informs
https://pubsonline.informs.org/journal/ijaa

INFORMS JOURNAL ON APPLIED ANALYTICS
Vol. 54, No. 1, January–February 2024, pp. 84–101
ISSN 2644-0865 (print), ISSN 2644-0873 (online)

THE FRANZ EDELMAN AWARD
Achievement in Operations Research

Meituan's Real-Time Intelligent Dispatching Algorithms Build the World's Largest Minute-Level Delivery Network

Yile Liang,[a,*] Haocheng Luo,[a] Haining Duan,[a] Donghui Li,[a] Hongsen Liao,[a] Jie Feng,[a] Jiuxia Zhao,[a] Hao Ren,[a] Xuetao Ding,[a] Ying Cha,[a] Qingte Zhou,[a] Chenqi Situ,[a] Jinghua Hao,[a] Ke Xing,[a] Feifan Yin,[a] Renqing He,[a] Yang Sun,[a] Yueqiang Zheng,[a] Yipeng Feng,[a] Zhizhao Sun,[a] Jingfang Chen,[b] Jie Zheng,[b] Ling Wang[b]

[a] Meituan Inc., Beijing 100102, China; [b] Department of Automation, Tsinghua University, Beijing 100080, China

*Corresponding author

Contact: liangyile@meituan.com, https://orcid.org/0009-0005-0141-9369 (YL); luohaocheng@meituan.com (HaL); duanhaining@meituan.com (HD); lidonghui03@meituan.com (DL); liaohongsen@meituan.com (HoL); fengjie18@meituan.com (JF); zhaojiuxia@meituan.com (JiuZ); renhao05@meituan.com (HR); dingxuetao@meituan.com (XD); chaying@meituan.com (YC); zhouqingte@meituan.com (QZ); situchenqi@meituan.com (CS); haojinghua@meituan.com (JH); xingke@meituan.com (KX); yinfeifan@meituan.com (FY); herenqing@meituan.com (RH); sunyang21@meituan.com (YS); zhengyueqiang@meituan.com (YZ); fengyipeng@meituan.com (YF); sunzhizhao@meituan.com (ZS); cjf17@mails.tsinghua.edu.cn (JC); j-zheng18@mails.tsinghua.edu.cn (JieZ); wangling@mail.tsinghua.edu.cn (LW)

Accepted: October 31, 2023

https://doi.org/10.1287/inte.2023.0084

Abstract. Meituan pioneered an on-demand food delivery service in China, which allows consumers to place food orders online for fast delivery. Leveraging its minute-level delivery network and boasting more than five million active couriers, Meituan delivers more than 60 million orders daily, ranking it as the world's largest in terms of order volume and courier count. It has transformed itself into one of China's most critical delivery infrastructure providers. To meet its service commitment to customers and improve the working experience of its couriers, Meituan must continuously optimize its order assignment decisions. Thus, it developed a real-time intelligent dispatch system, including a set of algorithms based on operations research and machine learning techniques, to precisely model the assignment problem in a dynamic and uncertain environment, and solved this NP-hard problem in seconds generating a high-quality solution. Since implementing this dispatch system in 2019, Meituan has realized a decrease of 20.96% in average order delivery time and 23.77% in average courier traveling distance per order. In addition, the system contributes to cost reductions of about \$0.23 billion annually. Moreover, the dispatch system enabled other new digital economy business formats, such as Meituan Instashopping and Meituan Grocery, to thrive.

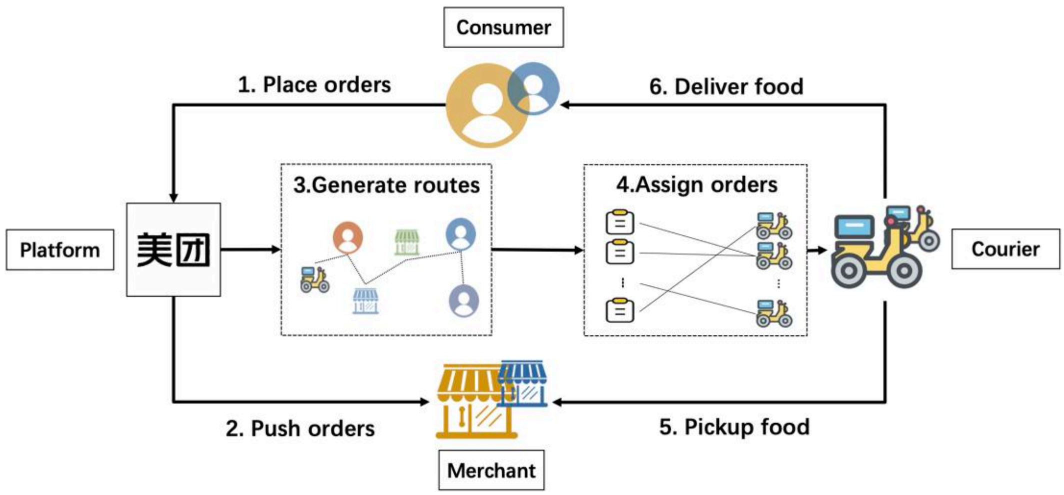
Keywords: order assignment problem • pickup and delivery problem • multiobjective optimization problem • combinatorial optimization • graph representation learning • machine learning • inverse reinforcement learning • Edelman Award

Introduction

In recent years, Meituan has pioneered an on-demand food delivery service in China, encompassing more than five million active couriers, covering more than 3,000 cities, providing more than 9.3 million merchants and 687 million consumers with a reliable and fast pickup and delivery process, and delivering more than 60 million orders each day. With a mere few clicks, consumers can enjoy delicious meals without leaving their homes, all delivered to their doorsteps within just a few dozen minutes. Meituan now holds the largest share of the Chinese on-demand food delivery market. Its minute-level delivery network has rapidly grown into the largest global one and evolved to be one of the most important domestic infrastructures in China.

Consumers anywhere in China can use this service to place food orders. Accordingly, the dispatch system of the delivery platform collects newly placed orders, notifies the corresponding merchants, and then assigns the orders to couriers who are responsible for the pickup and delivery of the orders. Meanwhile, the assignment procedure must seek a sophisticated equilibrium to satisfy several key stakeholders involved in the delivery transactions: consumers want their food delivered within the promised delivery period, merchants hope their food is served fresh to satisfy their consumers, and couriers want to deliver a sufficient number of orders to allow them to make decent wages in a safe and stable environment. Figure 1 shows the overall process, including the platform and stakeholders involved in this on-demand food delivery service.

Figure 1. (Color online) Meituan's On-Demand Food Delivery Process, Which Includes Six Steps and Four Stakeholders (i.e., Consumer, Courier, Merchant, Platform)

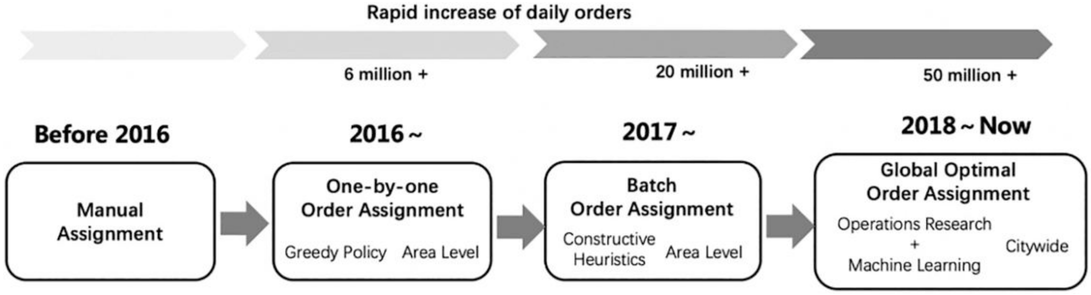


Note. (1) The consumer places an order on the platform, (2) the platform sends the order to the merchant and the merchant starts to prepare the food, (3) the platform generates routes for the order's candidate couriers, (4) the platform assigns the order to an appropriate courier based on the generated routes, (5) the courier accepts the order and picks up the food, and (6) the courier delivers the food to the consumer.

In Chinese culture, food is the primary sustenance in people's lives, and the assignment quality of the dispatch system greatly impacts the experiences of hundreds of millions of consumers. Moreover, the accumulated traveling distance of all the couriers in a given day equals the number of miles required to circle the earth approximately 1,500 times. High-quality assignments can effectively shorten couriers' daily routes while providing them satisfactory incomes, and lead to a significant reduction in carbon emissions (couriers typically use e-bikes for their deliveries; in Appendix A, we provide the calculation of an e-bike carbon footprint). Therefore, improving the assignment quality of the dispatch system constitutes a critical objective for the delivery platform. Meituan addressed these challenges through technological and analytical innovations and developments. It built a real-time intelligent dispatch system to continuously improve the assignment quality.

In the past few years, the dispatch system has evolved through four phases (Figure 2). In the initial phase (i.e., Phase 0), which was the manual assignment of couriers to orders, experienced dispatchers assigned the orders. The system next enabled automatic assignments of couriers to orders, one-by-one sequentially (i.e., not leveraging parallel computing capabilities), in a greedy manner (i.e., Phase 1), which soon encountered severe service quality degradation and unacceptable computational time as the order volume grew rapidly. The platform then upgraded the system into an area-level batch assignment mode based on constructive heuristic methods (i.e., Phase 2); this met the real-time requirements (within 10 seconds) but failed to ensure the assignment quality (i.e., the satisfaction of all stakeholder interests) as the daily order volume increased. The system finally evolved into Phase 3, which realized citywide optimal order assignment (OA) via operations research (OR) and

Figure 2. Evolution of the Dispatch System as the Order Volume Increased



Note. The steps in the evolution are as follows: (1) Phase 0, manual assignment of couriers to orders; (2) Phase 1, area-level one-by-one sequential order assignment based on a greedy policy; (3) Phase 2, area-level batch order assignment based on constructive heuristics; (4) Phase 3, citywide optimal order assignment based on operations research and machine learning methods.

machine learning (ML) techniques and served as the cornerstone to support more than 60 million orders each day. Meanwhile, the research and development group of the dispatch system grew into a team of 30–40 people.

Practically, the dispatch procedures are executed every 30 seconds at a city level. At each dispatch period, the dispatch system requires three stages to generate high-quality OA decisions (i.e., courier behavior estimation, courier candidate evaluation, and OA generation). The first two stages are used for modeling the OA problem, and the last stage is used to develop a high-quality assignment result. The dispatch procedure is shown in Figure 3.

Specifically, in the *courier behavior estimation* stage, the system formulates and solves the courier's pickup and delivery route planning (RP) problem assuming a new order is assigned, providing the evaluation basis of the next stage. In the *courier candidate evaluation* stage, it calculates the matching degree (MD) scores between each new order and available couriers, reflecting the requirements of each stakeholder. For example, "time score" represents the overtime severity (i.e., the extent to which the actual delivery time is later than the promised delivery time) of the order delivered by the courier, and "distance score" represents the total distance increase of the courier caused by the delivery of the order. *OA generation* focuses on solving the many (order)-to-one (courier) assignment problem to optimize the global MD scores.

Business Problem and Challenges

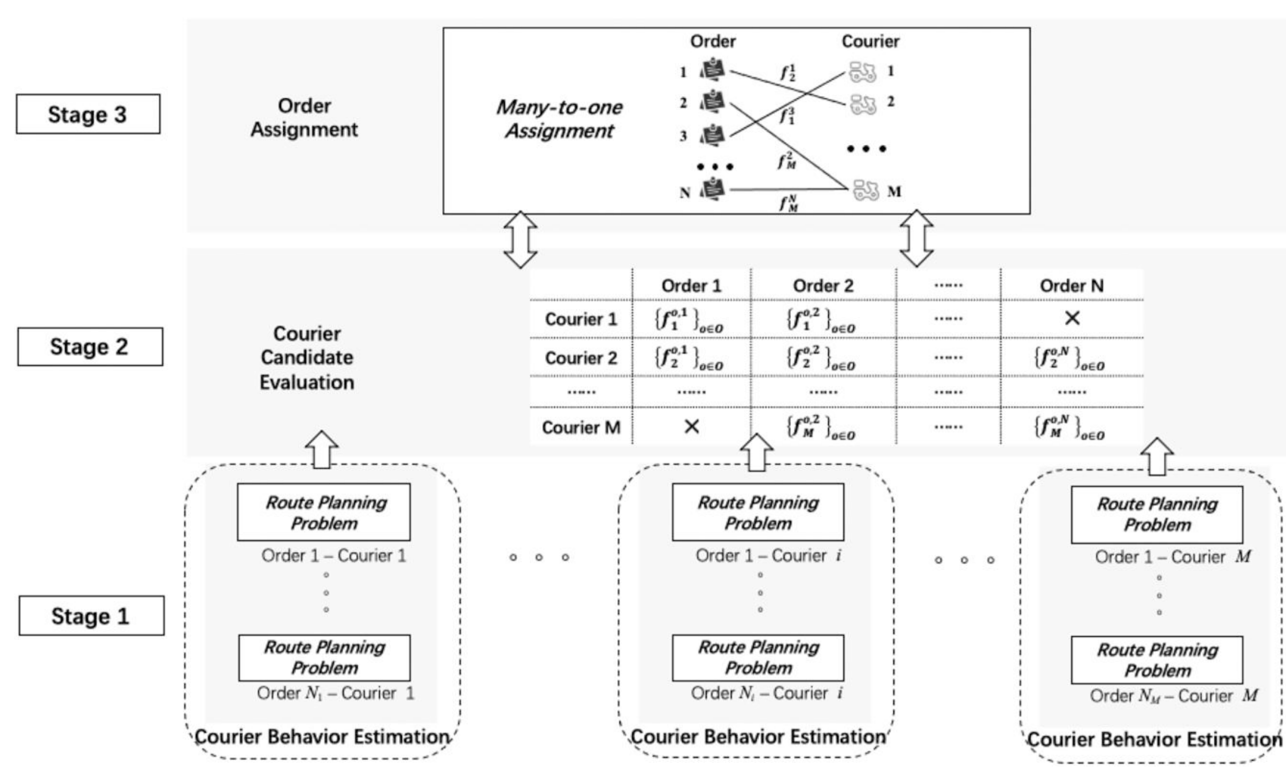
Problem Overview

In essence, the sequential dispatch process for OA for one day, as Figure 4 shows, can be formulated as a multiperiod, multiobjective, combinatorial optimization problem, which we define in Equation (B.1) in Appendix A. The current period's dispatch decisions affect the OA results in the subsequent dispatch period by changing the courier's status. Furthermore, the platform pursues global spatial-temporal optimality, that is, optimizing the maximum MD score across all orders and the couriers assigned in an entire day: $\{\sum_{t=1}^{T} F_t^{o*}\}_{o=1}^{O}$, instead of a single or a few dispatch periods, that is, $\{F_t^{o*}\}_{o=1}^{O}$. We provide explanations of the notation in Appendix B.

Technical Challenges

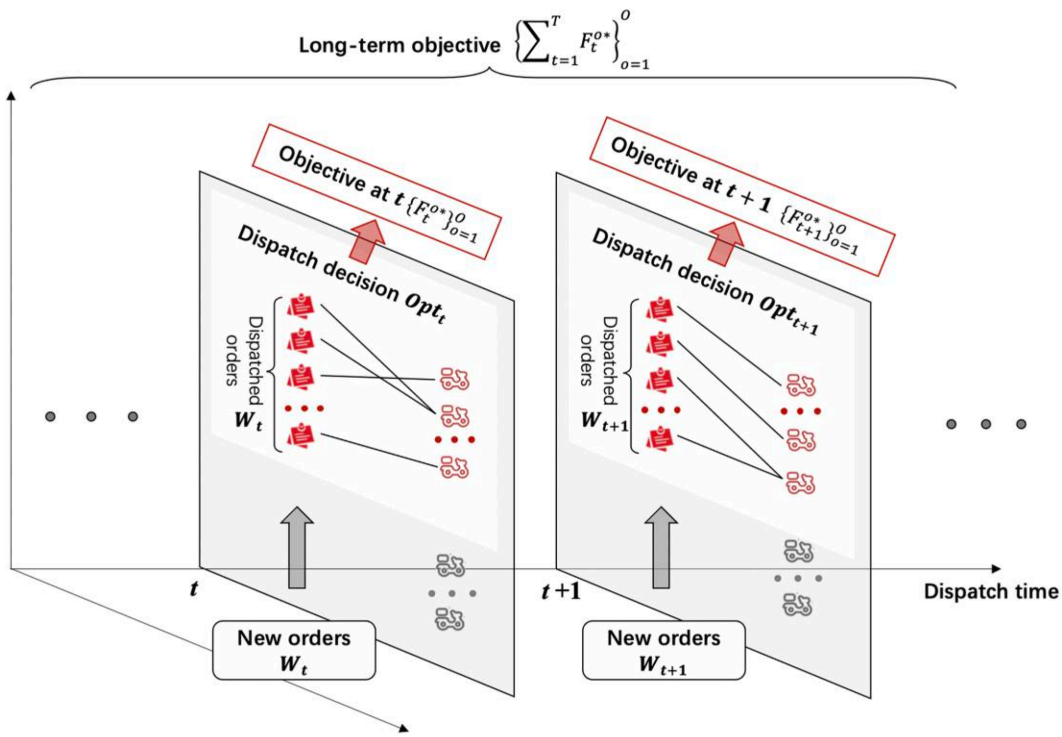
Modeling and solving the above multiperiod, multiobjective decision problem in real time proved to be a challenging task. Consistent with the characteristics of on-demand food delivery service and OA problems, the technical challenges we encountered are as follows:

Figure 3. Dispatch Procedure, Which Is Implemented Every 30 Seconds at a City Level



Note. (1) Courier behavior estimation formulates and solves the courier's route planning problem, providing an evaluation basis for the next stage; (2) courier candidate evaluation calculates the matching degree scores between the new orders and couriers based on the results of Step (1); and (3) order assignment generation formulates and solves the order assignment problem to optimize the global matching degree scores, satisfying each stakeholder.

Figure 4. (Color online) Sequential Dispatch Process



Note. At each dispatch period, the dispatch system collects newly arrived orders and the status of available couriers and determines the best match between them.

Dynamic and Sequential Decision Process. In our setting, dispatch is a dynamic and sequential decision process, which executes every 30 seconds at a city level (Figure 4). At each dispatch moment, the system collects the newly arrived orders and the status of available couriers, and then determines the best matching of the orders and available couriers. Because the system pursues global spatial-temporal optimality and the matching results of the current period will directly affect the results in subsequent periods, considering the influence of future information in the subsequent time steps is necessary to avoid greedy decisions and achieve long-term global optima.

However, predicting the future is difficult. The exact distributions of orders and courier status in the future, as well as the behavior of the couriers and the real-world environment are full of uncertainties, which further complicate the problem. Although we could use a simulation approach to address the uncertainties, building a highly precise simulation system is expensive both financially and computationally and therefore difficult in our setting.

Balancing Multiple Objectives. As we describe previously, OA should balance the goals of the various stakeholders. Consumers want to receive their food on time and exactly as ordered, merchants need the food to be served fresh to satisfy their consumers, couriers need to deliver a sufficient number of orders to allow them to make decent wages, and the platform's goal is to operate the delivery service efficiently.

These dispatch objectives should be precisely modeled in the MD scores between orders and couriers, and accurately calculated in the dispatch procedure. Meanwhile, tradeoffs among these objectives should be made carefully to achieve long-term optimality of the entire system and all the stakeholders.

A common method used to address these multiobjective problems is to combine the multiple objectives into a single combined objective using weights, a technique referred to as a scalarization method (Gunantara 2018). However, finding and calibrating the appropriate weights for different objectives is laborious and nontrivial. The relative scales of the different objectives change dramatically during the various periods of each day. For example, couriers may deliver many more orders during peak hours compared with off-peak hours, which can cause enormous differences in the objectives. Thus, the weights should be adaptively updated to ensure long-term optimality. However, theoretical methods to guide the selection of such weights and guarantee long-term optimality are lacking, especially in such an online and dynamic setting.

Uncertainty. Uncertainty is an inevitable characteristic of the food delivery process. Unexpected events (e.g., traffic jams), extended food preparation time by the merchants, and couriers' personal preferences (e.g., some couriers prefer orders in business district A and some prefer business district B) will greatly affect couriers' behaviors and the service quality of the related orders. However, the system can neither eliminate nor accurately predict these uncertainties. Moreover, the lack of adequate real-time monitoring of the status of couriers and merchants further increases the uncertainties.

As a result of these uncertainties, the system cannot make accurate point estimations on couriers' behaviors, for example, when the courier will pick up or deliver the order. Modeling the MD score and making dispatch decisions based on the results of inaccurate estimation will weaken the effectiveness of dispatch decisions, thereby affecting the experiences of consumers, couriers, and merchants. As an example, if the merchant is slow in preparing the food, the courier must wait in the restaurant until it is ready, which may cause the courier's subsequent orders to be late and result in a conflict between the courier and the merchant. In addition, the impact of uncertainties will be further intensified by the courier's current workload and external factors such as inclement weather.

Solving a Large-Scale NP-Hard Integer Programming Problem in Real Time. For each dispatch period, the many-to-one assignment problem is NP-hard. In our application, couriers are often assigned multiple orders for one delivery route, sometimes as many as five orders during the noon peak. Hence, the scale of our problem is significantly larger than a one-to-one problem. For example, the number of decision variables for a one-to-one assignment problem with N assignment objects is in the order of N^2, whereas the number of decision variables for a five-to-one assignment problem with N assignment objects is in the order of N^6. Furthermore, the MD scores between the couriers and orders are nonadditive; that is, the MD score of assigning several orders simultaneously to a courier is not equal to the sum of the MD scores of assigning the orders separately to the same courier. However, computing the MD scores of arbitrary order combinations and the couriers is unrealistic, especially in a real-time environment, which results in additional challenges to algorithm design.

Couriers usually travel rapidly; therefore, the OA problem must be solved in less than 10 seconds to ensure assignment quality, keeping the courier status (e.g., location and on-hand order number) unchanged during the information acquisition period and assignment generation period. Traditional supply chain optimization problems, although they may require large-scale search spaces, do not require immediate solutions; multiple-hour solutions are permissible. Thus, their methods cannot be adopted directly for on-demand delivery.

Technical Solution

Algorithm Structure of the Dispatch System

To resolve the previous challenges, the dispatch system first decomposes the original multiperiod, multiobjective stochastic assignment problem into a series of single-period, single-objective deterministic subproblems, which can be solved independently at each dispatch moment through weight and MD score design; see Equation (B.2) in Appendix A. Then it applies effective algorithms for solving the subproblem via OR methods and ML techniques to obtain a high-quality solution with superior computational efficiency. At each dispatch period, the algorithm is executed as follows:

In the *courier behavior estimation* stage, we update *the modeling and solution methods of the local courier's RP problem*, to continuously improve the route consistency rate between the planning routes from the solutions and the real ones that couriers actually follow in an uncertain and time-varying environment. It provides a reliable evaluation basis for the evaluation stage.

In the *courier candidate evaluation* stage, the system adopts an *online weight adaptation mechanism* realizing the temporal decomposition and multiobjective integration. It aims to guide the system states gradually evolving into an acceptable long-term balance among each objective. In addition, the system utilizes a *robust and adaptive MD score calculation method* to address the challenges due to the uncertainty in the delivery process. Hence, it transforms the OA problem at each dispatch moment into a single-objective, deterministic problem.

In the *OA generation* stage, we update the algorithm for solving many-to-one assignment problem to continuously improve the solution quality and computational efficiency. In addition to traditional OR methods, we adopt ML techniques to enhance the algorithm performance. For example, we effectively prune the search space and significantly improve the algorithm (for both the solution quality and computational efficiency), via graph neural network (GNN) methods to learn effective order combinations from experienced couriers' behaviors.

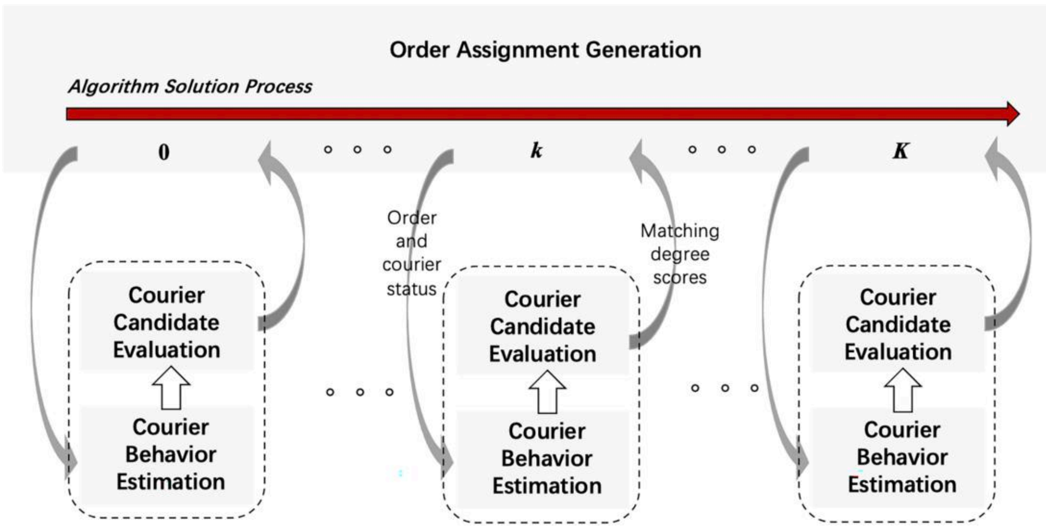
Figure 5 shows the execution of the dispatch algorithm at each period. We cannot compute the MD scores between every order combination and courier due to computational complexity; therefore, we execute the behavior estimation and evaluation of the courier candidates for some promising order combinations, which are determined by the search mechanism of the algorithm, at each iteration.

Here we introduce the major algorithm components of the dispatch system.

Route Planning: Domain Refined Heuristics with Inverse Reinforcement Learning

Different pickup and delivery routes for the same courier can result in different delivery distances and delivery

Figure 5. (Color online) Dispatch Algorithm Execution at Each Period

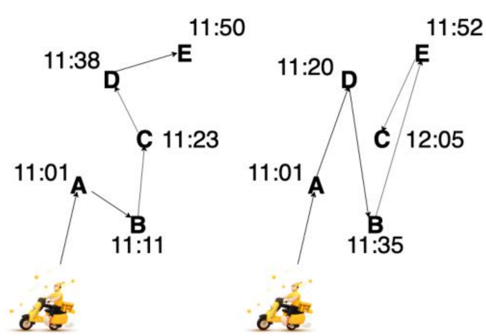


Note. In conjunction with the search mechanism and each iteration of the OA algorithm, the behavior estimation and evaluation of the courier candidates for part of the order combinations are executed sequentially.

time for each order. Figure 6 shows a simple example using different RP results. An inappropriate route implies increased distance as well as longer time for delivery. This route-related information acts as the most important evaluation basis for the matching decision between orders and couriers. Improving the route estimation accuracy is crucial to OA. Moreover, the RP service is called more than 48.3 million times per minute between 11 a.m. and 12 a.m. on a typical day. Taken together, these factors constitute demanding challenges for both the quality of the planning result and the computational efficiency of the algorithm.

To tackle these challenges, we designed a domain-refined heuristic search with inverse reinforcement learning (IRL) for RP. The heuristic search (Zheng et al. 2019) finds optimal solutions for the RP with delivery time constraints, given a specific optimization objective. The optimization objective, which directly determines the RP result, is learned from couriers' historical pickup and delivery routes to improve the route consistency rate between the planning routes and the real routes, as we explain in more detail below.

Figure 6. (Color online) RP Example in Which Different Routes Result in Different Total Delivery Times (Delivery Ends at 11:50 a.m. or 12:05 p.m., Depending on the Route Selected) and Distances for the Same Courier



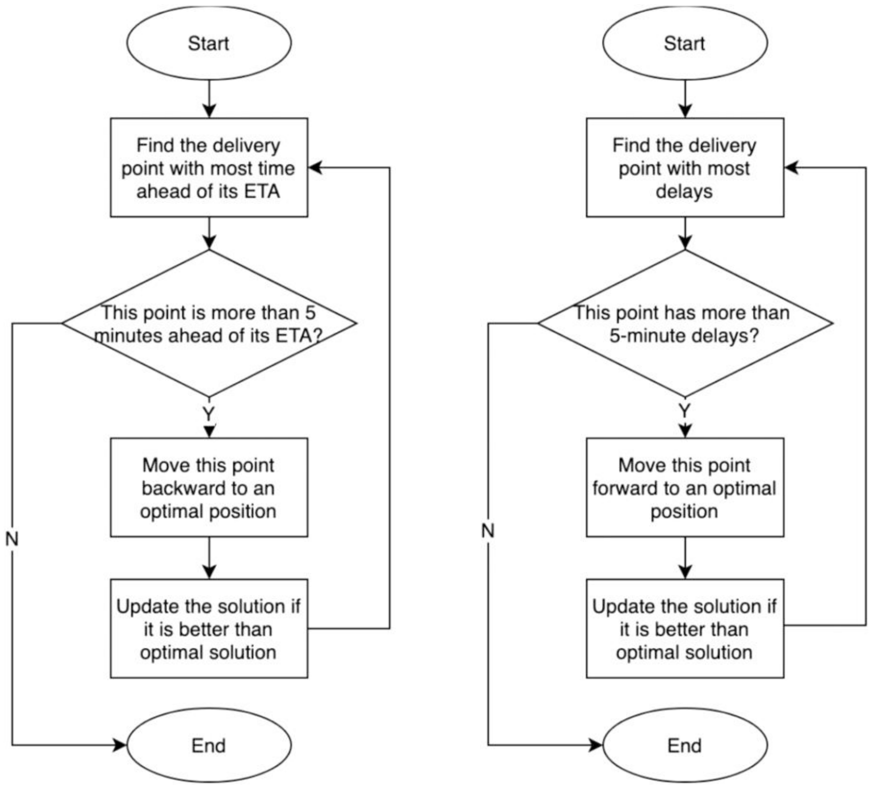
Notes. In this example, the route shown on the left ends with a delivery to E at 11:50 a.m. The route on the right, which is clearly a longer distance, ends with a delivery to C at 12:05 p.m. Most orders on the left route are delivered earlier than the orders on the right route.

Heuristic Search-Based Route Planning. The heuristic search in our system is a two-stage, fast heuristic method (Zheng et al. 2019), including an initialization stage and a local search stage. Its objective is to find the optimal route that minimizes a specific optimization objective $g(r)$, where r is the pickup and delivery route, and g is the objective function. This search problem has two main constraints, a precedence constraint (i.e., each order must be picked up before being delivered) and a capacity constraint (i.e., the orders should not exceed the total delivery capacity of a courier).

The initialization stage uses a greedy insertion procedure. We first sort the orders according to their estimated times of arrival (ETAs) and then insert the pickup and delivery points of each order sequentially and greedily. This greedy insertion provides an initial solution for the RP.

The local search stage is an optimization procedure based on the initial solution. We attempt to move delivery points with delays either forward or backward. As Figure 7 illustrates, on the left, we find the delivery point that provides the most time ahead of its ETA and we try to move it backward to find a better solution (where the objective value is lower). On the right, we try to move forward the point with the most delays. The solutions

Figure 7. Local Search Stage



Notes. On the left, we move the delivery point that provides the most time ahead of its ETA backward to find a better solution. On the right, we try to move forward the point with the most delays.

are updated using these local search operations. The pickup points are also moved simultaneously to achieve a better solution.

Through these two stages, we can successfully reach the global optima (which are obtained by exact algorithms in an offline fashion) within 10 milliseconds in 97% of the cases.

Route Planning Objective Function Learning with IRL. Proper modeling of the RP objective function directly determines the solution quality and reliability. Initially, we can set up a manually defined function based on the delivery distance and the time that exceeds the ETA (Zheng et al. 2019). However, it is obvious that the manual function cannot determine how couriers make decisions during the delivery process. Therefore, the probability that couriers will not be able to deliver orders by the expected time and that the planned routes are unusable is fairly high, especially in an uncertain and time-varying environment (i.e., an environment in which the order and courier distribution vary from time to time).

To ensure the consistency between planned routes and the couriers' actual delivery routes, we propose to learn the objective function from courier historical delivery routes. Specifically, a feature-based maximum entropy (MaxEnt) IRL (Ziebart et al. 2008, Kuderer et al. 2015, Finn et al. 2016) is used to learn a reward function (objective function $g(r)$ in our case) from courier historical delivery routes.

We assume that the possible routes for a courier are sampled from the distribution $p_\phi(r)$. The best route, which the courier prefers, has the highest reward (lowest cost) and the highest probability of being selected. Under this assumption, the probability of a route r is defined as

$$P_\phi(r) = \frac{1}{Z_\phi} e^{-g(r)}, \tag{1}$$

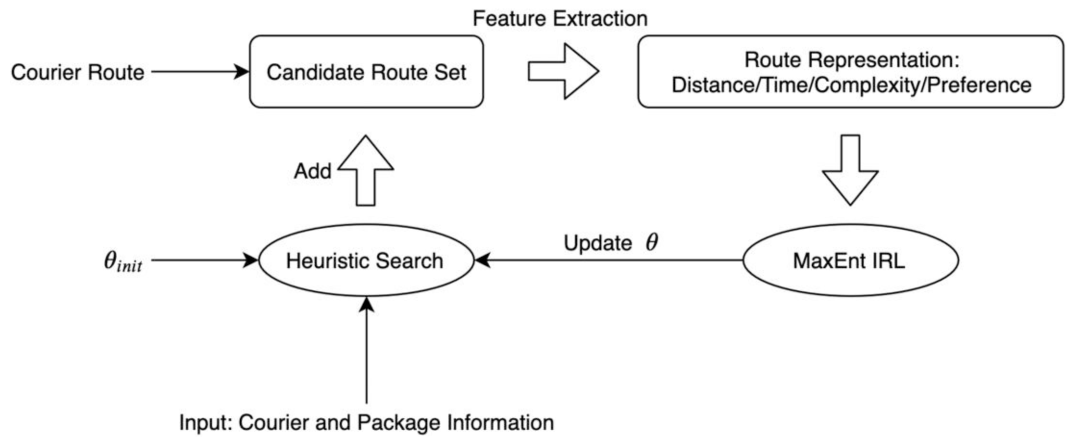
where $Z_\emptyset = \sum_{r \in R} e^{-g(r)}$ is the partition function and R is the set of all possible routes. Figure 8 presents an illustration of our learning pipeline.

Given the courier and order information, the heuristic search can produce a set of possible delivery routes during the local search, in addition to the best route with the minimum cost. In our solution, we use a linear cost function,

$$g(r) = \theta G(r), \tag{2}$$

where $G(r)$ is the feature vector of route r and θ is the feature weights to be learned. After the heuristic search

Figure 8. Inverse Reinforcement Learning (IRL) Pipeline



Notes. The IRL pipeline is an iterative process based on the courier and package information. A heuristic search is used to generate a candidate route set R based on the cost function parameter θ_{init}. Route features are extracted and used in the feature-based MaxEnt IRL. The main goal of the IRL is to determine the parameter θ, which maximizes the probability of the route that the courier uses in the reality. The updated θ is used in the next iteration.

has completed, all possible delivery routes are added to a candidate route set. We extract features from these routes and learn the best parameter θ, which maximizes the probability of the route that the courier employs in the reality. The parameter θ is then updated and used in the next run of the heuristic search. A fixed number of iterations and early stopping criteria can be used in the training phase, considering the training data set size. In practice, two main considerations are the feature representation of the route and the rationality of using the candidate route set in IRL. For the feature representation of the route, we use four kinds of route features based on our domain knowledge, namely, delivery distance, delivery time, route complexity, and courier's route preference. For example, we use the total navigation/line distance and the average navigation/line distance for pickup/delivery points in our implementation. For rationality of the candidate route set, we should sample all possible routes in an ideal IRL setting, although this becomes infeasible as the number of orders increases. Instead, we take advantage of the information in the local search stage of the heuristic search. Because we attempt to move points forward or backward in the local search, rich temporary routes (ranging from a few to about a hundred in our data set) can be generated for a single training sample. This temporary route set can cover more than 90% of a courier's real route. In that case, we treat this set as the possible route set R and replenish it by adding new routes during the training phase.

Through IRL, we provide a learning solution for the objective function in the heuristic search-based RP. Additionally, we develop our RP algorithm as a Java package, which serves the system using several clusters.

Matching Degree Score Calculation: Adaptive Risk Decision Making Through Bayesian Optimization

Considering the inevitable uncertainties of the delivery process and the impact of resulting invalid dispatch decisions, we introduce a risk management approach to improve decision effectiveness and the robustness of the resulting solution. The couriers' performance, such as on-time rate and delivery efficiency, reflected by the MD score, follows a probability distribution. The worst-case risk of the distribution can be quantified by the conditional value-at-risk (CVaR) (Tamar et al. 2015), which is a statistical risk measure widely used in both practice and theory. The average of the performance distribution is crucial under normal circumstances. Therefore, the MD score should comprehensively consider the courier's performance in both high-risk (f^e) and average (f^r) system states, as we show in Equation (3), where $\alpha \in (0,1)$ denotes the degree of risk aversion:

$$f = (1-\alpha)f^e + \alpha f^r. \tag{3}$$

Accordingly, there are two major challenges in MD score evaluation. On one hand, real-time decision making must be completed in milliseconds; however, the calculation of risk measures involves large-scale sampling and repeated RP calculations for each sample. An effective sampling method should be adopted to reduce complexity and simultaneously guarantee the accuracy of the calculation of tail risk. On the other hand, the system has different risk attitudes in different scenarios. Specifically, during the peak lunch hours or under extreme weather conditions, the system tends to adopt a more risk-averse approach, and vice versa. Realizing risk aversion degree adaptation (i.e., tailoring the degree of risk aversion to changing contexts) with minimum trial-and-error cost online is also critical.

The MD score evaluation process can be divided into two parts:

Calculation of Risk Measures. Without loss of generality, we only take the uncertainties of the meal preparation time into account. As Figure 9 shows, assume a courier is assigned L orders in total, and the mealtime of the lth pickup node, denoted as t_{ml}, follows an independent distribution denoted as $h_{ml}(t)$. First, we adopt online layered sampling (Ding and Wang 2020) for each mealtime to satisfy the requirement for a millisecond-level calculation. Next, considering that the execution times of nodes in sequence are highly correlated with each other, we integrate the pointwise mealtime samples into route samples through route construction. The total overtime interval of each route sample can then be separately evaluated as MD score samples. Finally, the MD score samples are aggregated into risk measures under two states (i.e., the high-risk state and the average state). Given the risk percentage β and the number of samples N, the CVaR of scores (i.e., f^r) is approximated by averaging the worst βN MD score samples, and f^e is obtained by averaging all MD score samples.

Risk Aversion Degree Adaptation. In practice, the system may have different risk preferences for uncertainties in different operational scenarios. We need to adapt the risk aversion degree to actual operational scenarios according to the near-real-time performance feedback. However, the mapping function between the risk aversion degree and the performance indicators of interest (e.g., on-time rate, travel distance) under the uncertain and time-varying environment does not assume any specific analytical functional forms. Therefore, we introduce the concept of black-box optimization (Frazier 2018), which refers to a type of problem where the objective function and constraints are not explicitly known or defined, and the optimization algorithm can only interact with the black box by providing input values and receiving corresponding output values. We adopted Bayesian optimization, as one of the most commonly used black-box optimization algorithms, in our system.

As Figure 10 shows, in the offline training phase, we choose a Gaussian process (GP) model to fit the probabilistic mapping function. In the online decision-making phase, we build the model input for each risk aversion degree in the candidate set, together with environmental context and time series variables, and run the GP model. The outputs are expressed as the mean and variance of the online performance indicators to be optimized. The best risk aversion degree for the next observation period (i.e., every 30 minutes in practice) is selected based on the upper confidence bound (UCB) (Contal et al. 2013) criterion, with the principle of

Figure 9. (Color online) Process of Calculating Risk Measures Has Three Steps: Online Sampling of Mealtime, Route Sample Integration, and MD Score Aggregation

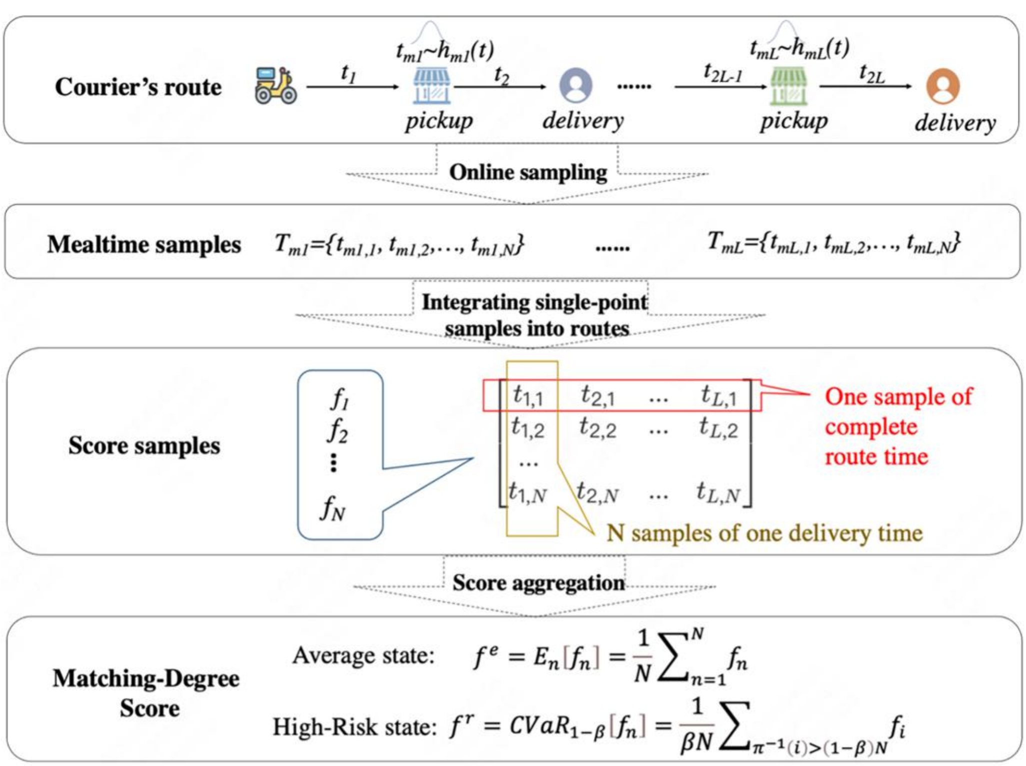
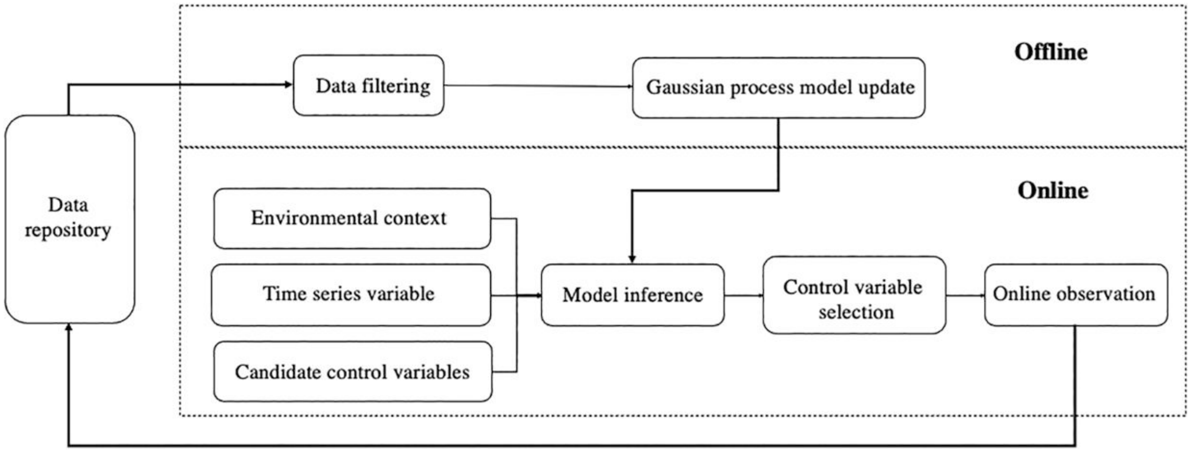


Figure 10. Risk Aversion Degree Adaptation Consists of Two Iterative Phases: Offline Model Training and Online Decision Making



delivery experience and efficiency balance. At the end of each observation period, the performance indicators of interest evolve to new states and the model is updated accordingly. This method can obtain the desired MD score with minimum trial-and-error cost and ensure the stability and decision feasibility of the online dispatch system.

Our decision-making system with risk aversion degree adaptation has undergone A/B testing in nine cities across China, each lasting for at least two weeks, where the optimal MD score of each time period fluctuates between 0.5 and 0.95. Evaluation results of the testing showed that the on-time rate increased by 0.27 percentage points (pp), and the average travel distance was reduced by 1.20%.

Dynamic Objective Weight Adaptation: Combining Online Decision Making with Offline Planning

To balance different dispatch goals, we implemented a target-oriented, multiobjective balancing framework inspired by Lyu et al. (2019) to adaptively adjust the weights of different objectives, such as courier efficiency and consumer experience, during the sequential dispatch process.

We first obtain from historical operational data the "ideal point," which represents a desired or highly acceptable value of each objective we seek for the long term and serves as the evolving target of the system during the entire dispatch process. Then we adjust the weight of each objective based on the gap between the objective's current value and corresponding target value at each dispatch period, with the goal of decomposing the original assignment problem into independent single-period problems and guiding the system states to finally converge to a solution nearest to the target.

Because the values of the objectives change dramatically over the daily horizon, as we explain earlier, we divide each day into a series of dispatch periods; $U^o_{p_t}$ is the "ideal point" of objective o during dispatch period p_t, where time step t is in period p_t.

During the online dispatch process, $U^o_{p_t}$ is used as the target at time step t; $\delta^o_t = U^o_{p_t} - f^o_t$ is calculated to measure the gap between the objective value f^o_t and the ideal point $U^o_{p_t}$ at time step t. To find weights for the next time step $t+1$, we use exponential smoothing to calculate a smoothed average of the gaps from previous time steps. The weight is updated according to $\eta^o_{t+1} = \gamma(\delta^o_t)^+ + (1-\gamma)\eta^o_t$, where γ is a smoothing parameter between zero and one.

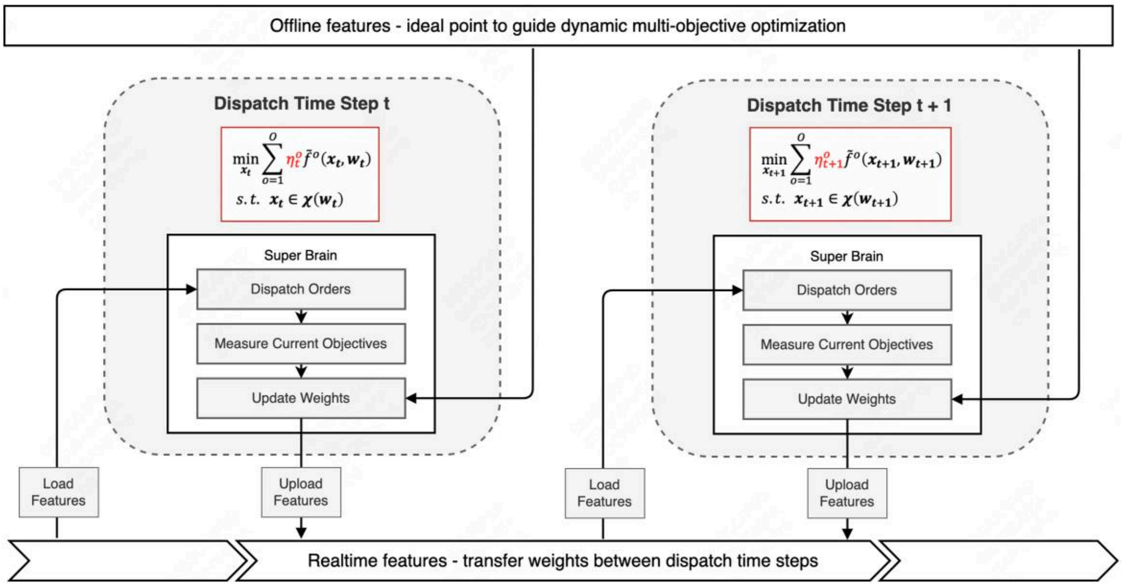
Figure 11 illustrates our current implementation. The ideal point is generated based on the performance of historical dispatch decisions. The objective values of historical decisions are averaged by cities and periods, where a filtering mechanism is also implemented to reduce the noise in historical data. Then, offsets are added to the average historical values to provide overall guidance for the dispatch process. These offsets can be tuned to suit the preferences of decision makers to accommodate different objectives. For example, if one objective is more important, we can add a larger offset to this objective, which can increase its weight during online dispatching.

We implemented this framework in the dispatch system and conducted A/B testing to measure its performance. When we tested it in five cities in China, each lasting for at least two weeks, the on-time rate increased by 0.33 percentage points (pp), and the average travel distance decreased by 0.92%. During this testing, the daily cumulative MD score of the experimental group improved by 3.6%–4.5%.

Order Assignment: Operations Research Combined with Machine Learning

The solution quality and computational efficiency of the OA algorithm are important. The dispatch system uses

Figure 11. (Color online) Implementation of the Multiobjective Balancing Framework



Notes. The "ideal point" is generated based on the performance of historical dispatch decisions and provided as offline features to the dispatch system. The dynamic weights are transferred as real-time features between different dispatch moments. During each dispatch moment, the gaps between current objective values and ideal points are measured and used to update the objective weights.

traditional OR methods in conjunction with ML techniques for OA and uses several techniques to optimize the OA algorithm performance.

Method 1: "Divide-and-Conquer" Framework with Imitation Learning and a GNN Algorithm Framework. To solve the many-to-one assignment problems in a real-time manner, the dispatch system employs an enhanced divide-and-conquer solution framework, in which it decomposes the original large-scale many-to-one assignment problem at each dispatch moment into a series of one-to-one subproblems, which are solved sequentially, resulting in a progressive online solution process. Each subproblem is simple and can be solved in polynomial time, thus greatly reducing the computational burden. To guarantee solution quality, the decomposition and iteration rounds are carefully guided and controlled by a global coordinator, which is an ML model trained by an imitation learning (IL) approach (Chen et al. 2022). High-quality expert solutions are obtained by a well-designed offline OR algorithm. Figure 12 depicts the divide-and-conquer framework.

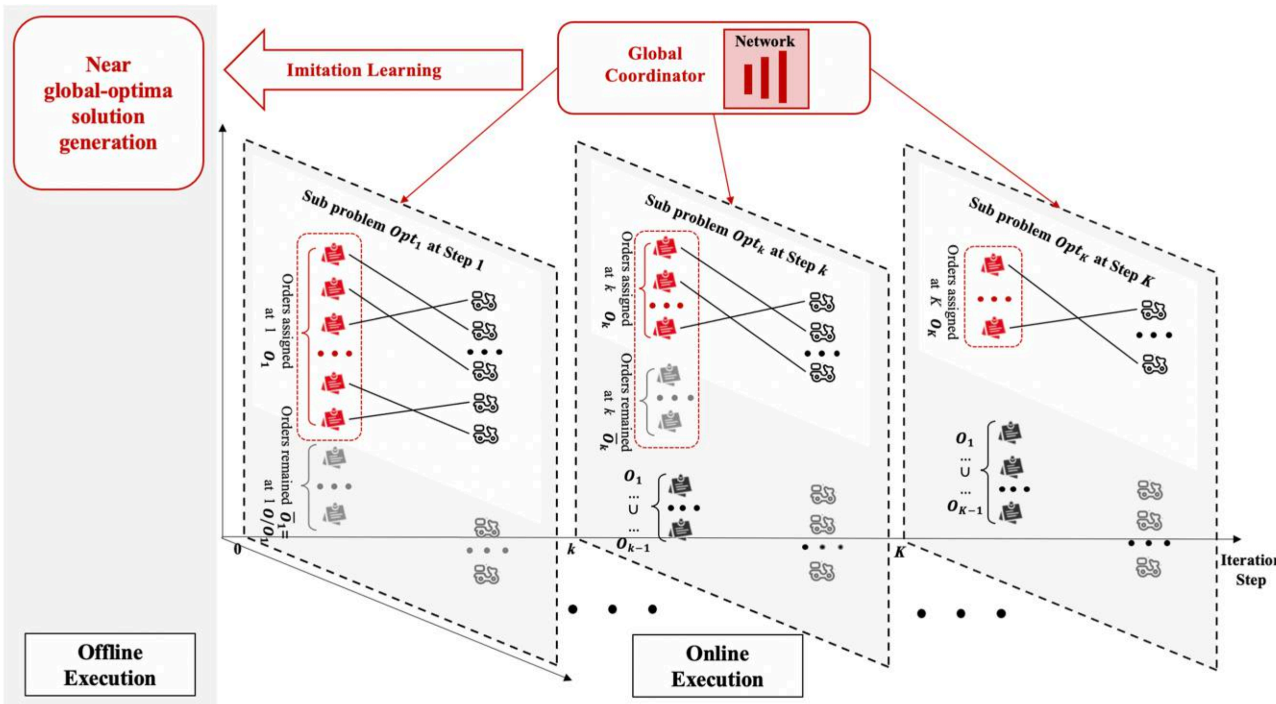
Divide-and-Conquer Strategy. The divide-and-conquer strategy splits orders into multiple iterative steps to construct the corresponding assignment results in sequence; that is, at each iteration, only a part of the orders are assigned to the couriers. The iteration process ends when a complete assignment result is generated. Within this framework, the many-to-one assignment result is built by separating the orders into different iterations rather than a direct combination of orders. Only limited MD scores are calculated at each iteration, thus greatly reducing the computational burden of the previous stages. Moreover, the subproblem at each iteration degrades to a polynomial-time problem in which the scale is much smaller than the original one and that can be solved easily by traditional OR methods, such as the Kuhn-Munkras algorithm (Munkres 1957).

Therefore, within this framework, choosing orders to assign at each round, that is, how to execute decomposition, is important to the solution quality. With well-designed decomposition strategies, the methods can achieve better solution quality with limited MD score calculation volume, with polynomial or linear complexity, instead of combinatorial explosion. Meanwhile, the algorithm for solving each subproblem is of polynomial complexity.

Execution Architecture. In practice, to improve the solution quality, the dispatch system adopts a ML coordinator to guide the decomposition online. The dispatch system is trained offline using the IL method and uses expert solutions from an enhanced tabu search method as its labels.

During the offline training phase, a traditional tabu search method is optimized to improve its solution quality. Based on the expert solutions generated by the enhanced tabu search method, we employ the IL method

Figure 12. (Color online) Imitation Learning (IL) Divide-and-Conquer Framework



Notes. The original many-to-one assignment problem is decomposed into a series of one-to-one subassignment problems, which are solved sequentially. A global coordinator is trained offline using IL methods with high-quality solutions as its labels. In addition, the coordinator controls and guides the online solution procedure.

(He et al. 2012) to train a global coordinator generating the optimal matching probability of the orders and their available couriers.

During the online solution stage, the orders to assign at each iteration are chosen according to the online inference outputs of the coordinator; that is, orders with higher average optimal matching probabilities are selected to be assigned at the current iteration. Because the coordinator extracts global information from the expert solutions, the algorithm can effectively avoid local greed; moreover, it can search more widely and deeply in the search space, thus coming closer to the quality of expert solutions.

The proposed algorithm, which we tested using nationwide data randomly sampled from actual operational data in 2021, improved assignment quality by 5%–10%; in addition, as the order volume increased, the degree of improvement also increased.

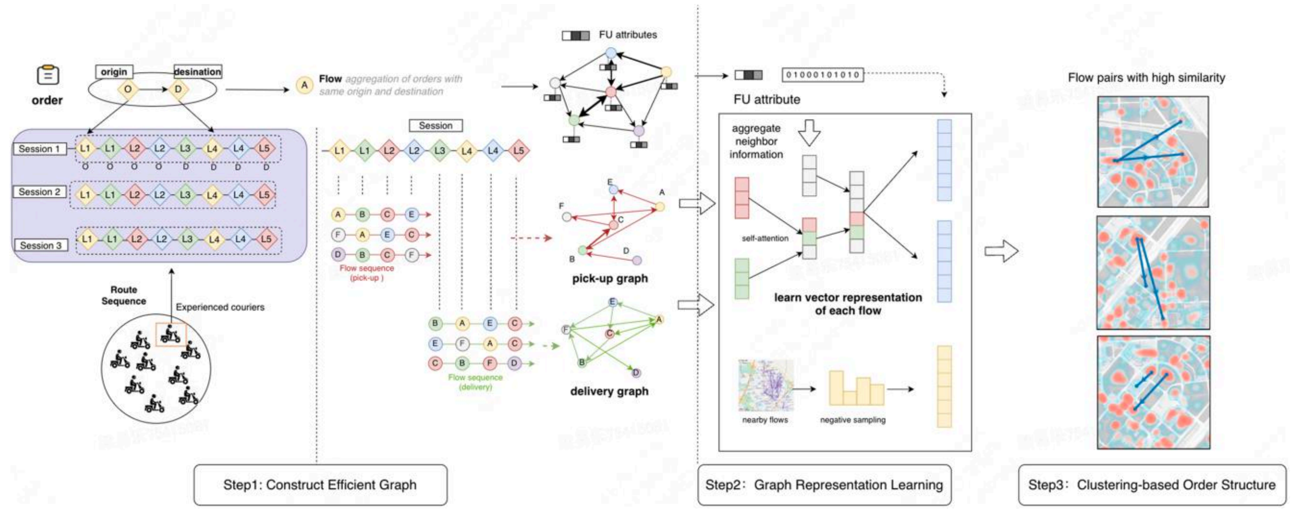
Online Order Combination: Learning from Experienced Couriers via GNN. To reduce the number of iterations and additionally improve the computational efficiency without sacrificing solution quality, it was necessary to construct an efficient online order combination mechanism, forcing the subproblem at each iteration to generate many-to-one assignments. The mechanism needed to recognize the complicated relationships between different orders and find effective ways to combine them as mutually exclusive groups.

An intuitive solution is to group orders with nearby origin-destination pairs and delivery time windows to share delivery resources. We call this method a static order combination when it only uses limited and static information from the order itself. After interviewing experienced couriers and observing the actual delivery of orders in practice, we found that many couriers could flexibly group various orders based on their personal perceptions and experiences learned over the course of their work, a subjective and dynamic approach that is more complicated than a static grouping approach.

Accordingly, we developed a GNN-based framework by learning directly from experienced couriers and using this knowledge to guide the online order combination; that is, by analyzing the historical patterns of experienced couriers, we can generate promising order combinations. With the learn-from-humans mechanism, this method could identify more delivery patterns and cluster orders by finding effective and flexible ways to solve the assignment problem. Figure 13 presents the framework of this proposed method. Detailed designs of the three steps described are as follows.

Delivery Network Construction. We first group orders with the same origin area of interest (AOI) and destination

Figure 13. (Color online) Framework for Modeling the Delivery Network via GNN



Note. The framework contains three steps: (1) construct efficient graph: constructing an efficient delivery graph with flow (i.e., aggregation of orders with the same origin and destination) as a basic unit by merging massive delivery records from experienced couriers; (2) graph representation learning: applying GNN to model the graph and extracting combination patterns from it as representative vectors of each flow; and (3) clustering-based order structure: using representative vectors to cluster orders into various groups to decompose assignment tasks and working with the above "divide-and-conquer" algorithm to achieve better matching results.

AOI as a flow session. Based on the couriers' experiences, we preprocess the delivery order sequence from selected couriers as independent delivery sessions and convert each session to a flow session. When regarding flow as a node and nearby relations in sequence as a link, we can merge these flow sessions into a union graph as a delivery network. Furthermore, each flow node also exhibits useful attributes to describe its crucial characteristics, for example, the delivery distance between origin and destination and the average delivery time in the past 30 days.

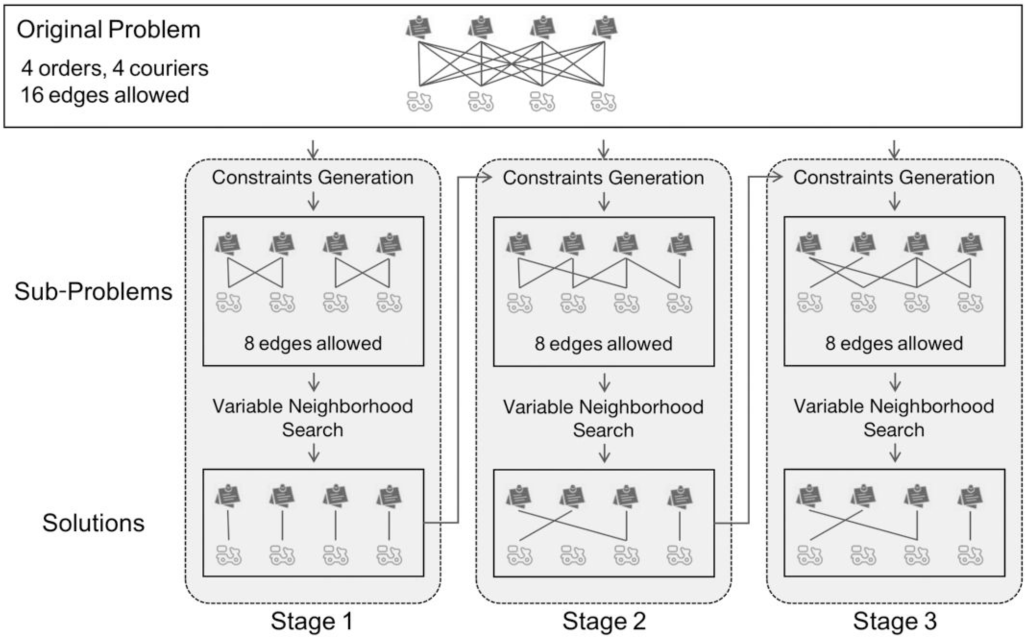
Graph Representation Learning. With the flow-based delivery network as input, we apply graph representation learning methods (Hamilton 2020) to extract specific patterns from it. Our goal is to learn the vector representation of a node, which contains its attributes and topology information in the graph after training. We apply classical graph learning models (Hamilton et al. 2017, Wang et al. 2018) to learn the basic vector representation of flow node and apply some well-known methods (Bengio et al. 2009) to optimize the quality of vector representation.

Online Order Combination. After obtaining the vector representation of the flow node, we define a similarity-based metric to combine the orders together. Basically, we use the dot-product results of vector representations of two flow nodes as the similarity metric of the combination. Similar flow nodes with large dot-product results can be regarded as a candidate group to participate in the assignment problem. As we show on the right side of Figure 13, we use this similarity metric to cluster the historical orders into communities and validate the clustering results (e.g., whether geographical similarity is encoded) by humans in various cases.

Method 2: Multistage Subproblem Reformulation with Machine Learning. We also propose a multistage solution algorithm based on neighborhood search and ML to solve the many-to-one assignment problem at each dispatch moment. The algorithm constructs subproblems by adding constraints to the original one, which limits the variations in the solution space of the current-stage subproblem compared with the previous stage. Specifically, one source of the complexity of the original many-to-one assignment problem is that orders can be arbitrarily grouped together as a batch and assigned to the same courier (the number of batched groups increases exponentially with the number of orders in the groups). To reduce complexity, the main constraints added to the subproblem are the selections of the newly batched groups that can be added in the current stage. In addition, we also include constraints on the number of assignable couriers for the orders/groups and the number of assignable orders for the courier. These added constraints greatly reduce the scale and complexity of the problem, resulting in a significant reduction in the construction and solution complexity. By solving the multistage subproblems sequentially, we can obtain an approximate solution to the original problem.

A key aspect of the proposed multistage solution algorithm is how to select appropriate new constraints. We utilize a combination of rule-based and ML-based constraint generation methods to achieve better results.

Figure 14. Use of the Proposed Multistage Solution Algorithm, Which Considers Four Orders and Four Couriers



Notes. At each stage, the constraints generation method is utilized to add constraints to the original problem, formulating a subproblem with smaller complexity (i.e., fewer edges allowed). The constraints generation method uses the original problem and the solution of the previous stage as inputs, forming a solution space that is close to the previous solution and, if possible, contains the optimal solution. The proposed variable neighborhood search algorithm is then implemented to solve the subproblem. The overall process stops when no additional new subproblems can be formulated or when solutions of consecutive stages remain the same. In this example, the method stops after three stages.

For the rule-based method, we use expert knowledge to generate various constraints. For example, we can implement a rule such that only the orders that have near pickup points can be grouped as a batch and assigned to the same courier, or a rule that each order can be assigned only to the courier when the order's pickup point is close to the courier's current location. Different rules will affect the construction of subproblems, which in turn will affect the solution quality and efficiency of the original problem. We apply these rules to our practical examples and choose the best rules that balance quality and efficiency. For the ML-based method, we use the branch-and-bound method to solve the many-to-one assignment problems offline and use the resulting batched groups to train an XGBoost-based classification model (Chen et al. 2015) to predict whether the batched groups will be present in the final solutions. In the online stage, the model predictions are used to generate constraints to guide the selections of batched groups. By combining these methods, the construction of each subproblem (i.e., the computational time of $f^{o,r}_{t,\overline{w}}$) can be completed within seconds.

Although we limit the complexity of the subproblems by adding constraints, they still cannot be solved in real time using exact methods such as branch-and-bound. To address this, we propose a variable neighborhood search algorithm (Hansen and Mladenović 2001) in which we design neighborhood operators to exploit the problem structure and numerical features. During the neighborhood search process, we efficiently select the optimal neighborhood operator by estimating the impact of the operator on the objective function. This enables an efficient search mechanism and high-quality solutions.

Figure 14 shows an example of the proposed method. Note that the proposed multistage solution algorithm provides some benefits of general large-neighborhood search algorithms (Pisinger and Ropke 2019). However, by carefully designing the added constraints based on the problem characteristics, we have been able to significantly reduce the complexity while facilitating the solution trajectory rapidly approaching the optimal solution. Typically, to solve the citywide order assignment problems, only three to five stages are needed to converge to a satisfactory near-optimal solution and the total construction and solution time is within 10 seconds.

Reassignment: Technical Solutions for Abnormal Scenario

In most cases, the courier who accepts the assignment is expected to deliver the assigned order. However, because we assign orders over 30 minutes on average before a courier's arrival, we cannot predict all the potential problems that couriers may encounter during the actual delivery process. If unforeseen problems arise, completing the original assignments may negatively affect the delivery experience and efficiency of the courier. For example, when delivering multiple orders, if the preparation of one order is delayed, the courier must wait for the order at the merchant's location, which will delay the arrival time of all the orders that the courier is delivering. We use the reassignment scheme to address this problem: based on the latest environmental and estimated information, we reassign orders that have not yet been picked up to other appropriate couriers if doing so will improve the quality of assignment and delivery efficiency in dynamic and uncertain environments.

Our system implements two types of reassignment. One is initiated by the system, and the second is initiated by the courier. Both types implement similar processes:

Step 1: Collecting orders for reassignment. For those initiated by the system, the orders that merit reassigning are identified by an online algorithm that monitors the delivery process of couriers. Specifically, we run RP algorithms each minute to estimate the delivery sequences and distances of the couriers in the cases in which the couriers retain all their assigned orders, as well as the cases in which one or more orders have been removed from a courier's assigned deliveries and reassigned to another courier. By comparing the estimated delivery processes of the courier before and after the removal, we choose the order that is most beneficial for delivery efficiency because the order needed to be reassigned. Additionally, couriers can also request reassignment through our application if they encounter abnormal problems that will negatively affect their delivery schedules.

Step 2: Finding appropriate couriers for the orders. Unlike our standard dispatch problem, the reassignment problem is a one-to-one assignment problem. We use the Hungarian algorithm (Kuhn 1955) and the same objectives as in a standard dispatch.

Step 3 (note: we skip this step for reassignments initiated by the courier): Contact the courier currently assigned to the order through a pop-up prompt on our application to inquire if that courier agrees to the reassignment. If the courier declines, we halt the reassignment process.

Step 4: Using our application, contact the courier who is considered to be more appropriate for the reassigned order to inquire if that courier is willing to take on the reassignment. If the courier accepts the reassignment, we complete the process; if the courier declines the reassignment, we stop the process. We note that if an order triggers a reassignment but is ultimately declined by either courier, this order will not trigger a reassignment again in the future.

With an overall decline rate of 30%, about 1% of all orders are completed by new couriers found through the reassignment process, according to nationwide statistics for June 2022. Prior to its nationwide implementation, a two-week A/B experiment, covering approximately 5% of the national order volume, was conducted in 10 cities and showed that the delivery intervals for these reassigned orders were on average shortened by over one minute.

Benefits

The dispatch system has provided significant benefits, including business development and financial benefits for Meituan, optimized experiences for couriers, consumers, and merchants, reduced environmental impacts, and enhanced response to the COVID-19 pandemic.

Business Development

Since its listing on the Hong Kong stock exchange in 2018, Meituan, as one of the world's largest online and on-demand delivery platforms, has achieved significant growth. From 2019 to 2022, the number of cities Meituan covers increased by 40% and the number of registered users increased by 60.76% (Table 1). In addition, the number of merchants on the platform increased by 55.93%, and the number of couriers serving the platform to deliver food increased by 32.08%. The daily average number of transactions in peak seasons (e.g., summer peak from July to August) increased by 100% to about 60 million.

As the number of transactions increases, the dispatch problem becomes more challenging, especially in peak hours (e.g., lunch hours from 11 a.m. to 1 p.m.) when about 20 million orders could be assigned to about five million couriers.

The dispatch system that we developed increased the efficiency of couriers and significantly improved the satisfaction of couriers, consumers, and merchants.

Courier and Consumer Experience

For consumers, the time from placing an order to receiving the delivery decreased by 20.96% with the implementation of the dispatch system. For couriers, the average number of daily delivered orders per full-time courier increased by more than 109%. In addition, the average courier's travel distance per order decreased by about 0.5 km for each order (from 2 to 1.5 km). For merchants, the average pickup time per order decreased by 16.0%.

Table 1. Our Estimates of Benefits Meituan Derived from the Dispatch System (2022 vs. 2019)

Benefit measure	Increase/decrease
Average delivery time per order	−0.96%
Average number of orders delivered by couriers per day	+109.63%
Average travel distance per order	−3.77%
Average meal waiting time	−6.00%

Note. Meal waiting time refers to the time interval from a courier's arrival at the merchant to picking up the meal.

Financial Benefits

From 2019 to 2022, the system contributed to a daily decrease of $0.64 million in costs (associated with payments to couriers) to Meituan, which amounts to a cost reduction of about $0.23 billion annually.

Environmental Impact

With the implementation of the dispatch system, we reduced about 532,125 kg of carbon dioxide (CO_2) emissions per day. We calculate this as follows:

The average delivery distance reduced per order × the number of orders per day × the CO_2 emission per kilometer by e-bike (the courier's primary mode of transportation) = the CO_2 reduction per day.

The average delivery distance reduced per order = 0.5376 km; the number of orders per day = 60,000,000; and the carbon footprint of an e-bike in China is about 16.5 g of CO_2 per kilometer. Appendix A provides the detailed steps for calculating the carbon footprint of an e-bike in China, using the following formula:

$$0.5376\text{ km per order} \times 60{,}000{,}000\text{ orders} \times 0.0165\text{ kg/km} = 532{,}125\text{ kg}.$$

Response to COVID-19

The dispatch system helped Meituan respond quickly to the many complexities associated with the COVID-19 pandemic. Under the severe epidemic situation in many cities in China, the demand for food delivery surged; however, the supply of couriers decreased significantly. Thus, the supply and demand ratio became extremely tight, and a considerable number of orders were canceled. By adapting the system flexibly, including denying "irresponsible refusals" for couriers, relaxing the consideration of punctuality (i.e., tolerating late deliveries), and changing Meituan's objective to completing the orders rather than delivering them on time, we provided better experiences for consumers, couriers, and merchants, especially in cities that were severely impacted by the pandemic.

Extending the Use of the Dispatch System

Meituan's vision is to help its customers eat better and live better. In this sense, the role of the dispatch system is critical. It enables the thriving of other new business formats in the digital economy within Meituan, such as Meituan shopping, Meituan groceries, and Meituan drugs.

With the rapid development of the digital economy in recent years, real-time assignment problems are becoming more common and fundamental in many industrial fields. The analytics and OR techniques used in a dispatch system are not only limited to a food delivery setting but are highly portable to other settings both within and outside of Meituan; examples include dispatching computational resources and displaying online advertisements.

Conclusions

After years of development and research, Meituan's real-time intelligent dispatch system, built on a set of algorithms derived from operations research and machine learning techniques, can accurately model and solve the order assignment problem within seconds, generating high-quality solutions. Since its implementation in 2019, Meituan has achieved a remarkable reduction of 20.96% in average order delivery time and 23.77% in average courier travel distance per order. Moreover, the system has contributed to yearly cost reductions of approximately $0.23 billion. It has successfully enabled Meituan to achieve a win-win situation by striking a balance among the goals and interests of all stakeholders involved.

Acknowledgments

The authors thank Hui Li, Xianying Fan, Sicheng Liu, Huanjia Lian, Ruoyu Zhang, Xingyu Liu, and Min Wu for promoting the system.

Appendix A. Calculation of an E-Bike Carbon Footprint

The carbon footprint of an e-bike can be calculated as follows.

In the following calculations, g/km represents grams per kilometer and km/h represents kilometers per hour.

The carbon footprint of an e-bike can be measured by manufacturing consumption + electricity consumption + caloric output.

(1) Manufacturing consumption: 7 g CO_2 per km

The European Cyclists' Federation (ECF) (European Cyclists' Federation 2011) estimates that an e-bike has an average manufacturing carbon footprint of 134 kg of CO_2. With a lifespan of 19,200 km, the manufacturing consumption of an e-bike can be calculated as 7 g/km = 134 kg/19,200 km.

(2) Electricity consumption: 3.2 g CO_2 per km

According to Bosch's e-bike range calculator (Bosch 2022), a city e-bike with a 500 watt hour (Wh) battery will provide a range of 94 km under typical commuting conditions (assuming an average speed of 22 km/h, mountain bike tires, sports assistance mode (i.e., a driving mode similar to the driving behavior of food delivery drivers), and 85 kg combined weight (i.e., the weight of the driver and the weight of the delivery)). Assuming a charging efficiency of 90% (some energy from the plug is not imparted into the battery), an e-bike will require 500 Wh/94 km/90% (i.e., 5.9 Wh) of electricity from the grid to travel each kilometer. In addition, the carbon intensity of electricity in China in 2021 was 0.549 g of CO_2 per Wh of electricity (Statista 2022). Thus, we calculate electricity use as 5.9 Wh × 0.549 g/Wh = 3.2 g CO_2 per km.

(3) Caloric output: 6.3 g CO_2 per km

The ECF (European Cyclists' Federation 2011) assumes an average 70-kg cyclist on an e-bike will burn only 4.4 calories per kilometer in addition to the calories the cyclist will burn when not riding an e-bike. Thus, with the ECF's estimate for caloric output emissions (1.44 g CO_2 per calorie), we calculate 6.3 g CO_2 per km (4.4 calories/km × 1.44 CO_2/calories) = 6.3 g CO_2 per km for caloric output.

Therefore, 7 g/km + 3.2 g/km + 6.3 g/km = 16.5 g/km (carbon footprint).

Appendix B. Mathematical Notation

Here we present the notation that we use in this paper.

The multiperiod, multiobjective dispatch problem can be formulated as

$$\text{optimize}_{x_t \in X_t} \left\{ \sum_{t \in T} F^o(W_t, R_t, x_t) \right\}_{o \in O}$$
$$\text{s.t. } x_t \text{ non-anticipative}, t \in T, \tag{B.1}$$

where x_t is nonanticipative, which means that decisions made at time t are based on the information we have at time t. This assignment problem can be decomposed into a series of single-period, single-objective deterministic assignment problems that are solved independently each time as we show here:

$$\begin{aligned} &\min_{x_t \in X_t} \sum_{o \in O} \eta_t^o \sum_{\overline{w} \in \overline{W_t}} \sum_{r \in R_{t,\overline{w}}} f_{t,\overline{w}}^{o,r} x_{\overline{w}}^r \\ &\text{s.t.} \sum_{\overline{w}(w)} \sum_{r \in R_{t,\overline{w}(w)}} x_{\overline{w}(w)}^r = 1 \qquad \forall w \in W_t \\ &x_{\overline{w}}^r = \prod_{w \in \overline{w}} x_w^r \qquad \forall \overline{w} \in \overline{W_t}, \end{aligned} \tag{B.2}$$

where $R_{t,\overline{w}}$ is the set of available couriers at time t for order combination $\overline{w}$, and $f_{t,\overline{w}}^{o,r}$ is the matching degree score of objective o for dispatching order combination $\overline{w}$ to courier r at time t.

Table B.1. General Notations

Notation	Description
t	Dispatch time, where $t \in T$
W_t	The set of new orders at time t
R_t	The set of available couriers at time t
x_t	Decision variables at time t, where $x_w^r = 1$ or $x_{\overline{w}}^r = 1$ means order w or order combination $\overline{w}$ is assigned to courier r
X_t	The set of constraints that restrict each order to be assigned to only one courier
o	o-th objective of the system, where $o \in O$
$\overline{w}$	Order combination, which contains one or more orders, $\overline{w} \in \overline{W_t}$ where $\overline{W_t}$ is the set of all possible order combinations at time t
$\overline{w}(w)$	Order combination, which contains order w,
F_t^o	Objective function value of objective o at time t, $F_t^o = F^o(W_r, R_t, x_t)$
η_t^o	Weight for objective o at time t

Table B.2. RP with IRL

Notation	Description
$g(r)$	Cost function of route r
$P_\phi(r)$	Probability of route r being chosen by a courier, $P_\phi(r) = \frac{1}{Z_\phi} e^{-g(r)}$, where $Z_\phi = \sum_{r \in R} e^{-g(r)}$ is the partition function and R is the set of all possible routes

Table B.3. Adaptive Risk Decision Making Through Bayesian Optimization

Notation	Description
f	Matching degree score, $f = (1-\alpha)f^e + \alpha f^r$, where f^e is the expected score, f^r is the risk score, and α denotes the preference for the risk
t_{ml}	Mealtime of the l-th pickup node, follows independent distribution $h_{ml}(t)$
T_{ml}	Sample set of t_{ml}, $T_{ml} = \{t_{ml,1}, t_{ml,2}, \ldots, t_{ml,N}\}$, where N is the number of samples
f_n	Matching degree score corresponding to the nth sample

Table B.4. Dynamic Objective Weight Adaptation

Notation	Description
p_t	Each day is divided into a series of periods and p_t is the period containing time t
$U_{p_t}^o$	"ideal point" of objective o during period p_t

References

Bengio Y, Louradour J, Collobert R, Weston J (2009) Curriculum learning. *Proc. 26th Internat. Conf. Machine Learn.* (Association for Computing Machinery, New York), 41–48.

Bosch (2022) eBike range assistant. Accessed September 29, 2023, https://www.bosch-ebike.com/en/service/range-assistant/.

Chen JF, Wang L, Ren H, Pan J, Wang S, Zheng J, Wang X (2022) An imitation learning-enhanced iterated matching algorithm for on-demand food delivery. *IEEE Trans. Intelligent Transportation Systems* 23(10):18603–18619.

Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, et al. (2015) XGBoost: Extreme gradient boosting. *R package version 0.4-2* 1(4):1–4.

Contal E, Buffoni D, Robicquet A, Vayatis N (2013) Parallel gaussian process optimization with upper confidence bound and pure exploration. Blockeel H, Kersting K, Nijssen S, Železný F, eds. *Proc. Machine Learn. Knowledge Discovery Databases: Eur.* Conf., Lecture Notes in Comput. Sci. (Springer, Berlin), 225–240.

Ding H, Wang Z (2020) Layered sampling for robust optimization problems. *Internat. Conf. Machine Learn.* (PMLR, New York), 2556–2566.

European Cyclists' Federation (2011) Cycle more often 2 cool down the planet! Accessed January 6, 2023, https://ecf.com/system/files/Cycle_More_Often_2_Cool_Down_the_Planet.pdf.

Finn C, Levine S, Abbeel P (2016) Guided cost learning: Deep inverse optimal control via policy optimization. *Proc. 33rd Internat. Conf. Machine Learn.* (Association for Computing Machinery, New York), 49–58.

Frazier PI (2018) A tutorial on Bayesian optimization. Accessed July 10, 2023, https://arxiv.org/pdf/1807.02811.pdf.

Gunantara N (2018) A review of multi-objective optimization: Methods and its applications. *Cogent Engrg.* 5(1):1–16.

Hamilton WL (2020) Graph representation learning. *Synthesis Lectures Artificial Intelligence Machine Learn.* 14(3):1–159.

Hamilton WL, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. *Adv. Neural Inform. Process. Systems*, vol. 30 (ACM, New York), 1025–1035.

Hansen P, Mladenović N (2001) Variable neighborhood search: Principles and applications. *Eur. J. Oper. Res.* 130(3):449–467.

He H, Eisner J, Daume H (2012) Imitation learning by coaching. *Adv. Neural Inform. Process. Systems*, vol. 25.

Kuderer M, Gulati S, Burgard W (2015) Learning driving styles for autonomous vehicles from demonstration. *Proc. IEEE Internat. Conf. Robotics Automation* (Institute of Electrical and Electronics Engineers, Washington, DC), 2641–2646.

Kuhn HW (1955) The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.* 2(1–2):83–97.

Lyu G, Cheung WC, Teo C-P, Wang H (2019) Multi-objective stochastic optimization: A case of real-time matching in ride-sourcing markets. Preprint, submitted March 20, http://dx.doi.org/10.2139/ssrn.3356823.

Munkres J (1957) Algorithms for the assignment and transportation problems. *J. Soc. Industry Appl. Math.* 5(1):32–38.

Pisinger D, Ropke S (2019) Large neighborhood search. Gendreau M, ed. *Handbook of Metaheuristics* (Springer, New York), 99–127.

Statista (2022) Carbon intensity of the power sector in China from 2000 to 2022. Accessed July 10, 2023, https://www.statista.com/statistics/1300419/power-generation-emission-intensity-china/.

Tamar A, Glassner Y, Mannor S (2015) Optimizing the CVaR via sampling. *Proc. 29th AAAI Conf. Artificial Intelligence* (Association for the Advancement of Artificial Intelligence, Palo Alto, CA), 2993–2999.

Wang J, Huang P, Zhao H, Zhang Z, Zhao B, Lee DL (2018) Billion-scale commodity embedding for ecommerce recommendation in Alibaba. *Proc. 24th ACM SIGKDD Internat. Conf. Knowledge Discovery and Data Mining* (ACM, New York), 839–848.

Zheng H, Wang S, Cha Y, Guo F, Hao J, Sun Z (2019) A two-stage fast heuristic for food delivery route planning problem. *Proc. INFORMS Annual Meeting*.

Ziebart BD, Maas AL, Bagnell JA, Dey AK (2008) Maximum entropy inverse reinforcement learning. *AAAI*, vol. 8 (AAAI Press, Palo Alto, CA), 1433–1438.

Yile Liang is the principal algorithm expert of Meituan Delivery. She leads the research and algorithm development of the real-time intelligent dispatch system. Her research interests include on large-scale optimization, stochastic optimization, machine learning, and their applications in the fields of decision intelligence. She received her PhD degree in electrical engineering from Tsinghua University.

Haocheng Luo is an algorithm expert at Meituan. His research interests include large-scale optimization and its application in logistics. He received his BS and PhD degrees both in electrical engineering from Tsinghua University.

Haining Duan is an algorithm expert at Meituan Delivery Platform. Her interests include stochastic optimization and their applications in on-demand delivery. She obtained a PhD degree from the Department of Electronic Engineering, Tsinghua University.

Donghui Li received his PhD degree and BE degree in the Department of Industrial Engineering, Tsinghua University. He is currently an algorithm expert with Meituan. His research focuses on the application of data-driven optimization methods to real-world problems, especially under dynamic and stochastic scenarios.

Hongsen Liao is a senior engineer at Meituan. He leads the research of vehicle routing problem and delivery time prediction. His research interests include human behavior modeling, human-computer interaction, and visualization. He received a BS degree and PhD degree in computer software from Tsinghua University, in 2012 and 2017, respectively.

Jie Feng is an algorithm expert at Meituan. He works on mining and applying human knowledge in the dispatching system to make it more intelligent. He obtained his PhD degree in electronic engineering from Tsinghua University, Beijing, China, in 2021. His papers, which cover the research topics on human mobility modeling and time series modeling, have total citations of more than 1,200.

Jiuxia Zhao is an algorithm engineer at Meituan. Her main research direction includes deep reinforcement learning and decision intelligence, especially for order dispatching problems and vehicle routing problems. She received her MS degree from the School of Automation Science and Technology from Xi'an Jiaotong University, Xi'an, China, in 2021.

Hao Ren is an algorithm expert at Meituan. His current research direction is the application of machine learning in rider behavior and decision making. He received his BS degree and MS degree in industrial engineering from Tianjin University.

Xuetao Ding is a principal research engineer of Meituan. His main research direction is machine learning and decision intelligence, including recommendation, order dispatching algorithm, causal learning for pricing, and supply demand optimization.

Ying Cha is a senior engineer at Meituan. She leads the research of global flexible intelligent delivery order scheduling and vehicle routing planning. Her research interests include operational research optimization and machine learning. She received her MS degree in systems engineering from Northeastern University.

Qingte Zhou is currently an algorithm expert at Meituan. His main focus includes order batching and order assignment in on-demand delivery system. He obtained an MS degree in logistics engineering from Tsinghua University.

Chenqi Situ is an algorithm expert at Meituan. He works on order assignment. He obtained an MS degree in control engineering from Huazhong University of Science and Technology.

Jinghua Hao is currently the leader of the algorithm group in delivery platform of Meituan. He obtained a PhD degree from the Department of Automation, Tsinghua University.

Ke Xing is a senior algorithm expert at Meituan. His current research interests include the application of machine learning, spatial data mining, and ubiquitous computing. He obtained a BS degree in computer science from Harbin Engineering University, Harbin, China.

Feifan Yin is currently a technical researcher at Meituan. His main focus is algorithm engineering in the real-time delivery system. He obtained an MS degree in computer science from Beihang University.

Renqing He is currently the technical head of Meituan Delivery.

Yang Sun currently plays the role of engineering director at Meituan. His main focus includes order assignment and autonomous robot that provides "last mile" delivery service. He obtained his BS degree in computer science at University of Toronto and MS degree in computational data science at Carnegie Mellon University.

Yueqiang Zheng is currently the principal development engineer at Meituan, responsible for the architecture design of Meituan's real-time dispatching system. He obtained his MS degree from the University of Science and Technology of China.

Yipeng Feng is currently a senior development engineer at Meituan, mainly responsible for the architecture design of Meituan's real-time dispatching system. He obtained an MS degree in computer science from Zhejiang University.

Zhizhao Sun is currently the chief technical officer of Meituan Waimai.

Jingfang Chen received his PhD degree in control theory and control engineering from Tsinghua University. He worked as a research intern at Meituan from 2019 to 2022. His main research interests include intelligent optimization on scheduling problems.

Jie Zheng received her PhD degree in control theory and control engineering from Tsinghua University in 2023. She worked as a research intern at Meituan from 2019 to 2022. Her main research interests include the scheduling problem under uncertainty with intelligent optimization.

Ling Wang received his PhD degree in control theory and control engineering from Tsinghua University in 1999. He has been with the Department of Automation, Tsinghua University, since 1999, where he became a full professor in 2008. He has authored five academic books and more than 300 refereed papers. His current research interests include computational intelligence-based optimization and scheduling.