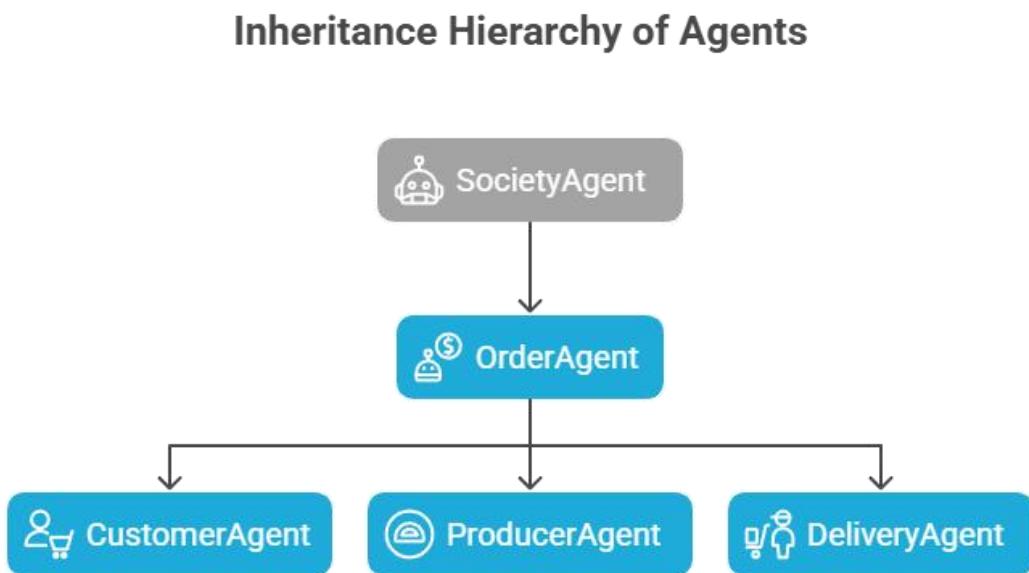


图 2 三级代理体系结构设计

(三) 运力变化与平台动态定价的反馈循环

本框架的核心功能之一，是清晰地展示骑手个体行为如何通过市场机制对平台算法产生影响。在“运力不足”场景测试中，当大量骑手（无论出于集体行动还是个人原因）选择下线时，系统内的供需比（在线空闲骑手数 / 订单池中的

订单数) 急剧下降。

平台的动态定价算法对此作出即时响应, `dynamic_pricing_multiplier` (动态定价乘数) 显著飙升。这一价格信号的变动产生了双重效应: 一方面, 它暂时抑制了部分价格敏感用户的下单需求; 另一方面, 高价订单吸引了仍在工作的“机会型”骑手和其他在线骑手积极接单, 甚至可能吸引部分已下线的骑手重新上线。这一过程形成了一个完整的反馈循环: 骑手集体行动 → 运力减少 → 平台提价 → 市场自我调节 → 形成新的平衡。这清晰地揭示了非正式议价权的涌现机制: 骑手的议价能力并非通过直接谈判获得, 而是通过改变市场供给状况, “迫使”算法系统自动调整价格来实现的。

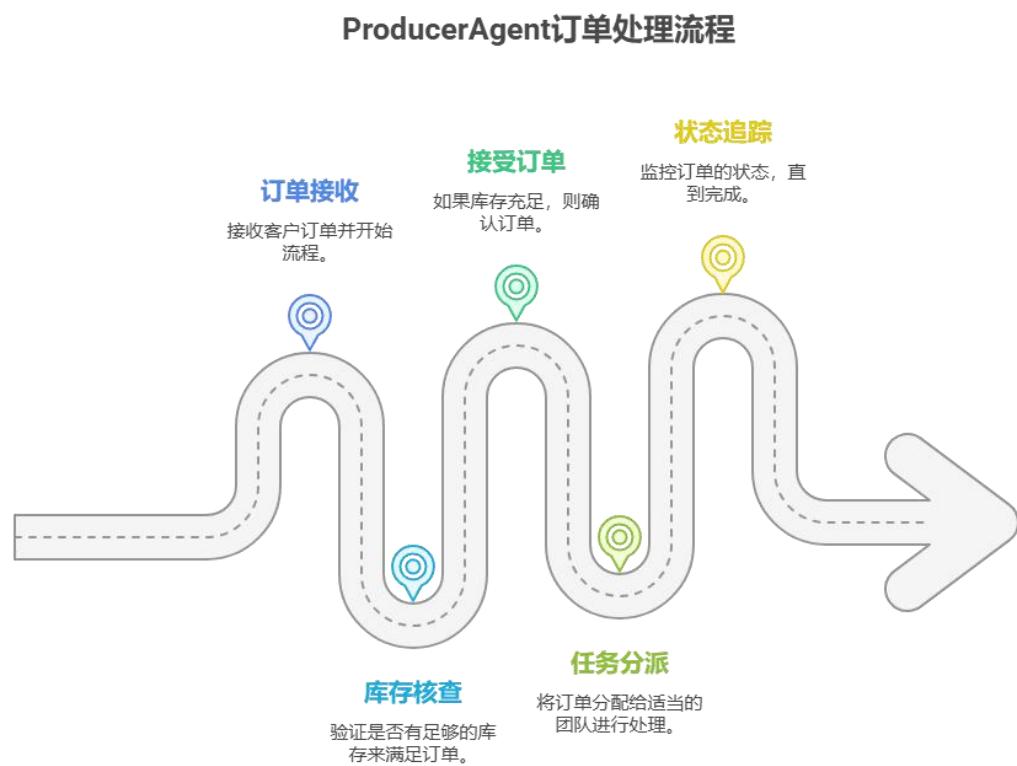


图 3 TriadDelivery 系统中的端到端订单处理流程

(四) 满意度驱动的混合决策模式

试点运行结果验证了混合决策模式的有效性。骑手的行为由计算层和认知层共同驱动。一个关键的驱动因素是骑手的动态满意度评分。该评分由 LLM 综合评估每日的收入、工作时长、天气、奖惩等多种因素后生成。满意度分数直接影响骑手的后续行为：满意度低于 6 时，骑手会触发“挑单”行为；满意度低于 3 时，骑手倾向于直接下线。这种基于满意度的行为模式，生动模拟了现实中骑手在“被困在系统里”(陈龙, 2020)与寻求自主性之间的矛盾与博弈，体现了劳动者在算法控制下的能动性。

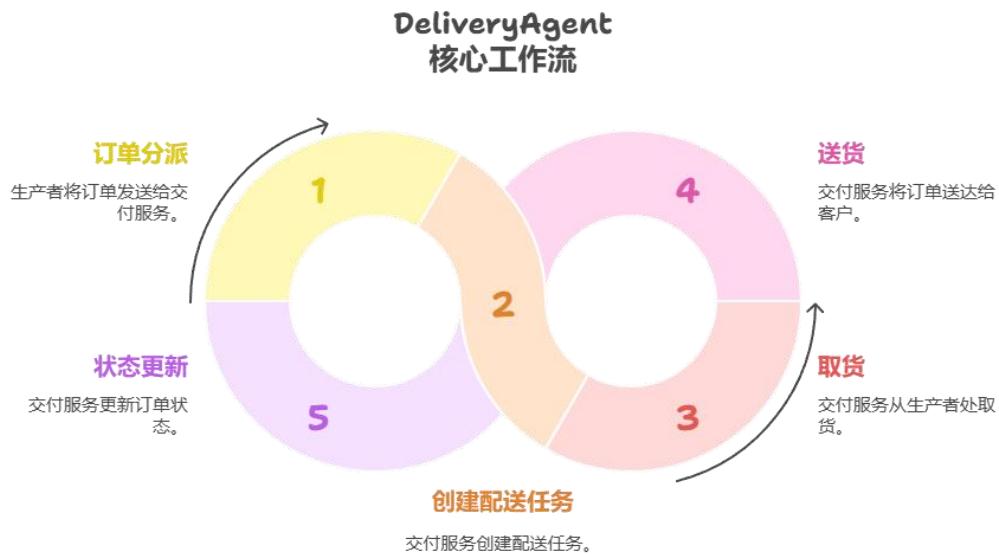


图 4 配送核心工作流程



图 5 系统内各代理协作全景图

九、讨论

本研究的仿真框架及其试点运行结果，为理解数字平台下的劳资博弈提供了新的理论洞见和方法论启示。

(一) “算法铁笼”中的能动性：非正式议价权的微观基础

我们的模型不仅重现了平台通过算法对劳动过程的精密控制，更重要的是，它以可计算的方式揭示了骑手能动性的具体作用路径。研究发现，骑手的议价权并非一个抽象概念，而是植根于其每一个具体的决策：是否接单、何时下线、如何应对低满意度。这些看似微不足道的个体行为，在特定条件下汇聚成一股能与算法“对话”的市场力量。这为斯科特(Scott, 1985)的“弱者的武器”理论在数

字劳动领域提供了新的实证支持，并清晰地展示了 Silver (2003) 的“结构性权力”（来源于市场地位）是如何在动态互动中被劳动者所利用的。

(二) 方法论的突破：从“计算代理”到“认知代理”

传统 ABM 中的 Agent 常被批评为“有计算无认知”的机器人(Sun, 2016)。

本研究的核心贡献在于，通过嵌入 LLM，我们构建了“认知代理”。这个代理不仅能计算预期收益，还能“理解”社交信息、评估主观满意度、并基于此进行策略调整。例如，基于满意度的挑单和下线行为，就是一种超越了简单经济最大化原则的、更符合人类心理的决策模式。这使得我们的仿真超越了纯粹的经济行为模拟，进入了社会心理与经济行为交织的复杂领域，极大地提升了模型的解释力与现实贴近性(Park et al., 2023)。

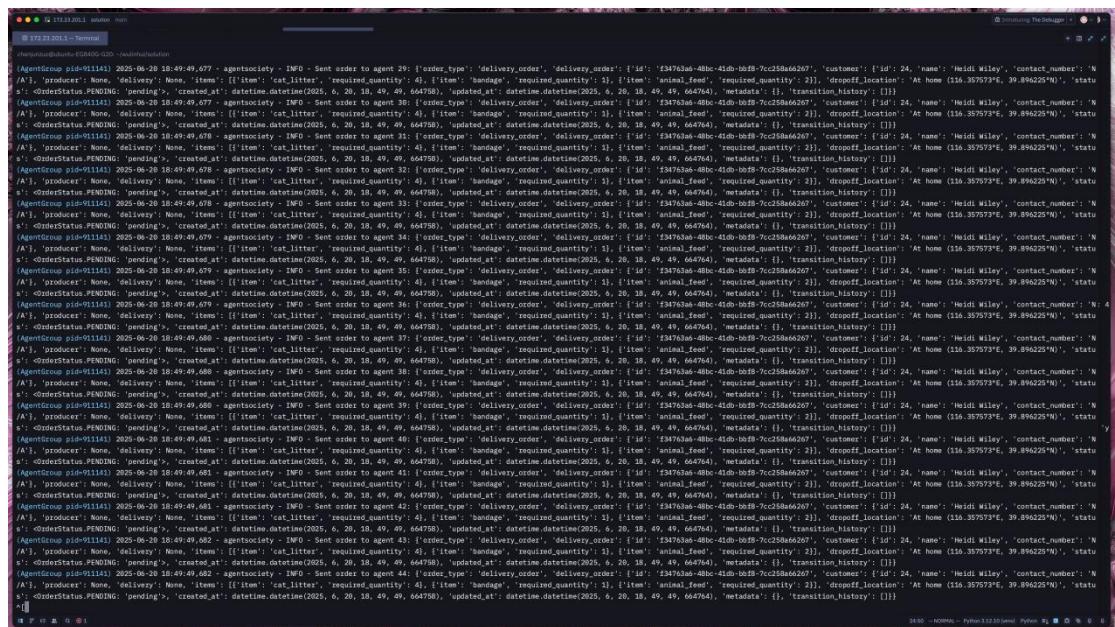


图 6 仿真运行过程中的多智能体交互状态

(三) 算法治理的实践启示

研究结果对平台治理和公共政策具有明确的启示。

对平台而言：仿真结果表明，试图完全压制骑手能动性的高压管理模式

(如通过严苛惩罚降低骑手挑单意愿) 可能会降低骑手满意度, 从而在关键时刻(如需求高峰) 引发更剧烈的运力波动或集体行动。一个更具韧性的治理策略, 应当承认并适度引导骑手的能动性, 例如建立更合理的申诉和反馈机制, 或在定价模型中更人性化地考虑劳动强度等因素。

对劳动者及组织者而言: 研究清晰地指出了信息网络和集体协调在提升议价能力中的关键作用。这提示劳动者权益组织, 应致力于构建和维护骑手间的信息共享平台, 并在关键时刻提供策略指导, 以合法合规的方式帮助骑手形成有效的集体声音。

同时, 我们也必须承认当前研究的局限性。首先, 实验规模相对较小 (10个骑手 Agent), 且仿真周期较短 (3 天), 这可能影响结果的普适性和对长期演化趋势的捕捉。其次, 模型对外部环境因素 (如宏观经济、多平台竞争) 的考虑尚不充分。未来的研究应致力于扩大仿真规模与时长, 引入多平台竞争, 并系统性地测试不同政策干预的效果, 同时寻求与真实世界数据的对比验证, 以不断提升模型的信度和效度。

十、参考文献

- [1] Acemoglu, D., & Robinson, J. A. (2012). *Why nations fail: The origins of power, prosperity, and poverty*. Crown Business.
- [2] Anwar, M. A., & Graham, M. (2020). Hidden transcripts of the gig economy: Labour agency and the new art of resistance among African gig workers. *Environment and Planning A: Economy and Space*, 52(7), 1269 – 1291.
- [3] Epstein, J. M., & Axtell, R. (1996). *Growing artificial societies: Social science from the bottom up*. Brookings Institution Press.
- [4] Flanders, A. (1968). Collective Bargaining: A Theoretical Analysis. *British Journal of Industrial Relations*, 6(1), 1-26.
- [5] Gilbert, N. (2008). *Agent-based models*. Sage.
- [6] Kellogg, K. C., Valentine, M. A., & Christin, A. (2020). Algorithms and the

- amplification of managerial control: A theory of remote-work algorithmization. *Academy of Management Review*, 45(4), 723-749.
- [7] Kenney, M., & Zysman, J. (2016). The rise of the platform economy. *Issues in Science and Technology*, 32(3), 61-69.
- [8] Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. W., Liang, P. P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (pp. 1-22).
- [9] Rosenblat, A. (2018). *Uberland: How algorithms are rewriting the rules of work*. University of California Press.
- [10] Rosenblat, A., & Stark, L. (2016). Algorithmic labor and information asymmetries: A case study of Uber's drivers. *International Journal of Communication*, 10, 3758-3784.
- [11] Scott, J. C. (1985). *Weapons of the weak: Everyday forms of peasant resistance*. Yale University Press.
- [12] Silver, B. J. (2003). *Forces of labor: Workers' movements and globalization since 1870*. Cambridge University Press.
- [13] Sun, P. (2019). Your order, their labor: An exploration of algorithms and laboring on food delivery platforms in China. *Chinese Journal of Communication*, 12(3), 308-323.
- [14] Sun, P., & Chen, Y. (2021). Counter-algorithmic practices on food-delivery platforms in China. *New Media & Society*, 23(12), 3466-3485.
- [15] Sun, R. (2016). *Anatomy of the mind: Exploring psychological mechanisms and processes with the Clarion cognitive architecture*. Oxford University Press.
- [16] Tesfatsion, L., & Judd, K. L. (Eds.). (2006). *Handbook of computational economics: Agent-based computational economics* (Vol. 2). Elsevier.
- [17] Vallas, S., & Schor, J. B. (2020). What do platforms do? Understanding the gig economy. *Annual Review of Sociology*, 46, 273-294.
- [18] Wood, A. J., Graham, M., Lehdonvirta, V., & Hjorth, I. (2019). *Good gig*,

- bad gig: Autonomy and algorithmic control in the global gig economy. *Work, Employment and Society*, 33(1), 56-75.
- [19] 常凯, & 郑小静. (2019). 平台经济背景下劳动关系的认定与规制. 法学研究, 41(1), 114-131. [Chang, K., & Zheng, X. J. (2019). Identification and regulation of labor relations in the context of the platform economy. *Chinese Journal of Law*, 41(1), 114-131.]
- [20] 陈龙. (2020). 困在系统里:数字资本主义时代的劳动控制——一项对平台外卖骑手的民族志研究. 社会学评论, 8(5), 63-82. [Chen, L. (2020). Trapped in the system: Labor control in the era of digital capitalism — An ethnographic study of platform food delivery riders. *Sociological Review of China*, 8(5), 63-82.]
- [21] 冯向楠, & 詹婧. (2019). 平台劳动中的抵抗:一项关于外卖骑手的研究. 新闻与传播研究, 26(11), 76-94. [Feng, X. N., & Zhan, J. (2019). Resistance in platform labor: A study on food delivery riders. *Journalism and Communication Research*, 26(11), 76-94.]
- [22] 罗斯琦, & 陈佳慧. (2024). 平台流动的悖论: 外卖骑手的能动性与劳动过程. 社会, 44(1), 136-163. [Luo, Q., & Chen, J. H. (2024). The paradox of platform mobility: Agency and labor process of food delivery riders. *Society*, 44(1), 136-163.]
- [23] 孙萍. (2019). “好”工作与“坏”工作:平台经济下“网约工”的劳动过程与工作体验研究. 社会, 39(5), 143-169. [Sun, P. (2019). “Good” work and “bad” work: A study on the labor process and work experience of “online contract workers” in the platform economy. *Society*, 39(5), 143-169.]
- [24] 吴清军, & 李贞. (2018). 算法权力:“互联网+”时代的新型权力形式. 探索

与争鸣, (11), 93-99. [Wu, Q. J., & Li, Z. (2018). Algorithmic power: A new form of power in the Internet+ era. *Exploration and Free Views*, (11), 93-99.]

十一、提交作品后续注意事项

1. 请将文档（页数不限，Word/PDF 格式）、代码（如有）及所有附件打包为 ZIP 文件，上传至“**BDSC 大赛作品提交收集**”线上云盘：

[https://cloud.tsinghua.edu.cn/u/d/d8873613d88c42cd978e/。](https://cloud.tsinghua.edu.cn/u/d/d8873613d88c42cd978e/)

ZIP 文件命名格式为：“赛道号_团队名称_作品名称”，例如“3_测试团队_某研究报告”。请尽量避免多次提交；如需提交多个版本，请标明版本号（例如“_v2”）。

2. 如果提交上有任何困难，可邮件发送到 (jy-w22@mails.tsinghua.edu.cn)