(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115510664 A
(43)申请公布日 2022.12.23

(21)申请号 202211225237.0
(22)申请日 2022.10.09
(71)申请人 东南大学
地址 210096 江苏省南京市玄武区四牌楼2号
(72)发明人 王帅 陆瑶 李宗晟 梅洛瑜
(74)专利代理机构 南京众联专利代理有限公司 32206
专利代理师 叶倩
(51)Int.Cl.
G06F 30/20(2020.01)
G06Q 10/08(2012.01)
G06Q 50/28(2012.01)
G06K 9/62(2022.01)
G06N 3/04(2006.01)
G06N 3/08(2006.01)
G06F 111/04(2020.01)

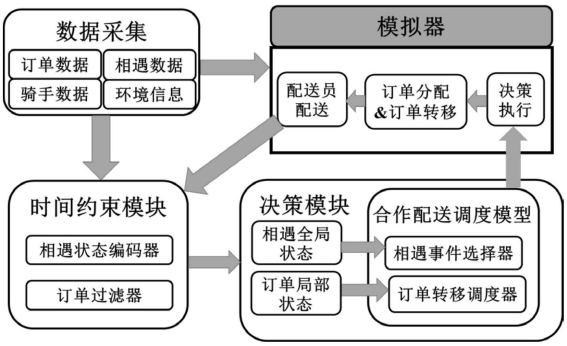
权利要求书4页 说明书13页 附图1页

(54)发明名称

基于分层强化学习的即时配送实时合作调度系统

(57)摘要

本发明公开了一种基于分层强化学习的即时配送实时合作调度系统,包括模拟器模块、决策模块和时间约束模块,模拟器模块用于模拟配送员在即时配送中的常规流程以及为算法构建训练和评估的环境,进行分配订单和调度配送员;决策模块:基于分层强化学习的合作配送调度模型,通过Actor-Critic网络提取特征,作出基于相遇交互的配送员合作配送决策,将该决策反馈至模拟器循环;时间约束模块综合考虑订单的实时剩余配送时间、订单的历史订单转移次数、即时配送的实时调度要求,对决策模块中的决策方案进行调度和指导。本系统通过调度推荐配送员相遇交互进行合作配送,以提高配送过程顺路单量、在满足配送时间约束的条件下提升配送效率、降低订单超时率的总体目标。



CN 115510664 A

1.基于分层强化学习的即时配送实时合作调度系统,其特征在于:包括模拟器模块、决策模块和时间约束模块,

所述模拟器模块:至少包括环境信息、配送员信息、订单信息和相遇信息,所有信息在模拟器模块中进行数据操作,用于模拟配送员在即时配送中的常规流程以及为算法构建训练和评估的环境,进行分配订单和调度配送员;

所述决策模块:采用基于分层强化学习的合作配送调度模型,根据模拟器模块收集到的配送员信息、订单信息和相遇信息,通过Actor-Critic网络提取特征,作出基于相遇交互的配送员合作配送决策,将该决策反馈至模拟器循环;

所述时间约束模块:根据相遇信息,提取配送员在相遇场景下的相遇交互时间约束特征,综合考虑订单的实时剩余配送时间、订单的历史订单转移次数、即时配送的实时调度要求,对决策模块中的决策方案进行调度和指导。

2.如权利要求1所述基于分层强化学习的即时配送实时合作调度系统,其特征在于:模拟器模块中,

所述环境信息的数据操作:至少包括每天的订单记录数据加载、每天的快递员配送记录及轨迹数据加载、每个时刻的配送员状态更新、实时可分配订单的配送员集合获取及初始的订单分派决策;

所述配送员信息的数据操作:至少包括配送员配送路径规划方案、订单分配及订单转移的接收、订单转移的丢弃、配送员接单或弃单的状态更新;

所述订单信息的数据操作:根据环境信息和配送员信息,对自身的订单信息进行逐一初始化及状态更新;

所述相遇信息:作为一个触发事件类,当检测到配送员之间的相遇时,获取相遇状态以支持订单转移决策。

3.如权利要求2所述基于分层强化学习的即时配送实时合作调度系统,其特征在于:所述决策模块中基于分层强化学习的合作配送调度模型,包括相遇时间选择器和订单转移调度器,

所述相遇事件选择器对相遇事件进行选择,观察高层状态$s_{k,t}^{h}$并给出一个动作$a_{k,t}^{h}$来决定是否在t时间的第k个相遇事件e_t^k时转移订单,当$a_{k,t}^{h} = 0$,相遇时间选择器选择不转移订单时,则继续处理下一次相遇事件;否则,调用订单转移调度器并在执行最后一个低层动作后接收反馈回来的延迟奖励;

所述订单转移调度器根据每个要转移的订单o_t^i的低层状态$s_{i,t}^{l}$,生成指示所选配送员接单o_t^i的低层动作$a_{i,t}^{l}$,环境接收分层动作$a_{k,t}^{h}$,$a_{k,t}^{l}$,并将低层和高层奖励反馈给订单转移调度器,使用相应的状态转换更新状态。

4.如权利要求3所述基于分层强化学习的即时配送实时合作调度系统,其特征在于:所述相遇时间选择器由高层参与者网络Actor及高层评价者网络Critic构成,高层参与者网络Actor根据编码的高层状态嵌入生成动作,以决定是否在每个相遇事件$e_t^k \in E_t$处转移订单,具体为:

获得在e_t^k相遇事件选择的可能长期回报$Q^h(s_{k,t}^h, a_{k,t}^h)$:

$$Q^h(s_{k,t}^h, a_{k,t}^h) = \begin{cases} \sum_{a_{k,t}^l} \pi_{\theta_{a^l}}(a_{k,t}^l | s_{k,t}^h) Q^l(s_{k,t}^h, a_{k,t}^h, a_{k,t}^l) \\ if\ a_{k,t}^h = 1 \\ r_{k,t}^h + \gamma V^h(s_{k,t+1}^h) \\ if\ a_{k,t}^h = 0 \end{cases}$$

其中，e_t^k处的高层状态定义为$s_{k,t}^h$；e_t^k处的高层动作定义为$a_{k,t}^h$；e_t^k处的低层动作定义为$a_{k,t}^l$；e_t^k处的高层奖励定义为$r_{k,t}^h$；折扣因子定义为γ；$\pi_{\theta_{a^l}}$表示由θ_{a^l}参数化的低层策略；Q^1:S×Ω×A→R是在给定高层动作$a_{k,t}^l \in \Omega$和观察状态的情况下执行低层动作$a_{k,t}^h \in A$的订单可转移值；$V^h(\bullet)$表示转移订单后的高层状态值；

在计算出可能的Q^h后，利用Softmax函数生成动作选择的概率，并提供高层策略$\pi_{\theta_{a^h}}$来决定相遇场合转移订单，所述高层策略$\pi_{\theta_{a^h}}$为：

$$\pi_{\theta_{a^h}}(a_{k,t}^h = \Omega_1 | s_{k,t}^h) = \frac{exp\left(Q^h(s_{k,t}^h, a_{k,t}^h = \Omega_1)\right)}{\sum_{\Omega'=\Omega_1}^{\Omega} exp\left(Q^h(s_{k,t}^h, a_{k,t}^h = \Omega')\right)}$$

其中，e_t^k处的高层状态定义为$s_{k,t}^h$；e_t^k处的高层动作定义为$a_{k,t}^h$；θ_{a^h}是高层参与者网络的网络参数。

5.如权利要求4所述基于分层强化学习的即时配送实时合作调度系统，其特征在于：所述高层评价者网络Critic通过高层参与者网络Actor根据$s_{k,t}^h$做出相遇事件选择决策时的状态值$(V_{\theta_{c^h}}^h(s_{k,t}^h))$来衡量长期奖励，所述状态值$\left(V_{\theta_{c^h}}^h(s_{k,t}^h)\right)$具体为：

$$V_{\theta_{c^h}}^h(s_{k,t}^h) = E\ (r_{t+1}^h + \gamma\ r_{t+2}^h + \gamma^2 r_{t+3}^h + \cdots + \gamma^{n-t} r_{n-t+1}^h)$$

其中，时间步t所有相遇事件的高层累计奖励定义为r_t^h, θ_{c^h}是高层评价者网络的网络参数。

6.如权利要求3所述基于分层强化学习的即时配送实时合作调度系统，其特征在于：所述订单转移调度器包括低层参与者网络Actor和低层评价者网络Critic，所述低层参与者

网络Actor根据低层编码状态嵌入生成动作来决定每个订单o_t^i在e_t^k传输给配送员c_t^j，具体为：通过三层前馈隐藏层，得到订单的可转移值Q^l，并通过Softmax函数输出最终的低层动作a^l，所述订单转移值$Q^l(s_{i,t}^l, a_{i,t}^l)$表示在给定$s_{i,t}^l$的情况下执行一个低层动作$a_{i,t}^l$的动作值，定义为：

$$Q^l(s_{i,t}^l, a_{i,t}^l) = r_{i,t}^l + \gamma \sum_{s_{i,t+1}^l} P(s_{i,t+1}^l | s_{i,t}^l, a_{i,t}^l) V^l(s_{i,t+1}^l)$$

其中，P(·)是低层状态转换概率，$V^l(s_{i,t+1}^l)$是执行$a_{i,t}^l$后的低层状态值；在e_t^k的订单o_t^i的低层动作生成遵循定义为的策略：

$$\pi_{\theta_{a^l}}(a_{i,t}^l = A_1 | s_{i,t}^l) = \frac{exp\left(Q^l(s_{i,t}^l, a_{i,t}^l = A_1)\right)}{\sum_{A'=A_1}^{A} exp Q^l(s_{i,t}^l, a_{i,t}^l = A')}$$

其中，A是e_t^k处的候选配送员的ID列表。

7.如权利要求6所述基于分层强化学习的即时配送实时合作调度系统，其特征在于：所述低层评价者网络Critic是函数逼近器，接收低层过滤状态嵌入作为输入，输出低层状态值V^l以评估$a_{i,t}^l$并反馈给低层参与者网络Actor进行更新策略网络参数；所述订单o_t^i在e_t^k的低层状态值函数$V_{\theta_{c^l}}^l(s_{i,t}^l)$定义为：

$$V_{\theta_{c^l}}^l(s_{i,t}^l) = E(r_{t+1}^l + \gamma r_{t+2}^l + \gamma^2 r_{t+3}^l + \cdots + \gamma^{n-t} r_{n-t+1}^l)$$

其中，θ_{c^l}是低层评价者网络的网络参数。

8.如权利要求5或7所述基于分层强化学习的即时配送实时合作调度系统，其特征在于：所述时间约束模块包括考虑时间约束的相遇状态编码器模块和订单过滤模块，所述考虑时间约束的相遇状态编码器模块用于对相遇开始时的状态进行编码，提取完整相遇过程的特征；所述订单过滤模块基于剩余配送时间和转移订单的频率来过滤不合适订单，所述过滤约束包括状态约束值和频率约束值，当状态约束值或频率约束值为1时，过滤订单。

9.如权利要求8所述基于分层强化学习的即时配送实时合作调度系统，其特征在于：所述状态约束中，使用状态上下文$SC_{o_t^i}$过滤剩余配送时间少的订单，该状态上下文被定义为e_t^k处订单o_t^i的二进制向量：

$$SC_{o_t^i} = \begin{cases} 1, & if RT_{o_t^i} <= \beta \\ 0, & 其他, \end{cases}$$

其中，$RT_{o_t^i}$是o_t^i的剩余配送时间，β是剩余时间阈值；

所述频率约束中，利用每个订单o_t^i在e_t^k的频率约束$F_{o_t^i,c_t^j}^M$来过滤不合适的订单，具体为

$$FC_{o_t^i} = \begin{cases} 1, & if TF_{o_t^i} > \epsilon^o \\ 0, & 其他, \end{cases}$$

其中，$TF_{o_t^i}$是o_t^i的转移时间，$\in^o$是订单的订单频率约束。

基于分层强化学习的即时配送实时合作调度系统

技术领域

[0001] 本发明属于计算机智能计算与运用的技术领域，主要涉及了一种基于分层强化学习的即时配送实时合作调度系统。

背景技术

[0002] 即时配送服务中，用户通过即时配送服务平台在家里或者公司在线下单；商店从平台在线接收相应的订单并开始准备商品；平台将订单调度分配给适当的配送员；配送员接单，到商店取得商品后将商品送达至用户处。在城市地区，平台调度一个配送员同时接多个订单进行并行派送，每个订单都有严格的配送时间约束。配送员状态和订单状态在实际配送的过程中不断动态变化，最优的配送员与订单匹配关系也会动态变化。这样的动态变化特征，给即时配送场景下的订单调度问题带来极大的挑战。即使根据相似度等要素对订单分组分派给配送员，配送员实时要送的订单组也不总是完全顺路。当一个配送员累积了这样的多个订单时，将会导致配送低效超时。因此在城市内订单量大而配送员数量有限、配送员和订单状态动态变化的背景下，设计一种有效的调度策略，在配送过程中动态调整配送员与订单间的匹配关系，提高配送员配送过程中的顺路单量，在满足配送时间的约束下提升平台配送效率是很有必要的。

[0003] 现有的配送员独立配送调度研究忽略了配送员之间的合作及配送过程中配送员状态和订单状态的动态变化，未充分利用配送员运力，当多单累积时易导致配送员配送低效；现有的合作调度方案主要针对基于部署的固定中转点基础设施、配送路线相对确定的场景，实现订单的分派和转移。由于(1)部署固定的额外基础设施需要较多成本；(2)即时配送配送员路线随实时分派的订单动态变化，如果到固定中转点会带来较远的额外绕行距离，较难满足即时配送的严格时间约束要求，因此这些方案不适合解决即时配送模式下的合作配送调度问题。其次，现有预测驱动的实时调度研究基于历史数据作相关信息的预测，指导实时的调度决策。由于基于预测的方法难以考虑未来不可控的变化，不适用；而现有检测驱动的实时调度相关研究仅应用在支持独立配送调度的紧急调度场景，不适合解决即时配送模式下的实时合作配送调度问题。

[0004] 近年来，随着实时配送系统智能升级，虚拟Beacon服务已经嵌入到配送员的智能手机，以支持相遇的可靠检测。虚拟Beacon服务支持智能手机同时广播和扫描蓝牙信号(包括设备ID等信息)，同时将扫描到的信息实时上传。通过匹配扫描到的设备ID与配送员ID，平台可以实时检测配送员之间的相遇事件。大量的配送员相遇事件为在配送过程中动态调整配送员与订单间的匹配关系(即配送员合作配送调度)提供了足够多的机会。

发明内容

[0005] 本发明正是针对现有技术中存在的问题，提供一种基于分层强化学习的即时配送实时合作调度系统，包括模拟器模块、决策模块和时间约束模块，模拟器模块：至少包括环境信息、配送员信息、订单信息和相遇信息，所有信息在模拟器模块中进行数据操作，用于

模拟配送员在即时配送中的常规流程以及为算法构建训练和评估的环境，进行分配订单和调度配送员；决策模块：采用基于分层强化学习的合作配送调度模型，根据模拟器模块收集到的配送员信息、订单信息和相遇信息，通过Actor-Critic网络提取特征，作出基于相遇交互的配送员合作配送决策，将该决策反馈至模拟器循环；时间约束模块：根据相遇信息，提取配送员在相遇场景下的相遇交互时间约束特征，综合考虑订单的实时剩余配送时间、订单的历史订单转移次数、即时配送的实时调度要求，对决策模块中的决策方案进行调度和指导。本系统通过调度推荐配送员相遇交互进行合作配送，以实现提高配送过程顺路单量、在满足配送时间约束的条件下提升配送效率、降低订单超时率的总体目标。

[0006] 为了实现上述目的，本发明采取的技术方案是：基于分层强化学习的即时配送实时合作调度系统，包括模拟器模块、决策模块和时间约束模块，

[0007] 所述模拟器模块：至少包括环境信息、配送员信息、订单信息和相遇信息，所有信息在模拟器模块中进行数据操作，用于模拟配送员在即时配送中的常规流程以及为算法构建训练和评估的环境，进行分配订单和调度配送员；

[0008] 所述决策模块：采用基于分层强化学习的合作配送调度模型，根据模拟器模块收集到的配送员信息、订单信息和相遇信息，通过Actor-Critic网络提取特征，作出基于相遇交互的配送员合作配送决策，将该决策反馈至模拟器循环；

[0009] 所述时间约束模块：根据相遇信息，提取配送员在相遇场景下的相遇交互时间约束特征，综合考虑订单的实时剩余配送时间、订单的历史订单转移次数、即时配送的实时调度要求，对决策模块中的决策方案进行调度和指导。

[0010] 作为本发明的一种改进，模拟器模块中，

[0011] 所述环境信息的数据操作：至少包括每天的订单记录数据加载、每天的快递员配送记录及轨迹数据加载、每个时刻的配送员状态更新、实时可分配订单的配送员集合获取及初始的订单分派决策

[0012] 所述配送员信息的数据操作：至少包括配送员配送路径规划方案、订单分配及订单转移的接收、订单转移的丢弃、配送员接单或弃单的状态更新；

[0013] 所述订单信息的数据操作：根据环境信息和配送员信息，对自身的订单信息进行逐一初始化及状态更新；

[0014] 所述相遇信息：作为一个触发事件类，当检测到配送员之间的相遇时，获取相遇状态以支持订单转移决策。

[0015] 作为本发明的一种改进，所述决策模块中基于分层强化学习的合作配送调度模型，包括相遇时间选择器和订单转移调度器，

[0016] 所述相遇事件选择器对相遇事件进行选择，观察高层状态$s_{k,t}^{h}$并给出一个动作$a_{k,t}^{h}$来决定是否在相遇e_t^k时转移订单，当$a_{k,t}^{h} = 0$，相遇时间选择器选择不转移订单时，则继续处理下一次相遇事件；否则，调用订单转移调度器并在执行最后一个低层动作后接收反馈回来的延迟奖励；

[0017] 所述订单转移调度器根据每个要转移的订单o_t^i的低层状态$s_{i,t}^{l}$，生成指示所选配送员接单o_t^i的低层动作$a_{i,t}^{l}$，环境接收分层动作$a_{k,t}^{h}$，$a_{k,t}^{l}$，并将低层和高层奖励反馈给订

单转移调度器，使用相应的状态转换更新状态。

[0018] 作为本发明的一种改进，所述相遇时间选择器由高层参与者网络Actor及高层评价者网络Critic构成，高层参与者网络Actor根据编码的高层状态嵌入生成动作，以决定是否在每个相遇事件$e_t^k \in E_t$处转移订单，具体为：

[0019] 获得在e_t^k相遇事件选择的可能长期回报$Q^h(s_{k,t}^h, a_{k,t}^h)$：

[0020] $$Q^h(s_{k,t}^h, a_{k,t}^h) = \begin{cases} \sum_{a_{k,t}^l} \pi_{\theta_{a^l}}(a_{k,t}^l | s_{k,t}^h) Q^l(s_{k,t}^h, a_{k,t}^h, a_{k,t}^l) \\ if\ a_{k,t}^h = 1 \\ r_{k,t}^h + \gamma V^h(s_{k,t+1}^h) \\ if\ a_{k,t}^h = 0 \end{cases}$$

[0021] 其中，e_t^k处的高层状态定义为$s_{k,t}^h$，e_t^k处的高层动作定义为$a_{k,t}^h$，e_t^k处的低层动作定义为$a_{k,t}^l$，e_t^k处的高层奖励定义为$r_{k,t}^h$，折扣因子定义为γ，$\pi_{\theta_{a^l}}$表示由θ_{a^l}参数化的低层策略；Q^l:S×Ω×A→R是在给定高层动作$a_{k,t}^l \in \Omega$和观察状态的情况下执行低层动作$a_{k,t}^h \in A$的订单可转移值；$V^h(\cdot)$表示转移订单后的高层状态值；

[0022] 在计算出可能的Q^h后，利用Softmax函数生成动作选择的概率，并提供高层策略$\pi_{\theta_{a^h}}$来决定相遇场合转移订单，所述高层策略$\pi_{\theta_{a^h}}$为：

[0023] $$\pi_{\theta_{a^h}}(a_{k,t}^h = \Omega_1 | s_{k,t}^h) = \frac{exp\left(Q^h(s_{k,t}^h, a_{k,t}^h = \Omega_1)\right)}{\sum_{\Omega' = \Omega_1}^{\Omega} exp\left(Q^h(s_{k,t}^h, a_{k,t}^h = \Omega')\right)}$$

[0024] 其中，e_t^k处的高层状态定义为$s_{k,t}^h$，e_t^k处的高层动作定义为$a_{k,t}^h$，θ_{a^h}是高层参与者网络的网络参数。

[0025] 作为本发明的又一种改进，所述高层评价者网络Critic通过高层参与者网络Actor根据$s_{k,t}^h$做出相遇事件选择决策时的状态值$(V_{\theta_{c^h}}^h(s_{k,t}^h))$来衡量长期奖励，所述状态值$\left(V_{\theta_{c^h}}^h(s_{k,t}^h)\right)$具体为：

[0026] $$V_{\theta_{c^h}}^h(s_{k,t}^h) = E(r_{t+1}^h + \gamma r_{t+2}^h + \gamma^2 r_{t+3}^h + \cdots + \gamma^{n-t} r_{n-t+1}^h)$$

[0027] 其中，时间步t所有相遇事件的高层累计奖励定义为r_t^h，θ_{c_h}是高层评价者网络的网络参数。

[0028] 作为本发明的又一种改进，所述订单转移调度器包括低层参与者网络Actor和低层评价者网络Critic，所述低层参与者网络Actor根据低层编码状态嵌入生成动作来决定每个订单o_t^i在e_t^k传输给配送员c_t^j，具体为：通过三层前馈隐藏层，得到订单的可转移值Q^1，并通过Softmax函数输出最终的低层动作a^1，所述订单转移值$Q^l(s_{i,t}^l, a_{i,t}^l)$表示在给定$s_{i,t}^l$的情况下执行一个低层动作$a_{i,t}^l$的动作值，定义为：

[0029] $$Q^l(s_{i,t}^l, a_{i,t}^l) = r_{i,t}^l + \gamma \sum_{s_{i,t+1}^l} P\left(s_{i,t+1}^l \middle| s_{i,t}^l, a_{i,t}^l\right) V^l(s_{i,t+1}^l)$$

[0030] 其中，P(·)是低层状态转换概率，$V^l(s_{i,t+1}^l)$是执行$a_{i,t}^l$后的低层状态值；在e_t^k的订单o_t^i的低层动作生成遵循定义为的策略：

[0031] $$\pi_{\theta_{a^l}}(a_{i,t}^l = A_1 | s_{i,t}^l) = \frac{exp\left(Q^l\left(s_{i,t}^l, a_{i,t}^l = A_1\right)\right)}{\sum_{A'=A_1}^{A} exp Q^l\left(s_{i,t}^l, a_{i,t}^l = A'\right)}$$

[0032] 其中，A是e_t^k处的候选配送员的ID列表。

[0033] 作为本发明的另一种改进，所述低层评价者网络Critic是函数逼近器，接收低层过滤状态嵌入作为输入，输出低层状态值V^1以评估$a_{i,t}^l$并反馈给低层参与者网络Actor进行更新策略网络参数；所述订单o_t^i在e_t^k的低层状态值函数$V_{\theta_{c^l}}^l(s_{i,t}^l)$定义为：

[0034] $$V_{\theta_{c^l}}^l(s_{i,t}^l) = E(r_{t+1}^l + \gamma r_{t+2}^l + \gamma^2 r_{t+3}^l + \cdots + \gamma^{n-t} r_{n-t+1}^l)$$

[0035] 其中，θ_{c^l}是低层评价者网络的网络参数。

[0036] 作为本发明的另一种改进，所述时间约束模块包括考虑时间约束的相遇状态编码器模块和订单过滤模块，所述考虑时间约束的相遇状态编码器模块用于对相遇开始时的状态进行编码，提取完整相遇过程的特征；所述订单过滤模块基于剩余配送时间和转移订单的频率来过滤不合适订单，所述过滤约束包括状态约束值和频率约束值，当状态约束值或频率约束值为1时，过滤订单。

[0037] 作为本发明的更进一步改进，所述状态约束中，使用状态上下文$SC_{o_t^i}$过滤剩余配送时间少的订单，该状态上下文被定义为e_t^k处订单o_t^i的二进制向量：

[0038] $$SC_{o_t^i} = \begin{cases} 1, & if RT_{o_t^i} <= \beta \\ 0, & 其他, \end{cases}$$

[0039] 其中，$RT_{o_t^i}$是o_t^i的剩余配送时间，β是剩余时间阈值；

[0040] 所述频率约束中，利用每个订单o_t^i在e_t^k的频率约束$F^M_{o_t^i,c_t^j}$来过滤不合适的订单，具体为

[0041] $$FC_{o_t^i} = \begin{cases} 1, & if TF_{o_t^i} > \epsilon^o \\ 0, & 其他, \end{cases}$$

[0042] 其中，$TF_{o_t^i}$是o_t^i的转移时间，$\in^o$是订单的订单频率约束。

[0043] 与现有技术相比，本发明具有的有益效果：提供了一种基于分层强化学习的即时配送实时合作调度系统，首先，在相遇事件选择器和考虑相遇的订单转移调度器的分层设计中，利用配送时间约束(即承诺的配送时间)计算平台收入和配送员收入作为奖励项的约束，并用相遇事件选择器过滤不合适的配送员相遇事件，以从相遇场景方面缩小强化学习的状态空间，减少调度计算时间；同时利用考虑时间约束的相遇状态编码器来提取完整相遇过程的表示嵌入向量(例如，相遇约束时间嵌入)，以支持实时的配送员合作调度；除此之外，在考虑时间约束的订单过滤模块中，考虑配送时间约束以及转移频率约束过滤不合适的订单，以加速在线调度，提高整体决策性能。本案系统基于相遇交互的配送员实时合作调度，在尽可能不改变配送员原有配送路径的条件下，高效地实现配送员间的合作配送，有效地提升整体配送效率及收益、降低订单超时率，可充分利用于外卖业务、跑腿业务、快车服务等行业，适用范围广泛。

附图说明

[0044] 图1为本发明基于分层强化学习的即时配送实时合作调度系统的结构图；

[0045] 图2为本发明基于分层强化学习的即时配送实时合作调度系统中模拟器模块的工作流程图。

具体实施方式

[0046] 下面结合附图和具体实施方式，进一步阐明本发明，应理解下述具体实施方式仅用于说明本发明而不用于限制本发明的范围。

[0047] 实施例1

[0048] 本实施例中涉及的相关符号定义具体如下：

[0049]

特征	符号	说明
订单	o_t^i	时间t要调度的第i个订单。
配送员	c_t^j	时间t，正在工作的第j个配送员。
配送员相遇事件	e_t^k	当两名或两名以上的配送员彼此都在对方的虚拟 Beacon 检测范围（15 米）内，且信号检测的时间间隔不超过 10 秒时，可视作其为一次配送员相遇事件。相遇事件e_t^k表示在时间t的第k个相遇事件。
订单配送时间	ODT	订单o_t^i的创建时间T_c^i与配送时间T_d^i(之间的差值，即$T_d^i - T_c^i$。
配送员配送时间	CDT	配送员c_t^j在时间t需要完成所有已接订单的总配送时间。
订单逾期率	OOR	超时订单数量占总订单数量的百分比。订单未及时完成即为超时，即$t_{pd}^i < (T_d^i - T_c^i)$。
平台收入	ORP	针对订单o_t^i的平台收入为订单价格p^i减去o_t^i的超时补偿费OC^i，即$ORP^i = p^i - OC^i$。

[0050]

配送员收入	CDR	我们假设配送员根据各自的配送距离分摊一个订单o_t^i的配送费和超时罚款。定义c_t^j为o_t^i的配送员收入为分配的配送费AF_j^i减去分配的o_t^i超时罚款AOF_j^i，即$CDR_j^i = AF_j^i - AOF_j^i$。

[0051] 基于分层强化学习的即时配送实时合作调度系统，如图1所示，包括如下模块：模块一：模拟器模块，用于模拟配送员在即时配送中的常规流程以及为算法构建训练和评估的环境；通过实际配送过程中平台采集的订单实时状态、配送员GPS位置、检测到的配送员相遇信息及区域的地理信息、道路交通状态；提取该部分数据的特征，用于驱动模拟器。我们构建的模拟器可以根据输入数据初始化配送员和订单的分布，并模拟真实配送环境来分配订单和调度配送员，流程如图2所示。

[0052] 首先构建模拟配送环境的底层类以驱动模拟器运行，设计如下：

[0053] 在模拟器中，我们构建了四个类，以便为模拟器中的操作构建对象，分别为：环境(Environment)、配送员(Cour ier)、订单(Order)和相遇(Encounter)；以下将分别对四个类进行定义。

[0054] 环境(Environment)：作为整体的模拟配送环境对象类，主要负责初始化及配送模拟过程中根据决策更新整体的环境状态；自身属性包括时间、实验区域内的配送员对象集合、订单对象集合等；主要操作包括实验区域每天的订单记录数据加载，实验区域每天的快递员配送记录及轨迹数据加载，每个时刻的配送员状态更新，实时可分配订单的配送员集合获取及初始的订单分派决策。

[0055] 配送员(Courier)：自身属性包括开始工作时间、配送员ID、当前所处GPS位置、正在负责配送的订单状态及信息集合、当前的配送路径站点列表信息、配送时间等；主要操作

包括配送员配送路径规划方案、订单分配及订单转移的接收、订单转移的丢弃、配送员接单或弃单的状态更新。配送员是实现合作配送调度的主体对象之一，在自身按照即时配送流程执行操作的同时，会接收到合作配送调度模型的交换订单指令，进而执行状态的更新。

[0056] 订单(Order)：自身属性包括订单ID、商家接单位置、用户送达位置、订单创建时间、到达商家时间、订单送达时间、订单预计送达时间、当前订单配送所处的阶段、订单被转移的历史次数、订单的价格、配送费等；主要操作为通过输入的数据对自身的订单信息进行逐一初始化及状态更新。订单类仅仅是用于创建订单对象以便于订单分派及转移的操作，关于订单自身，在本专利的模拟器中没有额外的自身操作。

[0057] 相遇(Encounter)：自身属性包括相遇的配送员ID列表，相遇开始的时间，相遇结束的时间，相遇时配送员的订单数量及订单对象列表等；主要操作为通过输入的数据对自身的订单信息进行逐一初始化及状态更新。相遇类作为一个触发事件类，当检测到配送员之间的相遇时，获取相遇状态以支持订单转移决策，关于相遇自身，在本专利的模拟器中没有额外的自身操作。

[0058] 在模拟器中，我们对即时配送场景下的各类事件进行建模，包括订单产生、订单分派、配送员交换订单、配送员基于路径规划的位置更新。配送员在接到新订单后，根据其具体情况更新自己的后续配送路径，然后再新接单和配送订单。各类事件的时间线设计如下：(1)在模拟器中新增订单，提取订单特征、配送员时空分布特征、配送员路径规划情况和道路环境特征。(2)基于上述提取的特征，检测配送员间的相遇事件，以提取配送员各个相遇场景下的特征。(3)进行订单转移决策，反馈新的配送员-订单匹配对。(4)在仿真器中执行新的配送员-订单匹配结果，更新每个配送员的状态和后续配送路径选择。

[0059] 其中，根据配送员历史轨迹数据，我们提取统计各个场景下(不同时空分布、订单状态、天气、交通状况)的配送速度分布，并根据提取的统计数据设置配送员的实时配送速度；我们从订单状态记录中提取配送员接单后的反应时间，即在相同时空分布下、订单接收时间与订单创建时间之间的差值；我们根据配送员的实时速度及他们的路径规划，计算配送员停下原有配送行为作相遇碰头交换订单的交通时间；根据配送员到店时间与到店取单的时间差，从订单状态记录获取配送员间交换订单的处理时间特征。考虑到配送相遇行为时间上的细粒度性，模拟器将时间步设置为1秒。此外，结合实际情况，模拟器设置中也添加了对配送员相遇交换订单过程的实际定量约束。

[0060] 模拟器中配送员的行为模式如下，若不进行相遇交换订单调度的决策，实验区域内的配送员按原有的订单分配根据平台推荐的路径正常进行订单配送；若进行相遇交换订单调度的决策，则根据合作配送调度模型推荐的找到最近的订单安全交换地点作订单转移操作，交换完成更新路径规划以进行新订单集合的配送。我们将模拟器部署在真实平台的开发环境中，开发环境与真实世界生产环境下有相同的数据流和接口，配送员数量及每个配送员的状态、每个订单的状态在每个时间步中进行变化。

[0061] 模块二：决策模块，采用基于分层强化学习的合作配送调度模型，针对检测到的大量配送员相遇交互事件，根据从模拟器模块收集的订单状态、配送员状态、相遇状态输入，通过Actor-Critic网络提取特征，作出基于相遇交互的配送员合作配送决策，来决定相遇的配送员间具体如何转移订单，即是否在当前相遇场景进行订单转移、对于每个订单决定是否转移、转移给哪位相遇配送员。针对检测到的大量配送员相遇事件，从即时配送平台提

取订单记录、配送员轨迹数据、配送员相遇信息，对复杂相遇场景进行特征分析、特征提取与建模，获取不同时间步下的配送员相遇时空分布、相遇时的配送员状态及订单状态信息，输入合作配送调度模型获取配送员合作配送调度决策，将该决策反馈至模拟器循环从而根据数据进行合作配送调度模型的训练。

[0062] 该合作配送调度模型方法主体框架为基于Actor-Critic算法的分层强化学习技术框架，主要包括相遇事件选择器与订单转移调度器，每层的内部都各包含一个Critic网络和Actor网络，用于评估策略网络与输出合作配送调度策略。在不同相遇场景下，考虑相遇的订单合作调度旨在(1)对复杂的相遇场景建模并选择合适的相遇场景；(2)在选定的相遇场景，调度相遇配送员间的订单转移。受分层抽象机理论的启发，我们将问题表述为分层马尔可夫决策过程，即将整个任务分解为两种子任务：(1)高层任务M_h：使用相遇事件选择器对相遇事件进行选择，(2)低层任务M_l：使用订单转移调度器进行考虑相遇的订单转移调度。具体来说，给定相遇事件e_t^k下的状态，智能体(即模型)选择分层动作(即(a)是否在相遇e_t^k时转移订单，或(b)哪一名相遇的配送员接收转移的订单o_t^i)，并将环境(即真实配送环境)执行这一系列决策后提供的反馈作为奖励(例如，订单配送时间ODT、平台收入ORP、配送员配送时间CDT、配送员收入CDR)。然后智能体从奖励中学习并更新状态。我们的最终目标是最大化每一回合(例如一天)的预期累积奖励，即通过订单在相遇配送员间的局部转移优化全局配送效率、收入和订单超时率。

[0063] 考虑到配送过程中的动态变化，我们将分层强化学习各个组件的定义如下：高层状态s^h：e_t^k处的高层状态定义为$s_{k,t}^h = \{\varepsilon_t^k, \epsilon_t^k, g_t^k\}$。

[0064] -ε_t^k：ε_t^k是相遇状态，包括相遇的时间步和位置，相遇趋势嵌入τ_t^k。为了模拟相遇趋势，我们构建一个相遇距离序列$seq_t^k = \{d_{t-(\xi-1)v}^k, \ldots, d_{t-v}^k, d_t^k\}$，其中ξ是序列的长度。每个元素是一段时间v(例如30秒)内的配送员间距离。然后我们利用门控循环单元网络GRU来提取相遇趋势嵌入$\tau_t^k = GRU(seq_t^k)$。

[0065] -ϵ_t^k：ϵ_t^k是配送员在e_t^k的状态，包括他们的实时位置、当前容量、他们的下一站位置(商家或客户的位置)、到达下一站的剩余配送时间和配送员的订单转移操作的频率。

[0066] -g_t^k：g_t^k包含订单的总体信息：(1)同一配送员配送的订单相似度，(2)不同配送员配送的订单相似度，(3)订单的最短逾期时间，以及(4)订单的最长剩余配送时间。订单的相似度是通过订单嵌入的向量之间的余弦相似度来计算的。

[0067] 高层动作a^h：e_t^k的高层动作定义为$a_{k,t}^h \in \Omega = \{0,1\}$，这是一个二进制值，用来表示相遇$e_t^k$时是否转移订单。如果$a_{k,t}^h = 0$，则表示在$e_t^k$没有执行任何订单转移动作。

[0068] 高层奖励r^h：e_t^k的高额奖励由两个因素衡量，考虑到配送员转移订单的意愿：(1)配送员总配送时间CDT和(2)配送员在e_t^k的总收入CDR，正式定义为$r_{k,t}^h$。

[0069] $$r_{k,t}^{h} = \varphi\frac{CDT^{b}}{CDT^{a}} + (1-\varphi)\frac{CDR^{a}}{CDR^{b}}$$

[0070] 其中CDT^{b}、CDT^{a}分别为配送员换单前后的总CDT，CDR^{b}、CDR^{a}为配送员的总CDR分别在合作配送之前和之后的当前订单。CDT^{a}越短，CDR^{a}越高，奖励项越大。$\varphi \in [0,1]$是权重因子。

[0071] 低层状态s^{l}：低层状态包含每个订单o_t^i在e_t^k中转的实时订单状态和配送员与订单间的匹配信息，包括o_t^i的剩余配送时间、o_t^i的下一站位置、类似于o_t^i的订单，o_t^i的下一站位置与配送员的下一站位置的相似度，配送员当前已分配的订单量，以及配送员当前的配送时间CDT。

[0072] 低层动作a^{l}：我们将每个订单o_t^i在e_t^k转移的低层动作定义为$a_{i,t}^{l} = CID$，其中$CID \in A$是所选配送员的配送员ID。

[0073] 如果CID等于o_t^i的原始配送员ID，则表明没有关于o_t^i的订单转移动作。低层奖励r^{l}：低层奖励将每个订单o_t^i在e_t^k的订单配送时间ODT和平台收入ORP考虑在内。

[0074] $$r_{i,t}^{l} = \varphi\frac{ODT_{i}^{b}}{ODT_{i}^{a}} + (1-\varphi)\frac{ORP_{i}^{a}}{ORP_{i}^{b}}$$

[0075] 其中ODT_i^b，ODT_i^a为o_t^i转移前后的ODT，ORP_i^b，ORP_i^a为o_t^i转移前后的ORP。

[0076] 在处理相遇e_t^k时，高层强化学习智能体(即相遇事件选择器)首先观察高层状态$s_{k,t}^{h}$并给出一个动作$a_{k,t}^{h}$来决定是否在e_t^k转移订单。如果高层强化学习智能体选择不转移订单(即$a_{k,t}^{h} = 0$)，则继续处理下一次相遇事件。否则，它调用低层强化学习智能体(订单转移调度器)并在执行最后一个低层动作后接收反馈回来的延迟奖励。低层强化学习智能体根据每个要转移的订单o_t^i的低层状态$s_{i,t}^{l}$，生成指示所选配送员接单o_t^i的低层动作$a_{i,t}^{l}$。环境接收分层动作$a_{k,t}^{h}, a_{k,t}^{l}$，并将低层和高层奖励反馈给智能体。最后，使用相应的状态转换更新状态。

[0077] 基于收集到的丰富数据，相遇事件选择器(高层强化学习智能体)旨在实际配送中优化平台配送效率。配送员一开始都根据平台派单方案进行独立配送。一旦检测到配送员相遇，系统会分析当前的相遇场景，作相遇场景的建模及选择，以决定配送员在当前相遇场景是否进行订单转移。相遇事件选择器根据状态输入，训练Actor-Critic网络以选择合适的相遇场景以进行配送员间的合作配送，在每个相遇事件$e_t^k \in E_t$计算配送员间转移订单的合适度，并选择合适的相遇事件转单，从而提升配送员的配送效率、收入并保证每个订单转移动作的及时性。主要由高层参与者网络Actor及高层评价者网络Critic这两个网络构成，具体设计如下：

[0078] 高层参与者网络Actor：高层参与者网络根据编码的高层状态嵌入(包括相遇嵌

入、配送员嵌入和一般订单嵌入)生成动作,以决定是否在每个相遇事件$e_t^k \in E_t$处转移订单。具体来说,我们将上述嵌入连接起来,并在将它们输入三层前馈隐藏层后得到相遇适合度值Q^h。$Q^h(s_{k,t}^h, a_{k,t}^h)$表示在$e_t^k$相遇事件选择的可能长期回报,定义为:

[0079] $$Q^h(s_{k,t}^h, a_{k,t}^h) = \begin{cases} \sum_{a_{k,t}^l} \pi_{\theta_{a^l}}(a_{k,t}^l | s_{k,t}^h) Q^l(s_{k,t}^h, a_{k,t}^h, a_{k,t}^l) \\ if\ a_{k,t}^h = 1 \\ r_{k,t}^h + \gamma V^h(s_{k,t+1}^h) \\ if\ a_{k,t}^h = 0 \end{cases}$$

[0080] 其中$\pi_{\theta_{a^l}}$表示由θ_{a^l}参数化的低层策略。

[0081] $Q^l: S \times \Omega \times A \rightarrow R$是在给定高层动作$a_{k,t}^l \in \Omega$和观察状态的情况下执行低层动作$a_{k,t}^h \in A$的订单可转移值。$V^h(\bullet)$表示转移订单后的高层状态值。

[0082] 在计算出可能的Q^h后,我们利用Softmax函数生成动作选择的概率,并提供高层策略$\pi_{\theta_{a^h}}$来决定相遇场合转移订单,其中θ_{a^h}是高层参与者网络的网络参数。

[0083] $$\pi_{\theta_{a^h}}(a_{k,t}^h = \Omega_1 | s_{k,t}^h) = \frac{exp\left(Q^h\left(s_{k,t}^h, a_{k,t}^h = \Omega_1\right)\right)}{\sum_{\Omega' = \Omega_1}^{\Omega} exp\left(Q^h\left(s_{k,t}^h, a_{k,t}^h = \Omega'\right)\right)}$$

[0084] 高层评价者网络Critic。高层评价者网络旨在通过高层参与者网络根据$s_{k,t}^h$做出相遇事件选择决策时的状态值$(V_{\theta_{c^h}}^h(s_{k,t}^h))$来衡量长期奖励。$\theta_{c^h}$收集价值网络的参数(高层评价者网络)。

[0085] $$V_{\theta_{c^h}}^h(s_{k,t}^h) = E(r_{t+1}^h + \gamma r_{t+2}^h + \gamma^2 r_{t+3}^h + \cdots + \gamma^{n-t} r_{n-t+1}^h)$$

[0086] 订单转移调度器(低层强化学习智能体)旨在决定如何在相遇事件选择器选择的相遇事件e_t^k中调整配送员与其订单之间的特定匹配关系,用于在选定的相遇事件中决策每个订单的交换动作,主要包括低层参与者网络Actor、低层评价者网络Critic两个网络,具体设计如下:

[0087] 低层参与者网络Actor:低层参与者网络根据低层编码状态嵌入生成动作来决定每个订单o_t^i在e_t^k传输哪个配送员c_t^j,包括详细的订单嵌入和配送员订单匹配嵌入。我们通过三层前馈隐藏层,得到订单的可转移值Q^l,并通过Softmax函数输出最终的低层动作a^l。订单转移值$Q^l(s_{i,t}^l, a_{i,t}^l)$表示在给定$s_{i,t}^l$的情况下执行一个低层动作$a_{i,t}^l$的动作值,定义为:

[0088] $$Q^l(s_{i,t}^l, a_{i,t}^l) = r_{i,t}^l + \gamma \sum_{s_{i,t+1}^l} P\left(s_{i,t+1}^l | s_{i,t}^l, a_{i,t}^l\right) V^l(s_{i,t+1}^l)$$

[0089] 其中P(•)是低层状态转换概率，$V^l(s_{i,t+1}^l)$是执行$a_{i,t}^l$后的低层状态值。在e_t^k的订单o_t^i的低层动作生成遵循定义为的策略：

[0090] $$\pi_{\theta_{a^l}}(a_{i,t}^l = A_1 \mid s_{i,t}^l) = \frac{exp\left(Q^l(s_{i,t}^l, a_{i,t}^l = A_1)\right)}{\sum_{A'=A_1}^{A} exp\, Q^l(s_{i,t}^l, a_{i,t}^l = A')}$$

[0091] 其中A是e_t^k处的候选配送员的ID列表。

[0092] 低层评价者网络Critic：低层评价者网络是函数逼近器，接收低层过滤状态嵌入作为输入，输出低层状态值V^l以评估$a_{i,t}^l$并反馈给低层参与者网络进行更新策略网络参数。o_t^i在e_t^k的低层状态值函数$V_{\theta_{c^l}}^l(s_{i,t}^l)$定义为：

[0093] $$V_{\theta_{c^l}}^l(s_{i,t}^l) = E(r_{t+1}^l + \gamma r_{t+2}^l + \gamma^2 r_{t+3}^l + \cdots + \gamma^{n-t} r_{n-t+1}^l)$$

[0094] 其中θ_{c^l}是低层评价者网络的网络参数。

[0095] 本步骤中，将复杂任务分解为多个子任务，以实现性能更好的解决方案；从相遇场景方面预先过滤不适合传递订单的相遇事件，缩小部分状态空间并加速调度以满足实时调度要求；此外，考虑多方面的奖励，从配送员、客户和平台等方面衡量配送效率、及时率和收入。

[0096] 模块三：时间约束模块，根据相遇数据，提取配送员在大量多样相遇场景下的相遇交互时间约束特征，综合考虑订单的实时剩余配送时间、订单的历史订单转移次数、即时配送的实时调度要求，对决策模块进行调度力度的指导，以实现在保证实时性的同时提供有效的配送员相遇转移订单推荐方案。

[0097] 时间约束模块主要包括两个子模块，即考虑时间约束的相遇状态编码器模块、订单过滤模块，具体设计如下：

[0098] (1)考虑时间约束的相遇状态编码器：考虑到相遇持续时长的限制，我们设计考虑时间约束的相遇状态编码器，以对配送员相遇的初始阶段用观察到的状态进行编码，来捕获完整的相遇特征(例如，相遇期限、相遇方向、相遇速度)，具体设计如下：

[0099] 一种融合多头自注意力和卷积的特征提取方案，利用它们各自在全局和局部信息建模方面的强大能力来帮助我们对整个相遇过程进行建模，这有利于相遇事件选择器和考虑相遇的订单转移调度器的后续决策。

[0100] 首先，我们使用符号$f_u \in F_t^k$来表示在e_t^k的高层状态或低层状态的特征组F_t^k中的第u个特征向量。然后对于第m个注意力头，我们使用编码矩阵W_Q^m、W_K^m、W_V^m将f_u投影到第m个查询、键、值表示中。我们计算以下Softmax函数：

[0101] $$\alpha_{u,z}^m = \frac{exp\left(W_Q^m f_u \cdot W_f \cdot (W_K^m f_z)^\top\right)}{\sum_{fz \in F_t^k} exp\left(W_Q^m f_u \cdot W_f \cdot (W_K^m f_z)^\top\right)}$$

[0102] 其中W_f是一个可训练的参数，用于考虑不同特征类型的影响(即相遇的特征、配送员的特征、配送员与订单之间的匹配特征、订单特征)，M是注意力机制的头数。此后，对于每个特征$f_z \in F_{k,t}$，我们用Softmax函数$\alpha_{u,z}^m$的输出对其值表示进行加权，然后将所有这些加权值表示加在一起作为第m个注意力头的输出。最后，我们将M个注意力头的输出连接起来，并将连接后的向量输入非线性ReLU激活函数，以输出第u个状态特征嵌入f_u'。

[0103] $$f_u' = ReLU(Concat(\sum_{f_z \in F_t^k} \alpha_{u,z}^l W_V^m f_z, \forall m \in M))$$

[0104] 这种状态特征嵌入收集每个状态特征的加权信息来进行全局注意力建模，将每个特征f_u投影到f_u'中。

[0105] 此外，我们还利用卷积层对相邻特征信息(即相同类型的特征)进行局部感知，旨在提取上述全局注意力嵌入$f_u' \in F'_{k,t}$的详细局部特征。对随后的相遇行为进行建模。然后我们将处理后的嵌入提供给相遇事件选择器和时间受限的订单过滤模块。

[0106] (2)考虑时间约束的订单过滤模块：本模块旨在预先过滤不适合交换的特定订单，旨在缩小强化学习的状态空间并加速调度以满足时间约束，具体设计如下：

[0107] 考虑到调度的实时性要求，利用过滤模块为低层强化学习智能体预先过滤不合适的订单，从而缩小低层状态空间以减少调度的计算时间。它还为低层强化学习智能体的探索提供了性能改进和订单转移开销之间的稳定权衡，这有利于协作订单调度的整体性能。具体来说，我们主要考虑剩余配送时间、转移订单的频率来过滤不合适的订单。如果o_t^i的状态约束值与频率约束值之一是1，我们为低层强化学习智能体过滤o_t^i。

[0108] 状态约束：状态约束主要考虑订单的状态和订单的剩余配送时间。首先，我们通过使用XGBoost预测配送员的配送路线来估计每个订单在e_t^k的配送时间。然后通过计算预计配送时间与当前时间之间的差值得到剩余配送时间。为了避免因额外的订单转移开销而产生的少量优化，我们使用状态上下文$SC_{o_t^i}$过滤剩余配送时间很少(例如10分钟)的订单，该状态上下文被定义为e_t^k处订单o_t^i的二进制向量：

[0109] $$SC_{o_t^i} = \begin{cases} 1, & if \quad RT_{o_t^i} <= \beta \\ 0, & 其他, \end{cases}$$

[0110] 其中$RT_{o_t^i}$是o_t^i的剩余配送时间，β是剩余时间阈值。

[0111] 频率约束：类似地，为了避免特定订单频繁转移的情况，这会带来更多的绕行时间，我们利用每个订单o_t^i在e_t^k的频率约束$F_{o_t^i,c_t^j}^M$来过滤不合适的订单。

[0112] $$FC_{o_t^i} = \begin{cases} 1, & if \quad TF_{o_t^i} > \epsilon^o \\ 0, & 其他, \end{cases}$$

[0113] 其中$TF_{o_t^i}$是o_t^i的转移时间，而$\in^o$是订单的订单频率约束(例如，一次)。本案的系

统基于部署的虚拟Beacon服务对相遇检测的支持，实现在尽可能不改变配送员原来的配送行为下通过配送员相遇交换订单，提高配送过程顺路单量，在满足配送时间的约束下提升平台整体配送效率，可用于在考虑时间约束条件下的资源分配任务，比如外卖业务、跑腿业务、快车服务等；可用于实时调度物流行业人员之间的合作，例如应用于顺风车、共享乘车业务等；也可用于解决动态变化场景下的实时调度问题，比如应用于智慧交通领域的救护车辆实时调度、机器人配送等；还可用于对交互行为建模分析及交互行为预测，利用个体数据信息及交互数据信息建立状态编码器模型技术，预测未来的交互行为，例如，应用于商业拉新业务等。

[0114] 需要说明的是，以上内容仅仅说明了本发明的技术思想，不能以此限定本发明的保护范围，对于本技术领域的普通技术人员来说，在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这些改进和润饰均落入本发明权利要求书的保护范围之内。

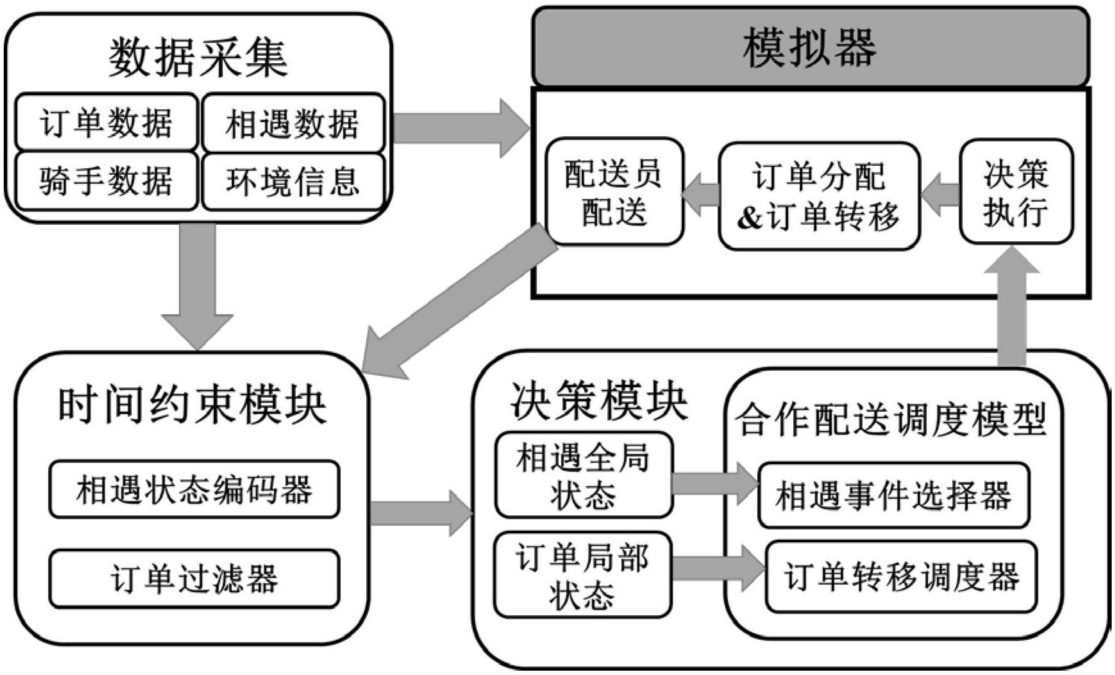

数据采集
订单数据
相遇数据
骑手数据
环境信息
模拟器
配送员配送
订单分配&订单转移
决策执行
时间约束模块
相遇状态编码器
订单过滤器
决策模块
相遇全局状态
订单局部状态
合作配送调度模型
相遇事件选择器
订单转移调度器

图1

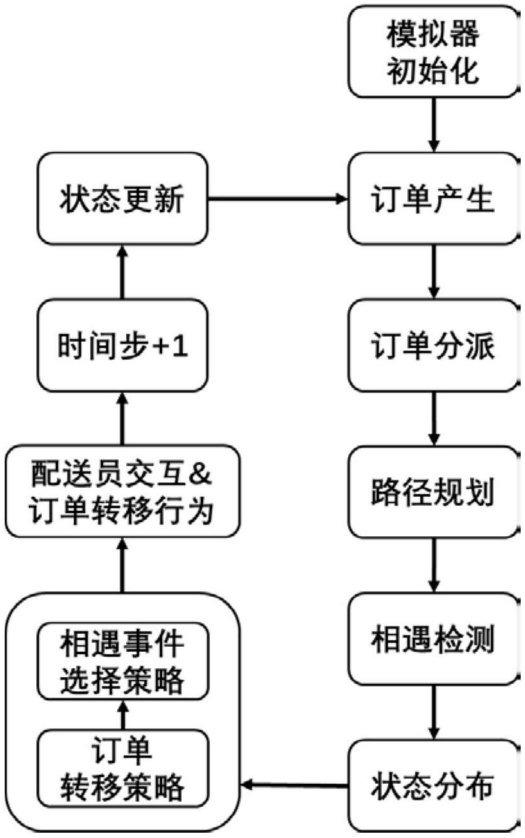

模拟器初始化
状态更新
订单产生
时间步+1
订单分派
配送员交互&订单转移行为
路径规划
相遇事件选择策略
订单转移策略
相遇检测
状态分布

图2