7. Appendix

7.1. Additional performance indicators

We analyze performance of our policies beyond total costs per order. In this appendix, we introduce additional performance indicators. Traditionally, allocation policies focus on the expected *drive-to-shop time* per order, which we define for a given policy Ω as

$$V_p^\Omega = \lim_{t\to\infty} \mathbb{E} \frac{1}{|\mathcal{I}_t|} \sum_{(i,j,t_{ij})\in\Omega_t} d_{ij}(t_{ij}). \tag{15}$$

Obviously, small V_p^Ω indicates an assignment of drivers near the restaurants. In our paper, we also consider the expected *waiting-in-shop time per order* that drivers wait at the restaurant,

$$V_w^\Omega = \lim_{t\to\infty} \mathbb{E} \frac{1}{|\mathcal{I}_t|} \sum_{(i,j,t_{ij})\in\Omega_t} (\tau_i + \xi_i - t_{ij} - d_{ij}(t_{ij}))^+, \tag{16}$$

which allows us to define the expected *total engagement time per order*, from matching to final delivery as $V_e^\Omega = V_p^\Omega + V_w^\Omega + \overline{\psi}$.

To analyze the service quality, we consider indicators such as expected *delay per order*,

$$V_d^\Omega = \lim_{t\to\infty} \mathbb{E} \frac{1}{|\mathcal{I}_t|} \sum_{(i,j,t_{ij})\in\Omega_t} \left(t_{ij} + d_{ij}(t_{ij}) - \tau_i - \xi_i\right)^+, \tag{17}$$

and *service level*,

$$V_l^\Omega = 1 - \lim_{t\to\infty} \mathbb{E} \frac{1}{|\mathcal{I}_t|} \sum_{(i,j,t_{ij})\in\Omega_t} \mathbf{1}\left(t_{ij} + d_{ij}(t_{ij}) - \tau_i - \xi_i > 0\right), \tag{18}$$

where $\mathbf{1}(\cdot)$ is the indicator function. Low V_d^Ω and high V_l^Ω indicate a high service quality.

We use two different indicators to measure the *efficiency of driver allocation or usage*:

$$\text{Expected idle time per driver}: V_q^\Omega = \lim_{t\to\infty} \mathbb{E} \frac{1}{|\mathcal{I}_t|} \sum_{(i,j,t_{ij})\in\Omega_t} (t_{ij} - \tau_j), \tag{19}$$

$$\text{Expected number of engaged drivers}: V_n^\Omega = \lambda_o \cdot V_e^\Omega, \tag{20}$$

where V_q^Ω describes the expected waiting-to-be-matched time of a driver. Short V_q^Ω implies short waiting for a driver to be assigned to an order, which is an important concern for participation of self-scheduled drivers in platform scheduling (Cachon et al. 2017). V_n^Ω indicates the total number of drivers engaged on average and relies on Little's Law.

The matching decisions have a significant impact on the number of orders that a driver can complete, ultimately affecting their earning potential (Ryan et al. 2018). Finally, as a reflection of drivers' social welfare, we analyze the expected *net-income per time unit of a driver*, which we define as:

$$V_s^\Omega = \frac{c_e \cdot V_e^\Omega - c_g \cdot v \cdot (V_p^\Omega + \overline{\psi})}{V_e^\Omega + V_q^\Omega}. \tag{21}$$

V_s^Ω can be interpreted as the ratio of a driver's total income $c_e \cdot V_e^\Omega$ (where c_e denotes the reward per unit time) minus driving cost $c_g \cdot v \cdot (V_p^\Omega + \overline{\psi})$ (where c_g is the cost per mileage, e.g., for gasoline, and v is the average driving speed) to the *work duration* of a driver, $V_e^\Omega + V_q^\Omega$. Indicator V_s^Ω reflects the net-income per time unit invested by a driver and is an important indication for the economic attractiveness of a matching policy for drivers.

7.2. SMDP model formulation

We formally define our model with known FPT (see Section 3) as an SMDP by describing its elements.

State: Even though the arrival processes are continuous, we consider that decisions are only made upon driver or order arrivals, driver abandonment, or edge expiration times. Hence, we can write $\mathbf{S}_n = (\mathbf{x}_n, \mathbf{y}_n, e_n, \tilde{t}_n)$ for the state that the planner observes at the time of the n^{th} event time. The set $\mathbf{x}_n = \{((\xi_i^r)_n, l_i)\}$ records information of all arrived but unmatched orders. Let $(\xi_i^r)_n$ represent the 'remaining' FPT of an open order i at the time of the n^{th} event, and l_i the order's restaurant location. We formulate the SMDP based on 'remaining' times, using the superscript $(^r)$ to distinguish remaining durations from absolute time epochs. $(\xi_i^r)_n$ can take negative values indicating that it already exceeds the FPT but the order is still not matched. $\tilde{t}_n$ is the 'time index', i.e., the current time in the periodicity δ of the time-dependent driving times. The set $\mathbf{y}_n = \{l_j\}$ indicates the location of each active driver j (note that we do not need to keep track of drivers' abandonment times as these are not known to the planner before the abandonment). Finally, the event indicator $e_n \in \{0,1,2,3\}$ indicates whether an order arrives ($e_n = 0$), a driver arrives ($e_n = 1$), a compatible edge expires ($e_n = 2$), or a driver abandons ($e_n = 3$). We denote the state space of all possible $\mathbf{S}_n$ by set $\mathcal{S}$.

Actions and costs: If the planner decides to match an order i with a driver j, $\mathbf{a}_n = (i,j)$; otherwise, no action is taken and $\mathbf{a}_n = \emptyset$. The action space is presented as $\mathbf{A}(\mathbf{S}_n)$ and contains all *feasible* matchings of orders and drivers. For a state $\mathbf{S}$ and action $\mathbf{a}$, the immediate cost is given by $C(\mathbf{S}_n, \mathbf{a}_n) = d_{ij}(\tilde{t}_n) + ((\xi_i^r)_n - d_{ij}(\tilde{t}_n))^+ + c \cdot (d_{ij}(\tilde{t}_n) - (\xi_i^r)_n)^+$ if $\mathbf{a}_n = (i,j)$ and $C(\mathbf{S}_n, \mathbf{a}_n) = 0$ if $\mathbf{a}_n = \emptyset$.

State transitions: For any state transition, the remaining FPT values are updated with inter-event (sojourn) time t_n^s by setting $(\xi_i^r)_n = (\xi_i^r)_{n-1} - t_n^s$ and time index $\tilde{t}_n$ is updated as $\tilde{t}_n = (\tilde{t}_{n-1} + t_n^s) \mod \delta$. If a driver abandons ($e = 3$), the driver is removed from the set of active drivers, i.e., $\mathbf{y}_n = \mathbf{y}_{n-1} \setminus \{l_j\}$. If the action $\mathbf{a}_n = (i,j)$ is selected, the planner will remove the corresponding elements respectively, i.e. $\mathbf{x}_n^+ = \mathbf{x}_n \setminus \{((\xi_i^r)_n, l_i)\}$ and $\mathbf{y}_n^+ = \mathbf{y}_n \setminus \{l_j\}$. At each transition, we can calculate the 'remaining' time to the expiration of an order-driver pair (i,j), i.e., $(t_{ij}^{c,r})_n$ from state $\mathbf{S}_n$, by $(t_{ij}^{c,r})_n = (\xi_i^r)_n - d_{ij}((t_{ij}^{c,r})_n + \tilde{t}_n)$.

State transitions depend on the sojourn time and probabilities to transit from a stage $\mathbf{S}_n$ to a stage $\mathbf{S}_{n+1}$ if action $\mathbf{a}_n$ is taken. The transition process is not purely Markovian because the remaining edge expiration times $t_{ij}^{c,r}$ are known and deterministic, only the order and driver arrival times and driver abandonment times are Markovian. Therefore, to describe the state transition behavior, we first need to characterize the random variable of the sojourn time $T^s(\mathbf{S}_n, \mathbf{a}_n)$ (with the realization being t_n^s), which is distributed according to cumulative distribution function

$$F(t^s|\mathbf{S},\mathbf{a}) = \begin{cases} \frac{1-e^{-(\lambda_o+\lambda_d+\lambda_a|\mathbf{y}|)t^s}}{1-e^{-(\lambda_o+\lambda_d+\lambda_a|\mathbf{y}|)\min\limits_{(i,j)}\left(t_{ij}^{c,r}|t_{ij}^{c,r}>0\right)}}, & t^s < \min\limits_{(i,j)}\left(t_{ij}^{c,r}|t_{ij}^{c,r}>0\right), \\ 1, & t^s \geq \min\limits_{(i,j)}\left(t_{ij}^{c,r}|t_{ij}^{c,r}>0\right), \end{cases} \tag{22}$$

where $|\mathbf{y}|$ is the number of unmatched drivers and $\min_{(i,j)}\left(t_{ij}^{c,r}|t_{ij}^{c,r}>0\right)$ is the minimum of the positive, feasible 'remaining' times to expiration of pairs $(i,j)\in\mathbf{A}(\mathbf{S}_n)$. If there is no pair (i,j) with $t_{ij}^{c,r}>0$, $F(t^s|\mathbf{S},\mathbf{a})$ reduces to $F(t^s|\mathbf{S},\mathbf{a})=1-e^{-(\lambda_o+\lambda_d+\lambda_a|\mathbf{y}|)t^s}$.

Based on the distributions of drivers and orders, and given that arrival times and locations are independent, we can formulate the transition probabilities $p_{\mathbf{S}_n,\mathbf{S}_{n+1}}(\mathbf{a}_n|t_{n+1}^s)$ from the above elements, which we omit for space considerations (details are available from the authors on request). The minimal expected total costs per order, $W^*(\mathbf{S}_0)$ (with initial state $\mathbf{S}_0$), is given by

$$W^*(\mathbf{S}_0)=\lim_{\tilde{n}\to\infty}\min_{\mathbf{a}}\mathbb{E}\left[\frac{\sum_{n=1}^{\tilde{n}}C(\mathbf{S}_n,\mathbf{a}_n)}{\sum_{n=1}^{\tilde{n}}\mathbf{1}(e_n=0)}\right], \tag{23}$$

where $\mathbf{1}(\cdot)$ is the indicator function. From the assumptions $\lambda_d>\lambda_o$ and $\lambda_a>0$ and the definition of the SMDP follows that at least for a policy that immediately matches order-driver-pairs, the embedded Markov chain of the SMDP is ϕ-irreducible, positive Harris-recurrent, and, hence, has a stationary distribution (Baxendale 2011).

7.3. Proofs

Proof of Proposition 1. If an edge is matched and it is not expired, according to the objective function, the planner is always weakly better off if she waits and matches them when the edge expires. If there is no edge in the graph, it means there are only orders (or drivers) in the system. When the next feasible driver (or order) arrives, it will establish a matching with an order either at a compatible edge expiring or instantly, because the edge with the new arrival is already expired. Note that because of the assumption $t+d_{ij}(t)<t'+d_{ij}(t')$ for any $t<t'$, any edge has only a unique edge expiring time. ■

Proof of Proposition 2. Using the cost function in Equation (2), we have the cost of matching the single order with driver j with remaining food processing duration ξ^r (recall that we drop order index i from the notation when considering the single-order-model) of $d_j+(\xi^r-d_j)^+ + c\cdot(d_j-\xi^r)^+$. From the above equation, we can make two observations: matching at the edge expiration time $\xi^r=d_j$ minimizes the cost of matching driver j to the single order, and if we consider only edge expiration times for matching, the best match among a given driver pool is the one with the shortest driving time d_j.

From the further observation that the driver with the shortest driving time is also the one for which the edge expires last, we can follow that it is never optimal to match a driver if one or more edges are not yet expired, which finalizes the proof. ■

Proof of Theorem 1. Parts (a1) and (a2): This proof first establishes two key arguments of the threshold structure of the optimal decision: First, we show that in the single-order-model the optimal policy decides only at two different points in time: At driver arrivals it chooses the better between the best existing driver and the incoming driver to 'keep in the system'. At edge expiration, the system

decides between matching the expiring driver and waiting for a better driver. The resulting stochastic process iterates over these times. Second, we show that for the decision at edge expiration, the future cost only depends on the remaining FPT and not on any other elements of the state space such as prior driver arrivals. This implies the threshold structure of the decisions. In the proof, we rely in parts on the formulation of the problem as an SMDP, as presented in Appendix 7.2.

From Proposition 2, we know that it is never optimal to allocate a driver to the single order at times other than the latest edge expiration time of existing drivers or the arrival of a new driver (if the edge is already expired at the time of arrival and there are no other non-expired edges). Hence, we can simplify the state $\mathbf{S}_n$ in our single-order system to $\mathbf{S}_n = (d_j, \xi^r)$ with driver j being the one whose edge expires last (the driver with the shortest travel time), and ξ^r being the remaining time until the food is prepared. Please note that before the arrival of the first driver, the state element d_j is naturally empty. In this case, we set $d_j = +\infty$.

Let us denote the expected minimum cost in state (d_j, ξ^r) at arrival time of driver j' (or at edge expiration time of driver j, $d_j = \xi^r$, in which case we set $d_{j'} = +\infty$) as $W(d_j, \xi^r | d_{j'})$. Furthermore, we use notation $W^+(d_j, \xi^r)$ to denote the expected minimum *future* cost right *after* a decision which results in state (d_j, ξ^r). We then construct function $W(d_j, \xi^r | d_{j'})$ from the following complete list of possible events.

(a) No arrival, but edge of existing driver j expires ($d_{j'} = \infty \wedge d_j = \xi^r$). In this case, it is optimal to *match or wait*, i.e., $W(d_j, \xi^r | d_{j'}) = \min\left\{d_j, W^+(d_j, \xi^r)\right\} = \min\left\{d_j + c\,(d_j - \xi^r)^+, W^+(d_j, \xi^r)\right\}$.

(b) Arrival of driver j' with expired edge and edge of existing driver j expired ($d_{j'} > \xi^r \wedge d_j > \xi^r$). In this case, it is also optimal to *match or wait*, i.e., $W(d_j, \xi^r | d_{j'}) = \min\left\{\min\{d_j + c\,(d_j - \xi^r)^+, d_{j'} + c\,(d_{j'} - \xi^r)^+\}, W^+(\min\{d_j, d_{j'}\}, \xi^r)\right\}$.

(c) Arrival of driver j' with expired edge and edge of existing driver j not expired ($d_{j'} > \xi^r \wedge d_j < \xi^r$). In this case, it is optimal to continue waiting with the non-expired edge, i.e., $W(d_j, \xi^r | d_{j'}) = W^+(d_j, \xi^r) = W^+(\min\{d_j, d_{j'}\}, \xi^r)$.

(d) Arrival of driver j' with non-expired edge and edge of existing driver j expired ($d_{j'} < \xi^r \wedge d_j > \xi^r$). In this case, it is optimal to wait with new driver and its non-expired edge, i.e., $W(d_j, \xi^r | d_{j'}) = W^+(d_{j'}, \xi^r) = W^+(\min\{d_j, d_{j'}\}, \xi^r)$.

(e) Arrival of driver j' with non-expired edge and edge of existing driver j not expired ($d_{j'} < \xi^r \wedge d_j < \xi^r$). In this case, it is optimal to wait, with the edge that expires last, i.e., $W(d_j, \xi^r | d_{j'}) = W^+(\min\{d_j, d_{j'}\}, \xi^r)$.

The expected *future* cost right after a decision and in state (d_j, ξ^r), $W^+(d_j, \xi^r)$, is given by taking the expectation over the next arrival:

$$W^+(d_j, \xi^r) = \mathbb{E}_{d_{j'}} \mathbb{E}_{\hat{t}_{j'}} \begin{cases} W(d_j, \xi^r - d_j | \infty), & \text{if } (\xi^r - d_j) \in \left[0, \hat{t}_{j'}\right], \\ W(d_j, \xi^r - \hat{t}_{j'} | d_{j'}), & \text{else,} \end{cases} \tag{24}$$

where j' is the next driver arrival with driving time $d_{j'}$ and exponential inter-arrival time $\hat{t}_{j'}$. Notably, the case $(\xi^r - d_j) \in \left[0, \hat{t}_{j'}\right]$ represents the event that the edge of driver j expires before the arrival of driver j'. Specifically, the equation can be enriched as

$$W^+(d_j,\xi^r) = \mathbb{E}_{d_{j'}}\mathbb{E}_{\hat{t}_{j'}} \begin{cases} \min\left\{d_j, W^+(\infty, \xi^r - d_j)\right\}, & \text{if } (\xi^r - d_j) \in \left[0, \hat{t}_{j'}\right], \\ W^+(d_j, \xi^r - \hat{t}_{j'}), & \text{if } (\xi^r - d_j) > \hat{t}_{j'} \text{ and } d_{j'} > (\xi^r - \hat{t}_{j'}), \\ W^+(\min\{d_j, d_{j'}\}, \xi^r - \hat{t}_{j'}), & \text{if } (\xi^r - d_j) > \hat{t}_{j'} \text{ and } d_{j'} \leq (\xi^r - \hat{t}_{j'}), \\ \min\left\{\min\{d_j + c\left[d_j - \xi^r\right]^+, d_{j'} + c\left[d_{j'} - \xi^r\right]^+\}, W^+(\infty, \xi^r - \hat{t}_{j'})\right\}, & \\ & \text{if } (\xi^r - d_j) < 0 \text{ and } d_{j'} > (\xi^r - \hat{t}_{j'}), \\ W^+(d_{j'}, \xi^r - \hat{t}_{j'}), & \text{if } (\xi^r - d_j) < 0 \text{ and } d_{j'} \leq (\xi^r - \hat{t}_{j'}). \end{cases} \tag{25}$$

Please note that, without loss of generality, we abused notation by writing the expectation $\mathbb{E}_{d_{j'}}$ for the expected travel time resulting from driver location $l_{j'}$.

The decision between matching the current driver and waiting for the future only occurs in case (a), and in case (b) if $d_{j'} < d_j$. For these two cases, the decision equation is the same, $\min\left\{d_j + c\left[d_j - \xi^r\right]^+, W^+(d_j, \xi^r)\right\}$, with j replaced by j' in the second case.

To show that the threshold with which drivers are matched depends only on the remaining FPT, we have to show that future cost are independent of prior driver arrivals. In order to show this result, we need the result that the optimal policy will never match the single order with an existing driver with an expired edge. Recall that an existing driver can only have an expired edge if it has not been matched at the time the edge expired (or the driver arrival time, whatever was later), i.e., if condition $d_j + c\left(d_j - \xi^r\right)^+ > W^+(d_j, \xi^r)$ was true at time $d_j = \xi^r$ (or the arrival time if the edge was already expired at arrival, $d_j > \xi^r$). From this time onwards, term $d_j + c\left(d_j - \xi^r\right)^+ = (c+1)d_j - c\xi^r$ is increasing at rate c in time. Term $W^+(d_j, \xi^r)$, on the other hand, can increase *at most* at rate c in time because of the following argumentation: For any sample path realization, the total costs $W^+(d_j, \xi^r)$ contain only a single cost term for a single match, given by structure $d_{j'} + c\left(d_{j'} - \xi^r\right)^+$ of some future driver j'. Thus, the maximum rate by which $W^+(d_j, \xi^r)$ can decrease in ξ^r, i.e., increase in time, is c, and condition $d_j + c\left(d_j - \xi^r\right)^+ > W^+(d_j, \xi^r)$ will always stay true. Hence, it is never optimal to match the order with this driver after the edge expiration time. From the fact that the optimal policy never matches an existing driver with an expired edge, we conclude that $W^+(d_j, \xi^r)$, at the time an edge expires or a driver with an expired edge arrives, is independent of d_j, i.e., $W^+(d^1, \xi^r) = W^+(d^2, \xi^r)$ for any two values $d^1 > \xi^r, d^2 > \xi^r$. Given this result, we can write without loss of generality $W^+(d_j, \xi^r) = W^+(\infty, \xi^r)$ for any $d_j > \xi^r$.

The decision given by expression $\min\left\{d_j + c\left[d_j - \xi^r\right]^+, W^+(\infty, \xi^r)\right\}$ has a threshold structure because the first term is obviously increasing in d_j while the second term is independent of d_j, such that both

terms can become equal at *at most* a single point in time. From the above argument also follows that case (b) in which $d_j < d_{j'}$ will never have a match as optimal decision.

Following this structure, and because we focus on expiring/expired edges for which $d_j \geq \xi^r$, the threshold can be derived from the following equation $d_j + c(d_j - \xi^r) \stackrel{!}{=} W^+(\infty, \xi^r)$. Equation (6) and the result of the theorem then follow directly as a consequence by setting

$$\tilde{d}(\xi^r) = \frac{W^+(\infty, \xi^r) + c\xi^r}{c+1}. \tag{26}$$

Part (b): From Equation (26) we observe that threshold function $\tilde{d}(\xi^r)$ is proportional to the sum of $W^+(\infty, \xi^r)$ and $c\xi^r$. In the proof of part (a) of this theorem, we have shown that $W^+(\infty, \xi^r)$ is independent of d_j and that it cannot decrease in ξ^r at a rate greater than c. Term $c\xi^r$ is obviously increasing in ξ^r at rate c. Hence, threshold function $\tilde{d}(\xi^r)$ as being proportional to the sum of $W^+(\infty, \xi^r)$ and $c\xi^r$ is non-decreasing in ξ^r. ■

Proof of Proposition 3. The pdf of a hypoexponential distribution is $\tilde{f}_k(x) = \sum_{j=k}^{\tilde{k}} \ell_j(0)\mu_j e^{-\mu_j x}$, where $\ell_j(0)$ is the Lagrange basis polynomial associated with point μ_j. It follows directly from Equation (2) that total costs are equivalent to

$$\begin{aligned}
\widetilde{W}^{\Omega^{k,\tilde{d}(\cdot)}}(k) &= \overline{d} + \int_0^{\overline{\xi}-\overline{d}} (\overline{\xi} - x - \overline{d}) \cdot \tilde{f}_k(x)dx + c \cdot \int_{\overline{\xi}-\overline{d}}^{\infty} (x + \overline{d} - \overline{\xi}) \cdot \tilde{f}_k(x)dx \\
&= \overline{d} + \int_0^{+\infty} (\overline{\xi} - x - \overline{d}) \cdot \tilde{f}_k(x)dx + (1+c) \int_{\overline{\xi}-\overline{d}}^{\infty} (x + \overline{d} - \overline{\xi}) \cdot \tilde{f}_k(x)dx \\
&= \overline{d} + \int_0^{+\infty} (\overline{\xi} - \overline{d}) \cdot \tilde{f}_k(x)dx - \int_0^{+\infty} x\tilde{f}_k(x)dx + (1+c) \int_{\overline{\xi}-\overline{d}}^{\infty} (x + \overline{d} - \overline{\xi}) \cdot \tilde{f}_k(x)dx \\
&= \overline{\xi} - \sum_{j=k}^{\tilde{k}} \mu_j^{-1} + (1+c)p(k)
\end{aligned} \tag{27}$$

with

$$p(k) = \int_{\overline{\xi}-\overline{d}}^{\infty} (x + \overline{d} - \overline{\xi}) \cdot \sum_{j=k}^{\tilde{k}} \mu_j e^{-x\mu_j} \left(\prod_{i=k, i\neq j}^{\tilde{k}} \frac{\mu_i}{\mu_i - \mu_j} \right) dx = \sum_{j=k}^{\tilde{k}} \frac{e^{-\mu_j(\overline{\xi}-\overline{d})}}{\mu_j} \left(\prod_{i=k, i\neq j}^{\tilde{k}} \frac{\mu_i}{\mu_i - \mu_j} \right). \tag{28}$$

Next, we transform expression $p(k)$ to allow us to evaluate it easily for varying values of k. We denote each element of the sum (on the right-hand side) in $p(k)$ as ϕ_j. Then, setting $j = k + m (0 \leq m \leq \tilde{k} - k)$, we obtain

$$\phi_j = \phi_{k+m} = \frac{e^{-\mu_{k+m}(\overline{\xi}-\overline{d})}}{\mu_{k+m}} \left(\prod_{i=k, i\neq k+m}^{\tilde{k}} \frac{\mu_i}{\mu_i - \mu_{k+m}} \right). \tag{29}$$

Given that $\mu_j = j\theta$, the ratio of ϕ_{k+m} and ϕ_{k+m+1} can be written as

$$\frac{\phi_{k+m}}{\phi_{k+m+1}} = \frac{e^{-\mu_{k+m}(\overline{\xi}-\overline{d})}}{e^{-\mu_{k+m+1}(\overline{\xi}-\overline{d})}} \cdot \frac{\mu_{k+m+1}}{\mu_{k+m}} \cdot \frac{\prod_{i=k, i\neq k+m}^{\tilde{k}} \frac{\mu_i}{\mu_i - \mu_{k+m}}}{\prod_{i=k, i\neq k+m+1}^{\tilde{k}} \frac{\mu_i}{\mu_i - \mu_{k+m+1}}} = (-1) \cdot e^{\theta(\overline{\xi}-\overline{d})} \left(\frac{k+m+1}{k+m} \right)^2 \frac{m+1}{D-(k+m)}, \tag{30}$$

which enables us to get

$$\frac{\phi_k}{\phi_{k+m}} = \frac{\phi_k}{\phi_{k+1}} \cdot \frac{\phi_{k+1}}{\phi_{k+2}} \cdots \frac{\phi_{k+m-2}}{\phi_{k+m-1}} \cdot \frac{\phi_{k+m-1}}{\phi_{k+m}} = (-1)^m \cdot e^{m\theta(\overline{\xi}-\overline{d})} \left(1+\frac{m}{k}\right)^2 \frac{m!(\tilde{k}-k-m)!}{(\tilde{k}-k)!}$$
$$= (-1)^m \cdot e^{m\theta(\overline{\xi}-\overline{d})} \left(1+\frac{m}{k}\right)^2 \frac{1}{\binom{\tilde{k}-k}{m}}, \tag{31}$$

expressing elements ϕ_{k+m} as the product of ϕ_k and a simple coefficient, finalizing the proof. ■

Proof of Proposition 4 **Part a (Quasi-convexity):** For given $(\overline{d}, \tilde{k}, \theta)$, the random variable (RV) $X(k)$ is given as the sum of independent exponential RV and can by written as $X(k) = \sum_{j=k}^{\tilde{k}} X_j$, with X_j as an exponentially distributed RV with rate $j\theta$. Note that $X(k+1) - X(k) = -X_k$, $X_j > 0$ for any j, and $X_j >_{st} X_{j+1}$ for any j (see also Shaked and Shanthikumar (1988)). Total costs can be written as a function of random variable $X(k)$ in the following way:

$$\overline{\xi} - X(k) + (1+c)[X(k) - (\overline{\xi} - \overline{d})]^+. \tag{32}$$

The n-step difference function at point k of the above cost term, which we denote by $\Delta W(k,n)$, is

$$\begin{aligned}
\Delta W(k,n) &= \overline{\xi} - X(k+n) + (1+c)[X(k+n) - (\overline{\xi} - \overline{d})]^+ - \left(\overline{\xi} - X(k) + (1+c)[X(k) - (\overline{\xi} - \overline{d})]^+\right) \\
&= X(k) - X(k+n) + (1+c)\left([X(k+n) - (\overline{\xi} - \overline{d})]^+ - [X(k) - (\overline{\xi} - \overline{d})]^+\right) \\
&= \left(\sum_{j=0}^{n-1} X_{k+j}\right) + (1+c)\left(\left[X(k) - (\overline{\xi} - \overline{d}) - \left(\sum_{j=0}^{n-1} X_{k+j}\right)\right]^+ - [X(k) - (\overline{\xi} - \overline{d})]^+\right) \\
&= Y(k,n) + (1+c)\left([X(k) - (\overline{\xi} - \overline{d}) - Y(k,n)]^+ - [X(k) - (\overline{\xi} - \overline{d})]^+\right),
\end{aligned} \tag{33}$$

where we set $Y(k,n) =_{st} \left(\sum_{j=0}^{n-1} X_{k+j}\right)$ for notational convenience. Note that $\Delta W(k,0) = 0 + (1 + c)\left([X(k) - (\overline{\xi} - \overline{d}) - 0]^+ - [X(k) - (\overline{\xi} - \overline{d})]^+\right) = 0$ and that $Y(k,n+1) >_{st} Y(k,n)$ for all k and n. Function $\Delta W(k,n)$ is stochastically convex in $Y(k,n)$ because it is composed of the sum of a linear term and a convex function $[\cdot]^+$.

Let us assume that total costs in Equation (32) stochastically (non-strictly) increase at some k', i.e., that $\Delta W(k',1) = \Delta W(k',1) - \Delta W(k',0) \geq_{st} 0$. Then, because function $\Delta W(k,n)$ is stochastically convex in $Y(k,n) = \left(\sum_{j=0}^{n-1} X_{k+j}\right)$ and this term is stochastically increasing in n, it holds that $\Delta W(k',n) \geq_{st} 0$ for all $n \geq 1$, implying that the total costs function in Equation (32) for no $k > k'$ drops below its value at k'. The immediate consequence is that it is non-decreasing after its first increase, which also holds for the expectation and establishes quasi-convexity of $\widetilde{W}^{\Omega^{k,\breve{d}(\cdot)}}$ in k.

Part b (Submodularity): For given $(\overline{d}, \tilde{k}, \theta)$, as shown in the proof of part (a) of Proposition 4, term $\mathbb{E}[X(k) - (\overline{\xi} - \overline{d})]^+$ is non-increasing in k. Hence, for the difference function holds $(1+c)\mathbb{E}[X(k+1) - (\overline{\xi} - \overline{d})]^+ - (1+c)\mathbb{E}[X(k) - (\overline{\xi} - \overline{d})]^+ \leq 0$. Taking the derivative of the difference function in c shows the submodularity of the total costs in k and c. Moreover, given that set $k \in [1, \tilde{k}]$ is a sublattice and if we further assume that the domain of c is a poset, Theorem 6.1 in Topkis (1978) applies: Since $\widetilde{W}^{\Omega^{k,\breve{d}(\cdot)}}$ is submodular in k and c, the optimal $k^*(c) \in \kappa^*$ is ascending in parameter c. ■

7.4. Model and policy extensions

In this appendix, we study endogenous FPT, driver-type dependent compensation schemes, variable driver speed, clustering, driver abandonment, and a socially friendly adaptation of the BKT policy that limits the idle times of drivers.

7.4.1. Endogenous FPT ξ_i: In practice, the congestion level of a restaurant may affect food processing times and customer behavior. In the main model that we describe in Section 3, we assume that neither the distribution of the FPT nor the arrival rate of orders to a certain restaurant are affected endogenously by the system state. In this subsection, we extended our model to allow the FPT and the arrival rate of orders to be affected by the congestion level of the restaurant. If congestion is high, food processing may take longer. In addition, the customer who observes the congestion may cancel the order and search for alternatives. Therefore, we integrate the real-time congestion level of a restaurant in the endogenous FPT prediction. To this end, we modify the simulation procedure with the following changes: We introduce a state variable 'number of orders waiting-to-be-fetched' for each restaurant to indicate its real time congestion level, that we classify as low, medium or high for simplicity. For example, at 15 orders or above (3 orders or below), we define the congestion level as high (low). Then, we adjust the arrival rate of the restaurant and/or its FPT distribution to this congestion level. We implement this situation in the simulation by advancing the next order arrival for this restaurant by 5 minutes if the congestion is low, and by postponing it by 5 minutes if the congestion is high. In the same sense, we also stretch (advance) the FPT by 10% of its predicted value if the congestion of the restaurant is high (low).

The left-hand side of Figure 10 shows the costs of the BKT policy for different values of k and buffer level under the modified arrival rate, and the right-hand side of Figure 10 indicates the costs of the BKT policy for the case of modified FPT with $\breve{d}^* = 300$. In both cases, we observe the same quasi-convex structure that we observed in the original model. For the left-hand side, we obtain minimal cost $W^{\Omega^{k^*,\breve{d}^*,\hat{P}^*}} = 699$ at $k^* = 8$ and $\hat{P}^* = 0.6$, which is slightly higher (+10.3%) than the costs in the original model. For the right-hand side, we obtain minimal cost $W^{\Omega^{k^*,\breve{d}^*,\hat{P}^*}} = 684$ at $k^* = 5$ and $\hat{P}^* = 0.4$, which is +7.9% higher than the costs of the original model. It is intuitive that costs increase in these scenarios, as the congestion level introduces additional uncertainty to the model.

7.4.2. Driver-type dependent compensation schemes: In practice, the unit fetch cost may depend on the type of driver (inhouse or crowdsourced) and the compensation scheme affect the matching decisions (*which type to match*). In this subsection, we extend the (B)KT policy to consider driver-type dependent compensation. To this end, we introduce two separate driver-reward functions $r_{in}(x)$ and $r_{cs}(x)$ for inhouse and crowdsourced drivers, respectively, with $x = d_{ij}(t_{ij}) + (\tau_i + \xi_i - t_{ij} - d_{ij}(t_{ij}))^+$. Reward function $r_{in}(\cdot)$ includes a fixed commission and a variable payment for each order; $r_{cs}(\cdot)$ only has

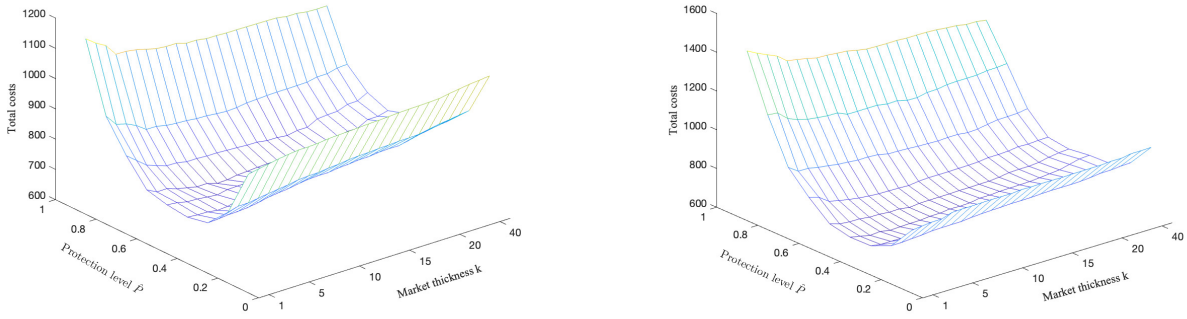

Figure 10 **Total costs under BKT policy as a function of thickening and protection levels for endogenously affected arrival rate (left-hand side) and FPT (right-hand side).**

a variable compensation for completed orders. Specifically, we set the fixed commission to 300, which is around $300/3600 \cdot 20 = 1.67$ yuan per order. We assume that the unit variable cost paid to a crowdsourced driver is +50% higher than that of an inhouse driver. We obtain $r_{in}(x) = 300 + x$ and $r_{cs}(x) = 1.5 \cdot x$.

We take the cost differences between both driver types into account in decision making by applying a transformation of the edge-expiration-timing of crowdsourced drivers. For any order i, given the reward functions, we redefine the edge expiration time of a crowdsourced driver to the time when the matching of an inhouse driver leads to the same cost as the matching of the crowdsourced driver at the edge expiration time, $\dot{t}_{ij}$, i.e, $r_{in}(\dot{t}_{ij}) = r_{cs}(t_{ij})$. We then adapt the BKT policy to take into account the re-defined edge times. For example, with our numbers, the redefined edge expiration time of a crowdsourced driver is $\dot{t}_{ij} = r_{in}^{-1}[r_{cs}(t_{ij})] = 1.5t_{ij} - 300$ if $t_{ij} \geq 600$ (which ensures $r_{in}(600) = r_{cs}(600)$) otherwise, $\dot{t}_{ij} = t_{ij}$. The crowdsourced driver will then be checked whether to match at $\dot{t}_{ij}$ time units before the end of the food processing. If matched, the order and the crowdsourced driver exit the system together. If not matched, the driver becomes inactive until t_{ij} time units before the FPT, after which, she can be used for delayed matching and incur penalty cost. Note that even if matched at t_{ij}, the order is also delayed but with zero penalty cost upon this moment.

Next, we test the adapted BKT policy for different values of variable commission of crowdsourced drivers and different values of fixed commission for inhouse drivers. In Table 6, we see that total costs increase in the *variable* commission of crowdsourced drivers. We also find that the share of matched inhouse drivers slightly increases, as the matching of crowdsourced drivers becomes more expensive. The results in Table 6 further show that total costs increase in the *fixed* commission for inhouse drivers. This time, the share of matched inhouse drivers slightly decreases due to the increased cost for this driver type.

7.4.3. Variable driver speed: Liu et al. (2021b) found that drivers may speed up to reach the restaurant on time and avoid overtime penalties. Next, we study how such behavior would affect our

Table 6 Results of adapted BKT policy $W^{\Omega^{k^*,\breve{d}^*,\hat{P}^*}}$ by variable commission for crowdsourced drivers.

Commission	Variable						Fixed					
Parameter	1x	1.2x	1.4x	1.6x	1.8x	2.0x	100	200	300	400	500	600
Total costs $W^{\Omega^{k^*,\breve{d}^*,\hat{P}^*}}$	904	944	982	1031	1066	1110	765	836	904	972	1040	1108
Share of inhouse drivers (%)	68.13	68.27	68.35	68.40	68.67	68.68	68.50	68.13	68.13	68.13	68.13	68.13

results by modifying our model: With probability p_{vd}, a delay event happens that threatens the timely arrival of a driver at the restaurant. In the simulation, if an order is matched after the edge expired, it is automatically treated as potentially delayed. In this case, the platform offers a fixed compensation of c_{up} if the driver speeds up, and the driver arrives on time with probability p_{vt}. In this case, the platform pays c_{up}. If the driver does not arrive on time, the compensation is not paid out.

We numerically analyze the results of this scenario. We set $p_{vd} = 0.05$, $c_{up} = 600$ ($\sim$3 yuan), and $p_{vt} = 0.5$, and vary the probability values. The results are shown in Table 7. We find that the total costs are positively increasing in the delay probability p_{vd}, indicating the necessity for the planner to take into account potential delay events. We also find that total costs are decreasing in p_{vt}, indicating that less reliability leads to higher costs. We conclude that incentives to motivate drivers to speed up may reduce overtime penalties.

Table 7 Results for BKT policy $W^{\Omega^{k^*,\breve{d}^*,\hat{P}^*}}$ as a function of probability p_{vd} and probability p_{vt}.

Delay probability p_{vd}	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
Total costs $W^{\Omega^{k^*,\breve{d}^*,\hat{P}^*}}$	638	642	645	650	654	657	659	662	664	668
On-time probability p_{vt}	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Total costs $W^{\Omega^{k^*,\breve{d}^*,\hat{P}^*}}$	669	665	662	657	654	652	649	646	642	638

7.4.4. Regional clustering: In Section 3, we defined the neighborhood of orders as the set of all available drivers. But Li and Netessine (2020) found that redundant market thickness may lead to search friction and consequently to a low matching rate. This motivated us to test the effect of our neighborhood assumption by analyzing *clustering*.

To this end, we divide the selected service region into n_c fixed *clusters*. Table 8 shows the results for $n_c \in \{1, 2, 4, 8\}$ and $\breve{d} = 300$, where $n_c = 1$ is the model that we studied in the main part of the paper. We report the percentage gap against this model, $W^{\Omega^{k^*,\breve{d},\hat{P}^*}}(n_c = 2) - W^{\Omega^{k^*,\breve{d},\hat{P}^*}}(n_c = 1))/W^{\Omega^{k^*,\breve{d},\hat{P}^*}}(n_c = 1) \times 100$. The tests results indicate that clustering of a region may be beneficial, in particular with respect to certain performance measurements. For example, for $n_c = 2$, both the expected delay per order V_d^{Ω} (-5%) and the overall service level (+5%) have been improved, which might be due to reduced search friction, as discussed above. If we continue to divide the region into more clusters ($n_c = 4$ and $n_c = 8$), the total costs increase because the average delay time V_w^{Ω} is seriously increased compared to the case $n_c = 2$. This reveals that, by further dividing the region into more clusters, the clustering restricts the availability of drivers and leads to a thin marketplace.

Intuitively, each sub-region may have different optimal parameters for the (B)KT policy. We show that for case $n_c = 4$ in Figure 11, the quasi-convex structure of the total costs that we showed in Subsection

Table 8 Percentage gap of total costs for $n_c \in \{1,2,4,8\}$ clusters.

n_c	Total Costs $W^{\Omega^{k^*,d,\hat{P}^*}}$	Drive-to-Shop V_p^Ω	Delay V_d^Ω	Service Level V_l^Ω (%)	Engagement Time V_e^Ω	Engaged Drivers V_n^Ω	Waiting Time V_w^Ω	Idle Time V_q^Ω	Net-income V_s^Ω
$n_c = 1$	634	110	39	64.42	1201	92	291	1522	8.7
$n_c = 2$	662	118	37	68.90	1242	95	324	1883	8.2
$n_c = 4$	735	112	58	62.68	1188	91	277	2661	6.3
$n_c = 8$	762	127	59	62.42	1208	93	281	1473	10.8

All time values in seconds.

4.3 for the simplified model can also be observed for each sub-region, but the optimal values of k^* and $\hat{P}^*$ differ between sub-regions (e.g., given $\breve{d} = 300$, $k^* = 4$ and $\hat{P}^* = 0.3$ for sub-region 1 while $k^* = 2$ and $\hat{P}^* = 0.2$ for sub-region 4). Our analysis shows that clustering may lead to certain benefits (in terms of service level) and optimal partitioning of clusters (e.g., Carlsson et al. 2021) remains an important research problem for on-time last-mile delivery.

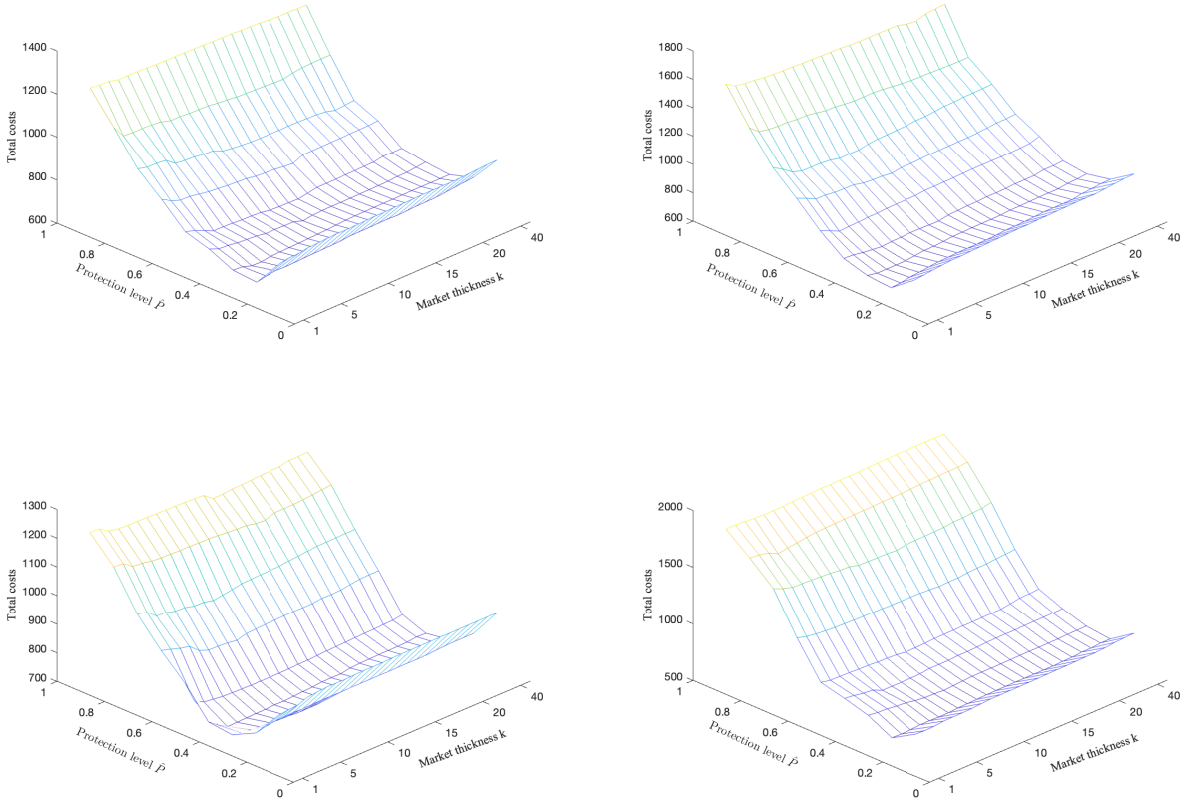

Figure 11 Total costs under BKT policy as a function of thickening and protection levels under $n_c = 4$ subregions.

7.4.5. Socially friendly BKT policy: The results of Section 5 indicate that the (B)KT policy effectively reduces the platform's cost by reducing the drivers total engagement time. However, the objectives of platform and of drivers are conflicting, and this effect leads to a decrease in expected net-income of the drivers, because of the longer (non-paid) idle time of drivers. In this subsection, we therefore propose a *socially friendly* adaptation of the (B)KT policy, by limiting the length of idle times for drivers. If a driver exceeds a threshold y of idle time, it receives a compensation c_q and leaves the platform (e.g., transfers to another platform or exits for a break) without matching.

We perform a numerical experiment to study the impact of this adaptation of the BKT policy. We set the threshold $y \in \{30, 45, 60, 75, 90, 120, 180, 240, 300, 360, \infty\}$ minutes (the last case corresponds to the model in the main part of the paper) and the compensation to $c_q = 900 \cdot y/60$ (5 yuan per hour). Figure 12 shows total costs of the platform and driver net income as a function of y. Both, total costs as well as driver income, decrease with greater y, indicating the inherent conflict between the interests of the platform and those of the drivers (see also Meituan 2021). The platform has to give up on part of its profit to maintain a higher driver income. Our modified BKT policy allows the platform to reduce the burden of overly long idle times of drivers and, by appropriately setting threshold y, to find a balance for $W^{\Omega^{k^*,\check{d}^*,\hat{P}^*}}$ and V_s^{Ω} that corresponds to its driver availability and company values.

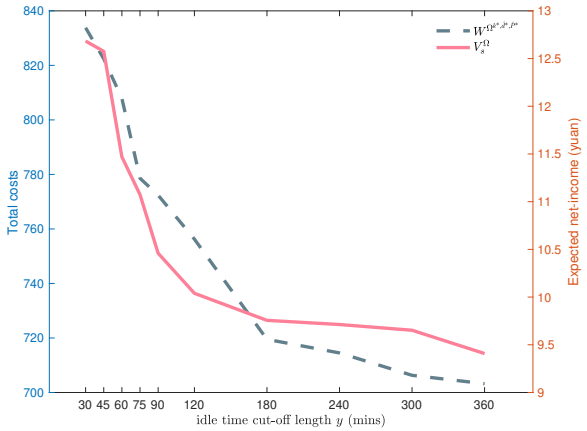

Figure 12 **Platform's total costs and driver's expected net-income as function of threshold y.**

7.4.6. Abandonment rate of drivers: The assumption regarding driver abandonment is crucial for ensuring the long-term stability of the matching system. However, the abandonment rate, λ_a (i.e., the number of drivers abandoning per hour), also impacts the availability of circulating drivers within the system, consequently affecting both the driving time and total costs. We perform additional numerical tests to further investigate the influence of this parameter on the results.

Table 9 **Results for BKT policy $W^{\Omega^{k^*,\check{d}^*,\hat{P}^*}}$ and V_p^{Ω} as a function of abandonment rate λ_a.**

Abandonment rate λ_a	0	1/20	1/10	1/5	1/2	1	5	10	20
Total costs $W^{\Omega^{k^*,\check{d}^*,\hat{P}^*}}$	634	634	634	630	647	691	913	1017	1153
Drive-to-shop V_p^{Ω}	110	110	110	106	108	116	154	191	211

In Table 9 we show the total costs per order and the average drive-to-shop time under different values of abandonment rate λ_a. A discernible trend becomes evident: as λ_a increases, both total costs, $W^{\Omega^{k^*,\check{d}^*,\hat{P}^*}}$, and drive-to-shop time, V_p^{Ω}, increase. Total costs increase significantly between $\lambda_a = 0$ and $\lambda_a = 20$, starting at 634 and ending at 1153 (+82%). This trend suggests that higher abandonment rates lead to higher costs (under the BKT policy), which is a consequence of the reduced availability of eligible drivers in the system. Interestingly, a slight decrease in $W^{\Omega^{k^*,\check{d}^*,\hat{P}^*}}$ is observed when λ_a increases from 1/10 to

1/5, likely caused by reduced search friction. This observation may hint at the existence of an optimal abandonment rate where the total costs $W^{\Omega^{k^*,\breve{d}^*,\hat{P}^*}}$ are minimized.

7.5. Solving the Example 1 with KT policy

In this subsection, we apply the KT policy ($k=1$ and $\breve{d}(\cdot)=2$) to Example 1 (see Section 4). In Figure 13, on the left-hand side, we observe that at $t=1$, the connection between Driver A and Order 1 is about to expire. However, with the arrival of Driver B, we know that there are two compatible drivers for Order 1. Consequently, we do not match Driver A at this moment. At time $t=2$, Driver B's availability expires, marking the last compatible driver for Order 1. Consequently, we match Order 1 with Driver B at this time. Simultaneously, Order 2 enters the system at time $t=2$. At $t=4$, the connection between Driver A and Order 2 is about to expire, leading us to match both. In this scenario, Driver B and Order 1 have exited the system, and we represent this in the right-hand side of Figure 13 (shadowed area). We conclude that the KT policy takes the same decisions as the optimal policy presented in Figure 3.

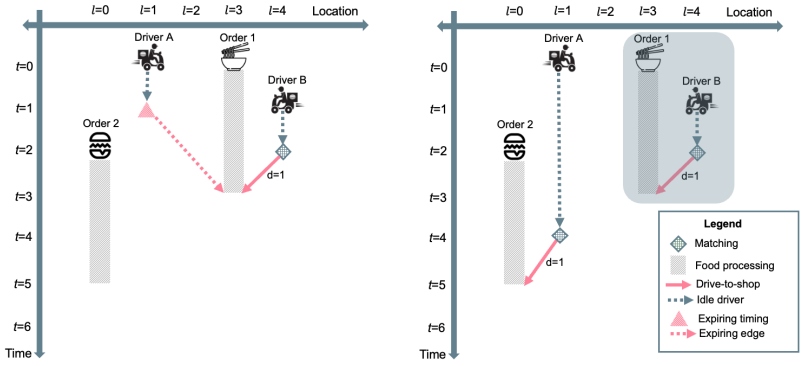

Figure 13 **Illustration of the KT policy applied to Example 1.**

7.6. Batching Policy and Results

The *Batching* policy (denoted by Ω^b) waits ω time units before solving a linear program to match the open orders with available drivers in a cost-minimal way while the unmatched orders and drivers are kept for the next stage optimization. We first present the required linear programming model for each stage computation as follows. Suppose the current time is t, which a sum of many ω time units. We denote $\mathcal{I}_t$ ($\mathcal{J}_t$) to represent the unmatched orders and driver within the interval $(t-\omega, t]$. Let the decision variables x_{ijt} indicate that an order i is matched to a driver j at time t. The minimal cost of such a problem could be obtained by solving another deterministic linear programming problem:

$$W^{\Omega_t^b} = \min \sum_{i\in\mathcal{I}_t}\sum_{j\in\mathcal{J}_t} x_{ijt}\left[d_{ij}(t)+(\tau_i+\xi_i-t-d_{ij}(t))^+ + c\cdot(t+d_{ij}(t)-\tau_i-\xi_i)^+\right] \tag{34}$$

$$\sum_{j\in\mathcal{J}_t} x_{ijt}=1, \forall i\in\mathcal{I}_t; \sum_{i\in\mathcal{I}_t} x_{ijt}\le 1, \forall j\in\mathcal{J}_t \tag{35}$$

$$x_{ijt}\in\{0,1\}, \forall i\in\mathcal{I}_t, j\in\mathcal{J}_t \tag{36}$$

Constraints (35) enforce that each order is served and prohibit that a driver serves more than one order at a time. However, in practice, if the exact FPT are unknown, we will replace the ξ with the predicted values $\hat{\xi}$ and solve the program. The above procedure will repeat many times until the end of horizon with an incremental ω on time counter. Once we get the matching scheme under predicted FPT, at the end, we need to recompute the total costs per order as the equation (34) for real FPT ξ. Finally, we present the simulation results under Ω^b policy given the exact FPT ξ and predicted FPT $\hat{\xi}$ of real data in the following table, i.e., $W_{\xi}^{\Omega^b}$ and $W_{\hat{\xi}}^{\Omega^b}$, varying in different batching intervals ω (measured by minutes).

Table 10 Simulation results for batching policy Ω^b given exact FPT ξ and predicted FPT $\hat{\xi}$.

ω (mins.)	1	2	3	4	5	6	7	8	9	10
$W_{\xi}^{\Omega^b}$	1228.87	1198.84	1168.13	1138.73	1107.53	1080.42	1049.38	1014.23	983.42	961.23
$W_{\hat{\xi}}^{\Omega^b}$	1352.45	1330.35	1314.37	1288.01	1273.53	1267.32	1249.64	1222.55	1231.11	1216.98
ω (mins.)	11	12	13	14	15	16	17	18	19	20
$W_{\xi}^{\Omega^b}$	931.23	904.80	883.19	854.49	839.02	814.02	818.74	824.78	830.46	825.00
$W_{\hat{\xi}}^{\Omega^b}$	1199.02	1200.96	1189.76	1204.50	1180.04	1208.84	1242.23	1237.84	1253.62	1249.74

7.7. Computation of Threshold Function for the Single-Order Case

Next, we describe how to compute the threshold function in the single order case, as given by Equation (8) in the main manuscript. For the computation, we have to transform the infinite-horizon problem into a finite-horizon problem, which we can then solve iteratively through Equation (25) in Appendix 7.3.

Without loss of generality, we set the order's arrival time to $\tau_i = 0$, such that $\xi^r = \xi$ (we omit i in notations for simplicity), where ξ is the FPT of this order. For the finite horizon counterpart, let us define the time horizon $\mathcal{T}$, which is a positive value as defined in the full-information case. If the system terminates in time $\mathcal{T}$, it means that, in this case, the remaining FPT turns into $\xi^r = \xi - \mathcal{T}$ and the final state is denoted by $\mathbf{S}_n = (-, \xi - \mathcal{T})$. For the termination state, a penalty cost $W^+(-, \xi - \mathcal{T}) = c \cdot (\mathcal{T} - \xi)$ is incurred, corresponding to the delay if the order is unmatched at the end of the horizon (similar to the full-information case of Equation (3)).

To compute the cost function, we discretize time into periods of length Γ (e.g., 1 minute in our case). Furthermore, we need to discuss different scenarios of new driver arrival. For the next driver j' with driving time $d_{j'}$, we assume an exponential inter-arrival time, $\hat{t}_{j'}$. For computational simplicity, we also set a maximum inter-arrival time for any new driver, $\hat{t}^{max}$. In addition, we assume that travel time for the new driver is also measured by the same time scale of length Γ, therefore, the ξ^r will change over the same timeline. As a consequence, the discretized state space can be fully enumerated.

With the finite horizon problem, we have to take into account the end of the system dynamics. That means, given a state (d_j, ξ^r), the possibility of event $\hat{t}_{j'} \geq \mathcal{T} - \xi + \xi^r$ occurs, which indicates that the next driver arrives after the system is terminated at time $\mathcal{T}$. In this case, this order is assumed to be unmatched at the end of the horizon and incurs a penalty cost $W^+(-, \xi - \mathcal{T}) = c \cdot (\mathcal{T} - \xi)$. The explicit treatment of

the terminate state guarantees the existence of a solution for the iterative procedure, provided that $\mathcal{T}$ is sufficiently large. In our numerical tests, we found that $\mathcal{T} = 10 \cdot \xi$ is already able to guarantee termination of our computational procedure. For the numerical experiments in our paper, we used the default values $\lambda_d = 0.4, c = 6, \Gamma = 1, \hat{t}^{max} = 50, \xi = 23, \mathcal{T} = 10 \cdot \xi$ and $\epsilon = 0.0001$.

7.8. Additional description of the data set

First, we analyze all orders ($\sim 120,000$) for a particular region and discuss general summary statistics. There are 289 restaurants providing service for customers in the region within this month. Among these restaurants, we report the information about the top 12 restaurants, see Table 11.

Table 11 Overall statistics for the 12 most popular restaurants (with monthly sold orders $\geq 2,000$).

Restaurant	number of orders	number of dishes per order	fetching congestion level	FPT (mins.)	shop-wait time (mins.)	go-to-shop time (mins.)	customer distance (m)
1	7575	1.17	12.18	15.81	8.29	7.05	1484
2	5444	1.01	10.46	29.67	9.64	19.56	1489
3	4103	1.06	6.34	18.14	8.84	8.85	1551
4	3462	1.11	3.78	15.83	8.68	6.71	1380
5	3223	1.29	10.43	22.17	7.76	13.94	1782
6	3033	1.05	6.00	21.58	8.32	12.79	1079
7	2915	1.08	5.60	18.31	7.50	10.33	1518
8	2702	1.10	3.72	17.78	6.97	10.31	1983
9	2393	1.07	4.99	27.92	8.02	19.43	1433
10	2370	1.08	3.14	20.93	9.65	10.80	1273
11	2193	1.08	2.49	14.93	6.12	8.33	1617
12	2055	1.05	5.16	28.26	8.04	19.71	1715
Average	3456	1.10	6.19	20.95	8.15	12.32	1525

The fetching congestion level (i.e., the average number of orders waiting to be fetched) is approx. six for the top restaurants, implying that these restaurants typically backlog orders and that assigned drivers may have to wait for the food to be ready. Mao et al. (2022) also mentioned that drivers wait on average 5.8 minutes for food processing in restaurants.

The arrival pattern of orders and drivers within a day are shown in Figure 14 (a), which has let us to define the *rush*, *no-rush* and *overall* scenarios. Figure 14 (b) shows the distribution of the number of finished tasks per driver, indicating that around 45% of drivers serve more than 20 orders per day, which is an indicator of driver income.

With the arrival time information, we can further investigate the arrival process of orders and drivers. As shown in Figure 15, the inter-arrival time of orders and drivers can be well approximated with an exponential distribution with rate 0.05 and 0.75 per second, respectively, which justifies the assumption of Poisson arrivals (i.e., with rates $\lambda_o = 3$ and $\lambda_d = 4$ per minute). Note that we only consider one-to-one matching in the model. Consequently, we have transformed orders of several items into separate deliveries.

We further provide the orders' delivery distances (i.e., distance between the restaurant and customer locations) in Figure 16 (a). Note that over 95% of orders are distributed within 3 km of the restaurant,

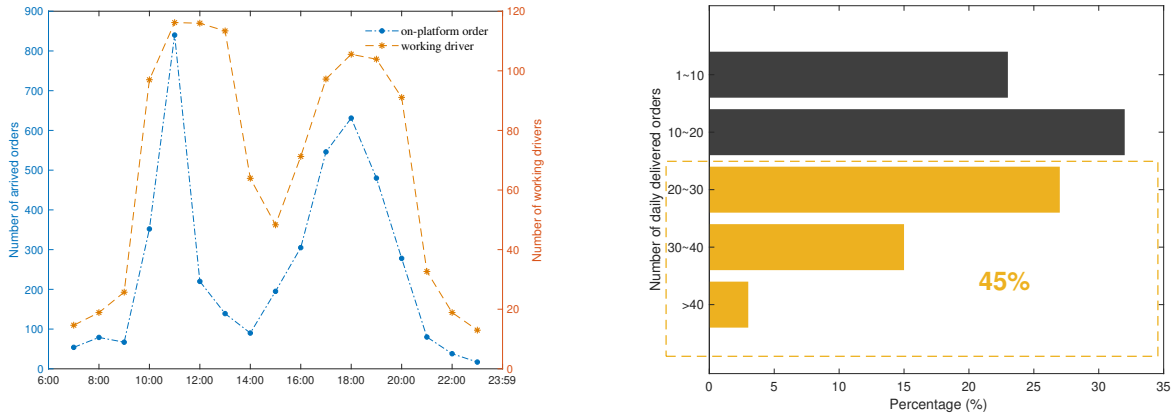

Figure 14 **Arrival pattern of on-demand delivery (left-hand side) and number of delivered orders per driver and day (right-hand side).**

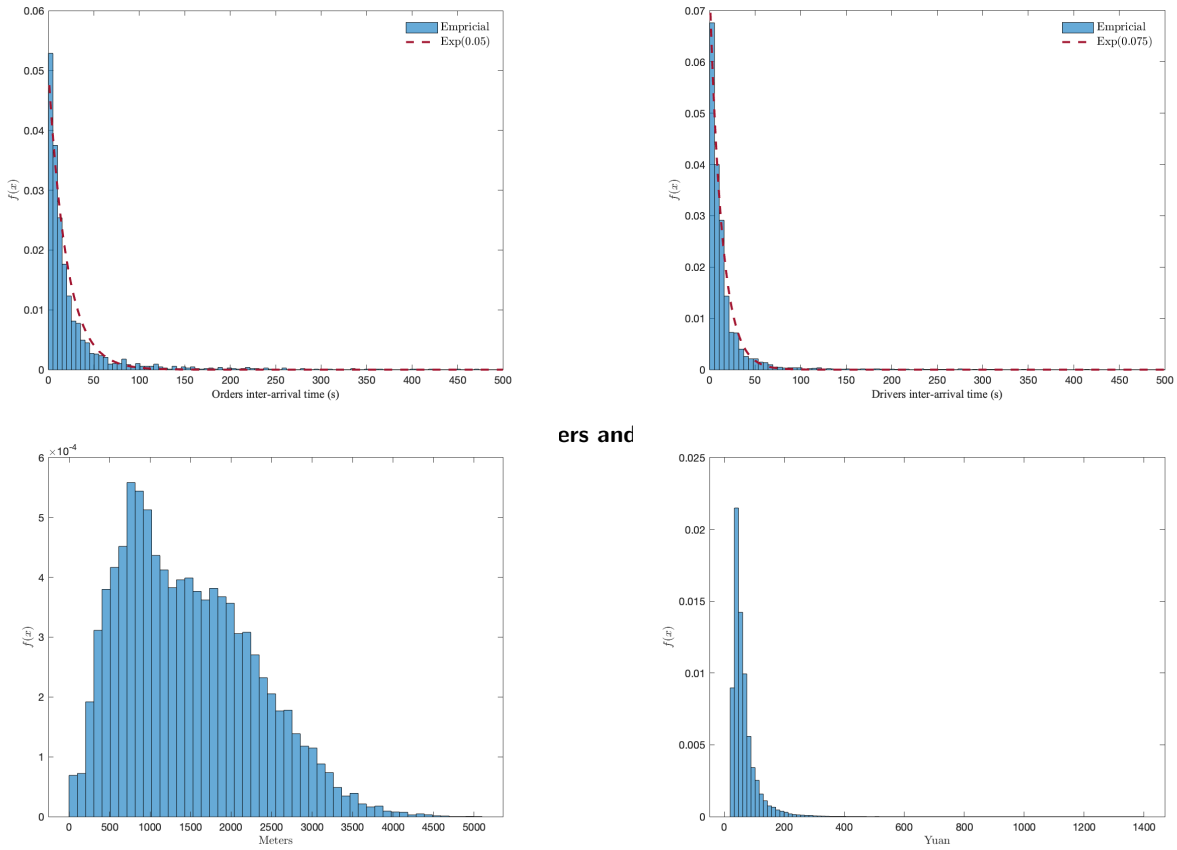

Figure 16 **Distribution of order distance (left) and order price value (right).**

which is consistent with Mao et al. (2022). Moreover, Figure 16 (b) indicates that the average food value of orders is $\sim$ 65 Yuan.

Finally, we provide more information about the FPT obtained from the data set analysis because they are key input for the KT policy design and implementation.

(a) *Distributional shape*. The distribution of FPT for the three different scenarios (rush, no-rush and overall) are shown in Figure 17. We observe that the distribution is more variable in the rush-scenario

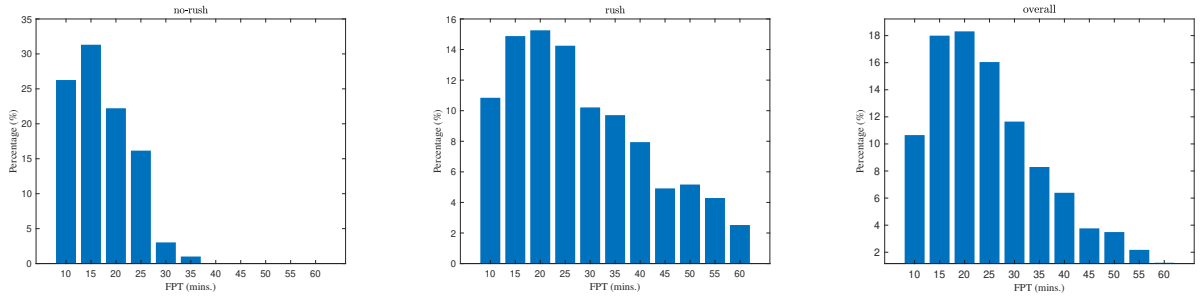

Figure 17 **The distribution of FPT values (in minutes) for each scenario.**

than in the non-rush scenario.

(b) *Weekdays vs. Weekends*. Then, we analyze the FPT values by the categories weekday and weekend. The results indicate that the average FPT values on weekdays are slightly higher than those on other days (by 11% on average).

(c) *Endogenous relationship between FPT and demand*. We investigate if demand is different than usual after periods with long FPT to check if the demand rate is endogenously related to the FPT. To this end, we decompose the working day into 30-minute-intervals. For each interval, we count the number of order arrivals and compute the average FPT for orders arrived in this interval. In the left Figure 18, we plot the relationship between the FPT and number of order arrivals. We find that both quantities are positively correlated ($R^2 = 0.55$), suggesting that customers were not averse to longer FPT and that their purchase intention is not discouraged by longer waiting. Furthermore, in the right Figure 18, we illustrate the correlation between the number of order arrivals in a 30-minute interval *following* the period for which we computed the average FPT (shown on the left y-axis) and the average FPT itself (on the right y-axis). Notably, we observe that during peak times (e.g., 10:00-12:00), there is a surge in order arrivals (indicating congestion) that corresponds with an increase in average FPT. However, despite this surge, there is not a significant drop in order arrivals during this period, as evidenced by the continuous upward trend (the pink curve) in 10:00-11:40. A similar pattern is evident during dinner hours (17:00-18:30). From these observations, we infer that a higher FPT does not deter customers from placing orders.

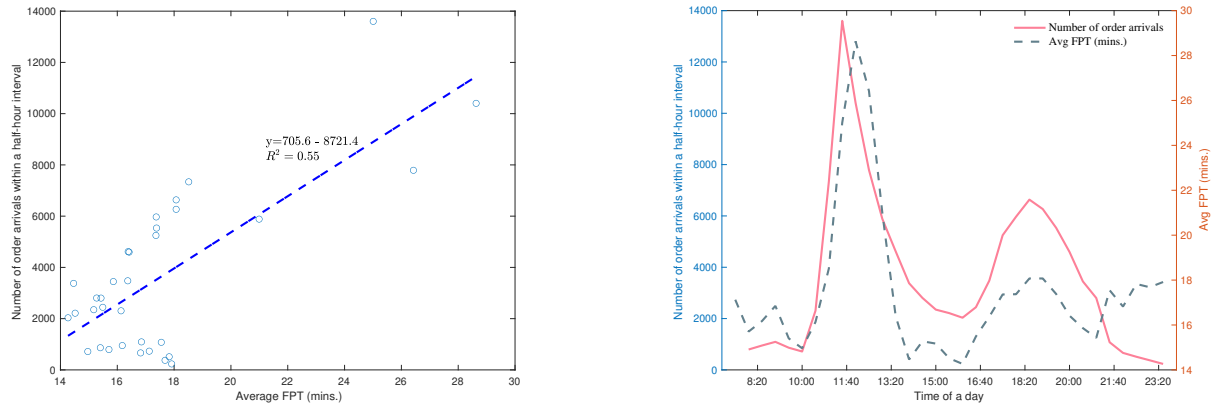

Figure 18 **The relationship between number of order arrivals per 30-minutes interval and FPT.**