CYBERNETICS AND PHYSICS, VOL. 14, NO. 3, 2025, 239–247

ONLINE MEAL DELIVERY SCHEDULING WITH ORDER AGGREGATION AND MACHINE LEARNING-GUIDED COURIER ASSIGNMENT

Paul Kondratov
TBank, Russia
pkondratov01@gmail.com

Elizaveta Tarasova
Saint Petersburg State Universite
Saint Petersburg, Russia
elizaveta.tarasova@spbu.ru

Olga Granichina
Herzen State Pedagogical University
Saint Petersburg, Russia
olga_granichina@mail.ru

Article history:
Received 10.11.2025, Accepted 27.11.2025

Abstract

The present work addresses the scheduling optimization problem in the context of online food delivery systems, focusing on strategic order aggregation and courier batch assignment under typical operational constraints. By representing the process as a variant of the Online Meal Delivery Scheduling Problem (MDSP), we analyze order characteristics, courier resources, and system objectives using classical scheduling theory terminology. Clustering orders via K-means and DBSCAN is formalized as batch scheduling for minimization of the makespan and total tardiness. Competitive analysis is applied to benchmark heuristic algorithms, with theoretical and empirical evidence supporting 21–42% improvement in total delivery time and 13–28% reduction in courier workload. Machine learning clustering integrates with online assignment to produce schedules approaching the lower bound of competitive ratio, ensuring resource-efficient, real-time order fulfillment.

Key words

meal delivery scheduling, batch scheduling, courier assignment, order aggregation, machine learning, makespan minimization, scheduling theory

1 Introduction

The rapid expansion of on-demand food delivery platforms has drastically increased the operational complexity of last-mile logistics. In major urban markets, the number of orders grows faster than the available courier workforce, leading to rising delivery times, unstable service quality, and significant operational inefficiencies [Marketing, 2023; Editorial, 2022; Data, 2023]. As a result, delivery platforms must solve a high-frequency online scheduling problem: dynamically assign couriers to geographically distributed, time-sensitive customer requests while accounting for restaurant preparation delays, courier availability, travel distances, and strict delivery deadlines.

From a scheduling-theoretic perspective, meal delivery can be viewed as a dynamic variant of the pickup-and-delivery problem with deadlines, combining elements of online scheduling, batching, and vehicle routing. Classical scheduling theory provides a formal basis for reasoning about lateness, tardiness, and makespan minimization [Graham et al., 1979; Lazarev and Gafarov, 2011]. However, unlike traditional machine scheduling settings, food delivery systems exhibit high spatiotemporal uncertainty, heterogeneous agent behavior, and strongly dynamic demand patterns, making purely deterministic or static planning approaches insufficient.

Recent research has focused on specialized formulations of the Meal Delivery Problem (MDP) and Meal Delivery Scheduling Problem (MDSP). Existing works include routing-based models [Reyes et al., 2018; Yildiz and Savelsbergh, 2019], dynamic pickup-and-delivery formulations with deadlines and random restaurant ready times [Ulmer et al., 2021], and optimization approaches for batching and assignment [Agnetis et al., 2023; Simoni and Winkenbach, 2023]. Heuristic courier dispatching and routing algorithms have also been proposed for high-frequency environments [Steever et al., 2019], while large-scale industrial platforms report the use of batching and predictive assignment rules [Khalkechev, 2020; Mao et al., 2019].

At the same time, machine learning (ML) methods have emerged as a promising tool for improving courier assignment decisions. Reinforcement-learning and predictive models have been applied to estimate delivery duration, prioritize orders, and guide real-time dispatch rules [Jahanshahi et al., 2022; Bozanta et al., 2022]. Yet, most existing ML-driven approaches focus either on routing or assignment in isolation, and few works systematically combine clustering-based order aggregation with ML-guided scheduling in a unified online framework.

This study addresses this gap by proposing an integrated approach to online meal delivery scheduling that combines (i) clustering-based order aggregation and (ii) machine learning-guided courier assignment. We model meal delivery as a dynamic scheduling problem with order arrivals over time and demonstrate that combining aggregation with predictive assignment significantly improves lateness, courier distance, and Pareto efficiency. Using both synthetic and real-world data, we show that the proposed approach consistently outperforms classical heuristics and non-aggregated baselines.

2 Related Work

Scheduling Theory Background. Classical deterministic scheduling problems have been extensively surveyed in [Graham et al., 1979] and formalized in standard monographs such as [Lazarev and Gafarov, 2011]. Core objectives relevant to delivery systems include the minimization of lateness, tardiness, or makespan, as well as multi-objective trade-offs [Lazarev et al., 2021]. Online and semi-online models provide a framework for dynamic job arrivals and partial information about future workloads [Dwibedy and Mohanty, 2022], although meal delivery systems typically operate under fully online uncertainty. Competitive-analysis perspectives on dynamic assignments with limited information have also been studied in [Zhang and Ye, 2002]. Recent work by [Khrustaleva et al., 2024] demonstrates the necessity of decomposing complex systems into manageable subsystems, enabling more robust and adaptable scheduling solutions. They show, that effectiveness of the optimization process is largely determined by its degree of detail and the appropriateness of the chosen objectives and control parameters.

Meal Delivery Scheduling and Routing. Recent years have seen a growing interest in optimization models tailored to on-demand food delivery. The Meal Delivery Routing Problem was formalized in [Reyes et al., 2018], while high-quality approximation and optimization methods for MDSP were proposed in [Yildiz and Savelsbergh, 2019] and [Ulmer et al., 2021]. Order aggregation and batching strategies have been shown to significantly reduce delivery distance and improve efficiency, as demonstrated in [Agnetis et al., 2023] and the crowdsourced delivery setting of [Simoni and Winkenbach, 2023]. Heuristic and dynamic routing algorithms for real-time courier dispatching were considered in [Steever et al., 2019]. Industrial perspectives on batching and assignment for large-scale systems, including algorithmic insights from Yandex.Eda and other platforms, are discussed in [Khalkechev, 2020; Mao et al., 2019].

Machine Learning for Courier Assignment. Machine learning has recently become an important component of courier dispatching. Reinforcement learning was applied to dynamic courier assignment and delivery-time prediction in [Jahanshahi et al., 2022], while hybrid RL and supervised-learning methods were explored in [Bozanta et al., 2022]. ML models have shown strong potential for reducing lateness by predicting delivery times, prioritizing orders, and guiding routing heuristics. However, existing studies largely treat assignment or routing independently, and relatively few works combine ML-based prioritization with clustering-driven order aggregation in a unified online scheduling framework.

Routing and Optimization Tools. Our method uses OR-Tools [AI, 2024], a widely-used combinatorial optimization library capable of solving vehicle-routing and pickup-and-delivery problems with time-window constraints. Such solvers provide robust and scalable route construction but require appropriate batching and prioritization strategies to achieve high performance in dynamic environments.

Overall, prior literature highlights the importance of batching, predictive assignment, and dynamic routing in meal delivery systems. Yet, an integrated approach combining clustering-based aggregation with ML-driven courier scheduling remains underexplored. This work aims to fill that gap.

3 Problem Formulation

The meal delivery scheduling problem addresses the challenge of dynamically assigning orders to couriers while optimizing delivery efficiency in an on-demand food delivery system. This problem can be classified as a dynamic scheduling problem with time constraints, combining elements of vehicle routing and task assignment under uncertainty.

Consider a meal delivery system operating over a time horizon with the following components. Let $J = \{1, 2, ..., n\}$ denote the set of orders arriving dynamically throughout the operational period. Each order $j \in J$ is characterized by:

r_j: creation time (when the customer places the order)
$p_{j,R}$: preparation time at restaurant R_j
d_j: delivery deadline (promised delivery time)
loc_{R_j}: restaurant location coordinates
loc_{C_j}: customer delivery location coordinates

Let $K = \{1, 2, ..., m\}$ represent the set of available couriers. Each courier $k \in K$ has:

$loc_{k,start}$: initial starting position
$speed_k$: movement speed
$stamina_{k,t}$: time-varying fatigue coefficient affecting speed

Let L denote the set of all possible locations in the delivery area, where $L = \{loc_{k,start}\}_{k\in K} \cup R_{set} \cup C_{set}$, with R_{set} being the set of restaurant locations and C_{set} the set of customer locations.

The travel time function $T(loc_a, loc_b, k, t)$ represents the time required for courier k to travel from location loc_a to location loc_b starting at time t:

$$T(loc_a, loc_b, k, t) = \frac{distance(loc_a, loc_b)}{speed_k \cdot stamina_{k,t} \cdot random_t},$$

where $stamina_{k,t}$ decays over time and $random_t$ introduces stochastic variation in travel conditions.

3.1 Decision Variables and Constraints

The primary decision variable is the binary assignment matrix x_{jk}:

$$x_{jk} = \begin{cases} 1, & \text{if order } j \text{ is assigned to courier } k \\ 0, & \text{otherwise} \end{cases}$$

Additional decision variables include:

$S_{j,prep}$: start time of order preparation at the restaurant
$C_{j,prep}$: completion time of order preparation
$A_{k,loc}$: arrival time of courier k at location loc
P_j: pickup time of order j by the assigned courier
C_j: delivery completion time of order j
$path_k = (loc_1, loc_2, ..., loc_p)$: route sequence for courier k

Constraint 1 (Assignment uniqueness): Each order must be assigned to exactly one courier: $\forall j \in J \quad \sum_{k \in K} x_{jk} = 1$.

Constraint 2 (Preparation timing): Restaurant preparation begins immediately upon order creation and respects preparation time requirements:

$$S_{j,prep} \geq r_j \quad \forall j \in J,$$
$$C_{j,prep} = S_{j,prep} + p_{j,R} \quad \forall j \in J.$$

Constraint 3 (Restaurant capacity): The number of orders being prepared simultaneously at restaurant r cannot exceed its capacity cap_r:

$$\sum_{j \in J: R_j = r} I_{j,prep}(t) \leq cap_r \quad \forall r \in R_{set}, \forall t,$$

where $I_{j,prep}(t) = 1$ if $t \in [S_{j,prep}, C_{j,prep}]$, and 0 otherwise.

Constraint 4 (Pickup timing): An order can only be picked up after preparation is complete and the courier has arrived at the restaurant:

$$P_j \geq \max(C_{j,prep}, A_{k,R_j}) \quad \forall j \in J, k \in K : x_{jk} = 1.$$

Constraint 5 (Delivery timing): The delivery time accounts for pickup time and travel time from the previous location on the courier's route:

$$C_j = P_j + T(loc_{prev}(C_j), loc_{C_j}, k, P_j) \quad \forall j \in J, k \in K : x_{jk} = 1,$$

where $loc_{prev}(C_j)$ denotes the location visited by courier k immediately before delivering order j.

3.2 Objective Functions

The problem involves two competing objectives that must be balanced:

1. Minimize total lateness:

$$L = \sum_{j \in J} \max(0, C_j - d_j) \to \min_{\{x_{jk}\}}. \quad (1)$$

2. Minimize total distance traveled:

$$Dist_{avg} = \frac{1}{|K|} \sum_{k \in K} Dist_k \to \min_{\{x_{jk}\}}, \quad (2)$$

where $Dist_k$ represents the total distance traveled by courier k along their assigned route.

These objectives are combined into a single normalized objective using weighted scalarization:

$$\min F(x) \quad (3)$$
$$F(x) = w_T \cdot \frac{T_{total}}{N_{ref,T}} + w_D \cdot \frac{Dist_{total}}{N_{ref,D}} \quad (4)$$
$$w_T + w_D = 1, w_T, w_D \geq 0 \quad (5)$$

where $N_{ref,T}$ and $N_{ref,D}$ are reference normalization values, and T_{total} represents total lateness across all orders.

This formulation captures the essential trade-off in food delivery systems: minimizing customer wait times while optimizing courier utilization and operational costs. The problem is NP-hard due to its combinatorial nature and dynamic arrival of orders, requiring efficient heuristic or learning-based solution approaches.

4 Methodology

Our approach combines simulation-based modeling with machine learning techniques to address the dynamic meal delivery scheduling problem. The methodology consists of three interconnected components: a discrete-event simulation framework, order aggregation strategies, and courier prioritization mechanisms.

4.1 Simulation Framework

We developed an agent-based simulation model that captures the dynamic nature of food delivery operations. The simulation operates over a finite time horizon with discrete time steps, where events occur at specific timestamps corresponding to order arrivals, preparation completions, pickups, and deliveries.

Agent Architecture. *Orders* are represented as autonomous entities characterized by their creation time r_j, preparation duration $p_{j,R}$, deadline d_j, and spatial attributes, including the restaurant location loc_{R_j} and the customer delivery location loc_{C_j}. Each order thus carries both temporal and geographic constraints that must be taken into account during the assignment and routing processes.

Restaurants operate as processing units with capacity constraints cap_r, maintaining internal preparation queues that follow the FIFO discipline. This component models the flow of orders through the preparation stage and reflects the delays induced by real-world kitchen pipelines.

Couriers function as mobile agents initialized at location $loc_{k,start}$ and moving with a baseline speed $speed_k$. Their effective movement efficiency is influenced by a dynamic stamina parameter $stamina_{k,t}$, which varies over time and affects both travel speed and the agent's ability to handle multiple deliveries.

Stamina Dynamics. Courier fatigue is modeled through a stamina parameter that decreases over time, affecting travel speed:

$$stamina_{k,t} = \max\{0.01, stamina_{k,t-1} - 0.003\}.$$

Simulation Flow. At each time step, the system processes events in the following sequence:

- New orders arrive and enter restaurant preparation queues.
- Completed orders become available for pickup.
- Idle couriers receive assignments based on prioritization strategies.
- Active couriers progress along their routes.
- Deliveries are completed and logged.

4.2 Order Aggregation Strategies

Order aggregation enables couriers to handle multiple orders simultaneously, potentially reducing total distance traveled while managing time constraints. We implement three clustering-based aggregation methods.

K-Means Clustering. Standard K-Means algorithm applied to customer delivery locations with $k = \lfloor N_{orders}/3 \rfloor$ clusters, where N_{orders} is the number of pending orders. Orders within the same cluster are assigned to a single courier when feasible.

DBSCAN (Density-Based Spatial Clustering). Density-based clustering automatically determines the number of clusters based on spatial proximity. The algorithm uses two parameters:

- ϵ: Maximum distance between two samples to be considered neighbors.
- $min_{samples}$: Minimum number of samples in a neighborhood to form a dense region.

DBSCAN identifies clusters of arbitrary shape and marks outlying orders as noise, which are handled individually.

Agglomerative Clustering. Hierarchical clustering using a bottom-up approach with an average linkage criterion. Orders are initially treated as individual clusters, then iteratively merged based on minimum inter-cluster distance until a predefined number of clusters is reached.

Route Optimization. Once orders are aggregated into clusters, we construct optimal delivery routes using the OR-Tools constraint programming solver. The routing problem minimizes total travel distance while respecting pickup-before-delivery constraints and order deadlines.

4.3 Courier Prioritization Strategies

When multiple orders await assignment, the system determines which courier should handle which order or cluster. We evaluate both heuristic and machine learning-based prioritization approaches.

Heuristic Methods.

Earliest Deadline First (EDF): Prioritizes orders by increasing deadline $d_{j_1} \leq d_{j_2} \leq \ldots \leq d_{j_n}$.

Slack Time Priority: Calculates slack time $slack_j = d_j - t_{current} - \hat{T}_{delivery,j}$, where $\hat{T}_{delivery,j}$ is the estimated delivery duration, assigning orders with minimum slack first.

First Process Time (FPT): Assigns orders with shortest expected processing time first, minimizing average completion time.

Longest Process Time (LPT): Assigns orders with longest expected processing time first to balance workload across couriers.

Machine Learning Approaches. We train regression models to predict delivery time $\hat{C}_j$ for each order-courier pair, then prioritize assignments that minimize predicted lateness $\max\{0, \hat{C}_j - d_j\}$.

Feature Engineering. To construct reliable predictors, the models rely on a structured feature representation that captures both the operational context and the current system state. Specifically, the feature set includes:

- Temporal features. Current time, time since order creation, time until deadline, estimated preparation time remaining.
- Spatial features. Distance from courier current location to restaurant, distance from restaurant to customer, total route distance.
- Courier features. Current stamina level, number of orders currently carried, total distance traveled so far.

- Order features. Preparation time, order priority weight.
- System state features. Number of pending orders, number of active couriers.

Model Architectures. To assess the effect of different regression paradigms on predictive performance, we consider several model architectures commonly used in applied machine-learning systems. In this study, three regression algorithms are compared:

- K-Nearest Neighbors (KNN). Non-parametric method that predicts delivery time based on the k most similar historical order-courier assignments using Euclidean distance in feature space.
- Random Forest. Ensemble of decision trees trained on bootstrap samples of historical data, with averaged predictions to reduce variance.
- CatBoost. Gradient boosting on decision trees with ordered boosting and symmetric tree structure, efficiently handling categorical features. CatBoost achieved the best performance with training MAE of 0.721 minutes and test MAE of 1.045 minutes.

Training Procedure. Models are trained on synthetic simulation data generated from 80% of scenarios, with 20% held out for validation. The training set consists of completed deliveries with features computed at assignment time and actual delivery completion time as the target variable. Hyperparameters are tuned using 5-fold cross-validation.

4.4 Hybrid Strategies

We combine aggregation and prioritization into integrated strategies denoted as “Clustering Method, Prioritization Method.” For example, “DBSCAN, ML” applies DBSCAN clustering followed by machine learning-based courier assignment. The baseline “No aggregation” strategy assigns each order individually using the specified prioritization method.

5 Experimental verification

5.1 Experimental Setup

This section describes the experimental protocols, data, and evaluation metrics used to validate the proposed optimization and prioritization strategies for dynamic meal delivery scheduling. Both synthetic datasets and real-world delivery data are utilized to ensure generalizability and relevance of findings.

Synthetic Data. Synthetic orders, couriers, and restaurants are generated to systematically test various configurations and loads (see Fig. 1):

Restaurants. $N_{center} = 2$ main restaurant clusters are generated, each with 2 locations per cluster, using a Gaussian mixture model centered within a defined city bounding box. **Orders.** Customer delivery locations are sampled from a normal distribution centered around corresponding restaurant clusters. Order creation times (t_{create}) follow a skew-normal distribution centered on the system’s peak period ($t_{peak} = 60$ min, scale = 1.2, skew = 0.9). Order deadlines are set as $d_{initial} \sim \mathcal{N}(45, 15^2)$, with a minimum of 30 and maximum of 120 minutes. **Couriers.** Each simulation run randomly assigns couriers’ starting locations within the city bounding box. Courier counts are varied from 10 to 90 to simulate low and high supply regimes.

This design covers a broad spectrum of realistic meal delivery scenarios — from under-staffed to saturated market conditions.

Real Data. To verify practical validity, experiments use an anonymized dataset from a Russian meal delivery platform comprising 1,531 orders and 120 unique couriers collected in 2022. The dataset contains:

- Restaurant and customer coordinates (latitude, longitude)
- Timestamps for order creation, preparation, pickup, and delivery
- Courier vehicle information
- Delivery durations

Experimental Protocol

1. For each scenario, 80% of orders are used for training machine learning models (in hybrid strategies), and 20% as a validation/test set.
2. All aggregation and prioritization strategies are evaluated on the exact same order and courier configuration per seed for fairness.
3. For machine learning pipelines (CatBoost/Random Forest/KNN), features are normalized using Min-Max scaling over each set.
4. Experiments are repeated across 20 random seeds to assess robustness .

Evaluation Metrics. Performance is judged using a composite normalized objective function balancing customer punctuality and delivery operational efficiency:

$$F(x) = w_T \cdot \frac{T_{total}}{N_{ref,T}} + w_D \cdot \frac{Dist_{total}}{N_{ref,D}}, \qquad (6)$$

where $w_T + w_D = 1$, $N_{ref,T} = 15$, $N_{ref,D} = 150$ (see Eq. (20) and discussion in Section 3.3.2 of the thesis). Individual metrics include:

1. Total lateness: $L = \sum_{j\in J} \max(0, C_j - d_j)$.
2. Average courier distance: $\frac{1}{|K|}\sum_k Dist_k$.
3. Share of on-time deliveries.
4. Pareto front diagrams (see Fig. 4.15 for synthetic data and Fig. 4.18 for real-data experiments).
5. Statistical significance tests, including the Friedman test and the Nemenyi post-hoc procedure for pairwise comparison (see Table 7 in the thesis).

Table 1. Normalized objective values for the evaluated strategies

Prioritization	—	Agglo	DBSCAN	KMeans
ML long-first	0.827	0.831	0.832	—
Slack coeff.	—	—	—	0.870
Deadline	0.986	—	0.942	—
Fast-first	—	—	1.220	—

5.2 Results

This section presents the experimental results evaluating the effectiveness of the proposed aggregation and prioritization strategies across both synthetic and real-world delivery scenarios. Performance metrics include lateness, share of on-time deliveries, courier distance, and statistical significance of differences across methods.

All results are supported by robust statistical testing. Friedman test shows significant overall differences across methods ($p < 0.05$). Nemenyi post-hoc test identifies "DBSCAN, ML" and "Agglomerative, ML" as outperforming classic heuristics (see Table 1).

The numerical and graphical results presented in this section highlight both the quantitative performance and the operational behavior of the evaluated scheduling strategies. The analysis begins with aggregated performance metrics and their statistical significance, followed by spatial and temporal visualizations that illustrate how different prioritization and aggregation policies influence delivery dynamics.

Figure 1 illustrates the spatial configuration of restaurants and customers before and after preprocessing. Normalization produces a more uniform and compact geometric layout, which improves the stability of clustering-based aggregation methods used in several of the evaluated strategies.

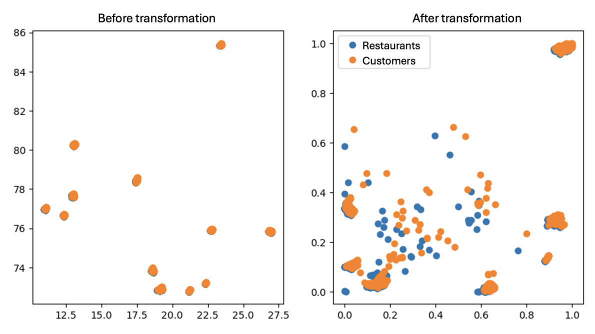

Figure 1. Spatial distribution of restaurants and customers before (left) and after (right) normalization

5.2.1 Synthetic Data Results

Experimentation on synthetic datasets demonstrates clear benefits of combined aggregation and ML-based prioritization:

Best Performing Strategies: "DBSCAN, ML", "Agglomerative, ML", and "ML no aggregation" consistently yield the lowest normalized composite objective values $F(x)$, with "ML no aggregation" serving as a robust baseline (see Table 1).

Table 1 reports the values of the normalized objective function for all prioritization–aggregation combinations. The best-performing strategies are those based on ML-guided long-first prioritization, with or without clustering, while heuristic approaches exhibit noticeably higher objective values.

Statistical Significance. To assess the robustness of the observed performance differences, all experiments were repeated over twenty independent random seeds. The aggregated results indicate that the improvements achieved by aggregation combined with ML-based prioritization are not due to stochastic variation. Statistical analysis using the Friedman test confirms the presence of significant differences among the evaluated methods. To further localize these effects, pairwise comparisons were conducted using the Nemenyi post hoc procedure, with the corresponding p-values reported in Table 2.

Table 2. Pairwise p-values of the Nemenyi post hoc test

Model	1	2	3	4	5	6	7
1	1	1.0	1.0	1.0	1.0	0.327	0.000
2		1	1.0	1.0	1.0	0.464	0.000
3			1	1.0	1.0	0.485	0.000
4				1	1.0	0.955	0.000
5					1	1.000	0.000
6						1	0.048
7							1

The obtained values show that ML-based approaches consistently and significantly outperform the majority of heuristic baselines, whereas the differences among the ML-based variants themselves do not reach statistical significance. This suggests that the primary performance gains arise from incorporating predictive prioritization, while the choice of a specific clustering method affects performance only marginally within the ML-driven framework.

Quantitative Gain. Order aggregation enables a 21–42% increase in number of deliveries per courier and a 13–28% reduction in total mileage compared to non-aggregation (baseline) methods.

5.2.2 Real Data Results

Validation on an anonymized real dataset (1,531 orders, 120 couriers) supports the simulation findings:

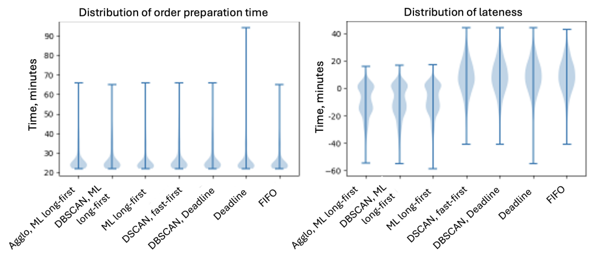

Figure 3. Distribution of delivery completion times (left) and early/late deliveries (right)

Delivery Time Distributions. Aggregation strategies systematically shift delivery completion times earlier, leading to a larger proportion of orders being fulfilled before their promised deadlines. This effect is clearly reflected in the delivery-time distributions and the corresponding Pareto analysis (Fig. 2), where aggregation produces solutions that dominate heuristic baselines across both lateness and traveled distance.

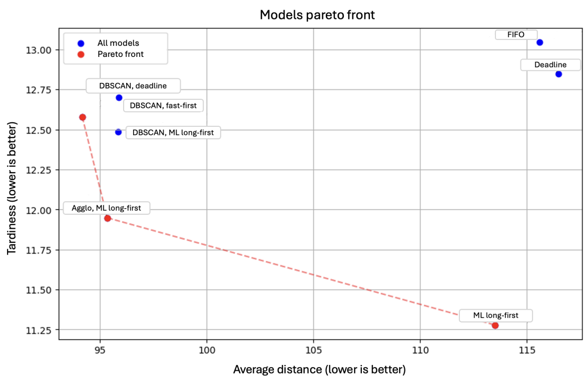

Figure 2. Pareto front of the evaluated strategies on real delivery data

The trade-off between courier mileage and mean lateness on real data, also illustrated in Fig. 2, reveals a well-defined Pareto front. Among the evaluated approaches, the ML long-first strategy attains non-dominated performance, achieving the lowest average lateness while maintaining competitive travel distance. These results indicate that incorporating predictive prioritization enables a more favorable balance between timeliness and operational efficiency.

Efficiency Scaling. The effect of aggregation is most pronounced at higher market demand rates (more than 4 simultaneous orders). Under such conditions, combining DBSCAN clustering with ML prioritization achieves the best trade-off: maximum deliveries with minimum late arrivals.

Operational Metrics. On real data, best strategies reduce the share of late deliveries by 11.3–12.6% and courier mileage by 13–28% compared to baseline assignment heuristics.

The distribution of delivery completion times and early/late deliveries is shown in Figure 3. ML-based prioritization yields noticeably more concentrated distributions with fewer extreme values, whereas heuristic rules such as deadline-first produce significantly heavier tails.

ML Model Performance. Comparisons across regression models (CatBoost, Random Forest, KNN) reveal:

CatBoost. Achieves the lowest Mean Absolute Error (MAE) for delivery time prediction: train MAE = 0.721 minutes, test MAE = 1.045 minutes.

Overall Impact. Use of machine learning decreases median lateness from 12–14 minutes (baseline) to 5–10 minutes (“ML first long” strategy), improving normalized composite objective by up to 11.6% (see metrics and Table 1).

Feature Contributions. The most important features for ML assignment are courier–order distance, time to deadline, and number of loaded orders.

These results demonstrate that order aggregation, especially when combined with ML-based assignment, significantly enhances efficiency and punctuality in high-load delivery environments.

6 Discussion

The results demonstrate that combining order aggregation algorithms with machine learning-based courier prioritization substantially improves the efficiency and punctuality of dynamic meal delivery systems, especially under high market load. The “DBSCAN, ML” and “Agglomerative, ML” strategies consistently expand the Pareto frontier, offering a larger set of solutions with reduced lateness and courier mileage compared to classic heuristics or non-aggregation approaches.

Practical Implications. Order aggregation allows couriers to batch multiple orders, drastically improving operational throughput (by 21–42%) and reducing total travel distance (by 13–28%). This effect is most pronounced during peak periods or when the system experiences more than four concurrent orders. Machine learning-based assignment (particularly with CatBoost) yields more reliable delivery time predictions, reducing median lateness from 12–14 minutes to just 5–10 minutes and enhancing overall customer satisfaction. These results suggest that integrating ML and clustering in real-world platforms could directly translate into noticeable improvements in user experience and reduced operational cost.

Limitations. The study uses a simplified agent-based simulator, with certain abstractions:

- restaurant preparation processes are modeled as FIFO queues and do not account for complex multi-item kitchen dynamics;
- courier stamina, traffic effects, and order preparation variability are modeled stochastically but not from real operational data;

The clustering algorithms do not consider customer delivery time preferences, which may need to be incorporated in practice.
- all ML models are trained on synthetic scenarios; real-world accuracy depends on data fidelity.

Directions for Future Research. Potential directions for further work include several extensions aimed at increasing both the realism and the applicability of the proposed framework. One natural step is to enrich the simulation environment by incorporating more detailed models of kitchen operations and dynamic traffic patterns, thereby making the synthetic environment closer to real-world delivery conditions. Another promising line of research lies in integrating multi-objective optimization techniques, which would allow the system to balance additional objectives such as eco-friendly routing or fairness among couriers.

A further extension involves analyzing the influence of dynamic customer preferences and the role of real-time feedback loops, which may significantly affect assignment decisions and overall system behavior. Finally, an important long-term direction is the deployment of the proposed methodology in operational delivery platforms, enabling large-scale field testing and the development of continuous learning mechanisms under real operating conditions.

Overall, these findings underpin the value of hybrid optimization and data-driven assignment methods and offer a path for more flexible, robust food delivery management in complex urban landscapes.

7 Conclusion

This work examined a set of scheduling strategies for the dynamic meal delivery problem, combining order aggregation and ML-guided courier prioritization. The results obtained on synthetic and real-world datasets show that machine learning–based prioritization consistently improves delivery punctuality and reduces variability in completion times, while clustering-based aggregation provides additional gains in high-load regimes. The statistical analysis confirms that these improvements are significant, and the visual results illustrate clear operational differences between heuristic and data-driven approaches.

The proposed framework offers a flexible basis for integrating predictive models and aggregation mechanisms into real-time delivery platforms. Future research may focus on extending the simulation environment with more detailed traffic and preparation models, as well as on validating the approach in large-scale industrial deployments.

8 Acknowledgment

The theoretical results in Section 1-5,7 were carried out at Saint Petersburg University, project no. 121061000159-6 and the experimental results in Section 6 were carried out with the support of the A.I. Herzen Russian State Pedagogical University (project No. 49VG ”Quality of life of Russian teachers: methodology, theory, measurement methods”).

References

Agnetis, A., Cosmi, M., Nicosia, G., and Pacifici, A. (2023). Two is better than one? order aggregation in a meal delivery scheduling problem. *Computers & Industrial Engineering*, **183**, pp. 109514.

AI, G. (2024). Or-tools: Software for combinatorial optimization. Accessed 15.06.2024.

Bozanta, A., Cevik, M., Kavaklioglu, C., Kavuk, E. M., Tosun, A., Sonuc, S. B., Duranel, A., and Basar, A. (2022). Courier routing and assignment for food delivery service using reinforcement learning. *Computers & Industrial Engineering*, **164**, pp. 107871.

Data, T. (2023). How the e-grocery delivery market has changed over the past three years. In Russian, accessed 08.01.2024.

Dwibedy, D. and Mohanty, R. (2022). Semi-online scheduling: A survey. *Computers & Operations Research*, **139**, pp. 105646.

Editorial, F. R. (2022). Extra pay for delivery: Courier salaries increased by 30% but still in shortage. In Russian, accessed 08.01.2024.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. In Hammer, P. L., Johnson, E. L., and Korte, B. H., editors, *Discrete Optimization II*, vol. 5 of *Annals of Discrete Mathematics*, pp. 287–326. Elsevier.

Jahanshahi, H., Bozanta, A., Cevik, M., Kavuk, E. M., Tosun, A., Sonuc, S. B., Kosucu, B., and Basar, A. (2022). A deep reinforcement learning approach for the meal delivery problem. *Knowledge-Based Systems*, **243**, pp. 108489.

Khalkechev, R. (2020). How we saved couriers' time: Logistics in yandex.eda. In Russian, accessed 08.01.2024.

Khrustaleva, I., Khrustalev, M., Brovkina, Y., Khokhlovskiy, V., and Shkodyrev, V. (2024). The model for optimizing cutting parameters for processing products on metal-cutting machines. *Cybern. Phys*, **13**, pp. 111–122.

Lazarev, A. A. and Gafarov, E. R. (2011). *Scheduling Theory: Problems and Algorithms*. Moscow State University Press. In Russian.

Lazarev, A. A., Lemtyuzhnikova, D. V., and Werner, F. (2021). A metric approach for scheduling problems with minimizing the maximum penalty. *Applied Mathematical Modelling*, **89**, pp. 1163–1176.

Mao, W., Liu, M., Rong, Y., Tang, C. S., and Zheng, H. (2019). Faster deliveries and smarter order assignments for an on-demand meal delivery platform. *SSRN Working Paper*.

Marketing, R. (2023). Analysis of the russian ready-to-eat food delivery market: 2022 results and forecast to 2027. In Russian, accessed 08.01.2024.

Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., and O'Neil, R. (2018). The meal delivery routing problem. *Optimization Online*, **6571**.

Simoni, M. D. and Winkenbach, M. (2023). Crowdsourced on-demand food delivery: An order batching and assignment algorithm. *Transportation Research Part C: Emerging Technologies*, **149**, pp. 104055.

Steever, Z., Karwan, M., and Murray, C. (2019). Dynamic courier routing for a food delivery service. *Computers & Operations Research*, **107**, pp. 173–188.

Ulmer, M. W., Thomas, B. W., Campbell, A. M., and Woyak, N. (2021). The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science*, **55** (1), pp. 75–100.

Yildiz, B. and Savelsbergh, M. (2019). Provably high-quality solutions for the meal delivery routing problem. *Transportation Science*, **53** (5), pp. 1372–1388.

Zhang, G. and Ye, D. (2002). A note on on-line scheduling with partial information. *Computers & Mathematics with Applications*, **44** (3), pp. 539–543.