Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0000-0000 | EISSN 0000-0000 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000
© 0000 INFORMS

Meituan's Real-time Intelligent Dispatching Algorithms Build World's Largest Minute-level Delivery Network

Yile Liang, Haocheng Luo, Haining Duan, Donghui Li, Hongsen Liao, Jie Feng, Jiuxia Zhao, Hao Ren, Xuetao Ding, Ying Cha, Qingte Zhou, Chenqi Situ, Jinghua Hao, Ke Xing, Feifan Yin, Renqing He, Yang Sun, Yueqiang Zheng, Yipeng Feng, Zhizhao Sun
Meituan Inc., liangyile@meituan.com, luohaocheng@meituan.com, duanhaining@meituan.com, lidonghui03@meituan.com, liaohongsen@meituan.com, fengjie18@meituan.com, zhaojiuxia@meituan.com, renhao05@meituan.com, dingxuetao@meituan.com, chaying@meituan.com,zhouqingte@meituan.com, situchenqi@meituan.com, haojinghua@meituan.com, xingke@meituan.com, yinfeifan@meituan.com, herenqing@meituan.com, sunyang21@meituan.com, zhengyueqiang@meituan.com, fengyipeng@meituan.com, sunzhizhao@meituan.com,

Jingfang Chen, Jie Zheng, Ling Wang
Tsinghua University., cjf17@mails.tsinghua.edu.cn, j-zheng18@mails.tsinghua.edu.cn, wangling@mail.tsinghua.edu.cn,

Meituan pioneers an on-demand food delivery service in China which allows consumers to place food orders online for instant delivery. With its world's largest minute-level delivery network and over 5 million active couriers, Metiuan delivers more than 60 million orders every day and has evolved to be one of the most important infrastructures in China. To meet the service commitment and improve couriers' working experience, it is of great importance for Meituan to continuously optimize the order assignment decisions. Thus it builds 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 solve this NP-hard problem in seconds with high quality. Since implementation, the dispatch system has realized a decrease of 20.96% in average order delivery time and 23.77% in average riding distance per order. And it can contribute to about $0.23 billion cost reduction in one year. Moreover, the dispatch system enables the thriving of other new business formats of the digital economy in Meituan, such as Meituan Instashopping and Meituan Grocery.

Key words: order assignment problem; pick-up and delivery problem; multi-objective problem; combinatorial optimization; graph representation learning; machine learning; inverse reinforcement learning

Introduction.

In recent years, Meituan has been pioneering an on-demand food delivery service in China, covering more than 3,000 cities with over 5 million active couriers, providing over 9.3 million merchants and

687 million consumers with a reliable and fast delivery process. It now occupies the largest share of the Chinese on-demand food delivery market. Its minute-level delivery network has rapidly grown into the largest one globally, delivering over 60 million orders every day, which has evolved to be one of the most important domestic infrastructures. According to the financial report in 2022, the revenue of the core local business of Meituan in the second quarter is $5.4 billion. The net profit for the same period is $1.2 billion.

In this scene, food orders are continuously placed by consumers anywhere. Accordingly, the dispatch system of the delivery platform collects newly-placed orders, pushes them to the corresponding merchants, and then assigns them to the couriers who are responsible for the pick-up and delivery. Meanwhile, a sophisticated equilibrium should be carefully sought during the assignment procedure, to satisfy several key stakeholders involving in the delivery transactions: consumers want reliable and fast service, merchants hope their food served fresh to satisfy their consumers, and couriers pursue to deliver enough orders so as to make a decent wage in a safe and stable environment. The overall procedure of the platform and stakeholders involved in the on-demand food delivery service are shown in Figure 1.

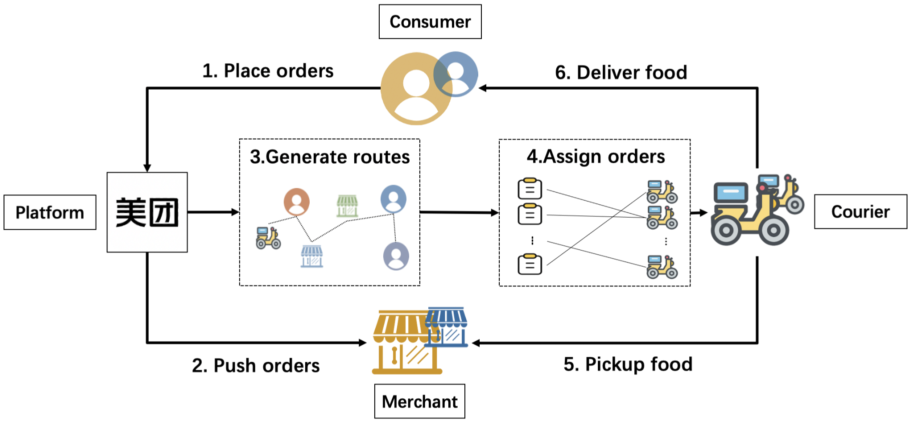


Figure 1 **General procedure of on-demand food delivery contains 6 steps with 4 stakeholders (consumer, courier, merchant, platform). 1) consumer places order on the platform, 2) platform pushes the order to the merchant, and the merchant starts to prepare food, 3) platform generates routes for the order's candidate couriers, 4) platform assigns the order to an appropriate courier based on the results of 3), 5) the courier accepts the order and picks up food, 6) the courier delivers food to the consumer.**

In Chinese culture, food is the first and foremost sustenance in people's lives, and the assignment quality of the dispatch system will greatly impact hundreds of millions of consumers' experiences. Moreover, the accumulated traveling distance of all the couriers a day can circle the earth for about 1500 rounds. High-quality assignments can effectively shorten couriers' daily routes while guaranteeing their incomes, thus leading to a significant reduction in carbon emissions. Therefore, how to improve the assignment quality of the dispatch system constitutes a fundamental problem for the delivery platform.

In response, Meituan addresses the above challenges through technological innovations and developments. It builds a real-time intelligent dispatch system, continuously improving the assignment quality.

In the past few years, the dispatch system evolves through three phases. With courier grabbing and dispatcher's manual assignment as the start point (i.e. *phase 0*), the system enabled automatic assignment in a greedy one-by-one manner, i.e. *phase 1*, which soon encountered severe service quality degradation and unacceptable computation time as the rapid growth of order volume. Then the platform upgraded the system into area-level batch assignment mode based on constructive heuristic methods, i.e. *phase 2*, which met the real-time requirements, however, failed to ensure the assignment quality as the daily order volume increases. Now the system has evolved into *phase 3*, which realizes citywide global optimal order assignment (OA) via operations research (OR) and machine learning (ML) techniques, serving as the cornerstone to support over 60 million orders every day. Meanwhile, the research and development group grows into a team with 30-40 people.

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

Specifically, *courier behavior estimation* formulates and solves the courier's pick-up and delivery route planning(RP) problem assuming a new order is assigned, providing the evaluation basis of

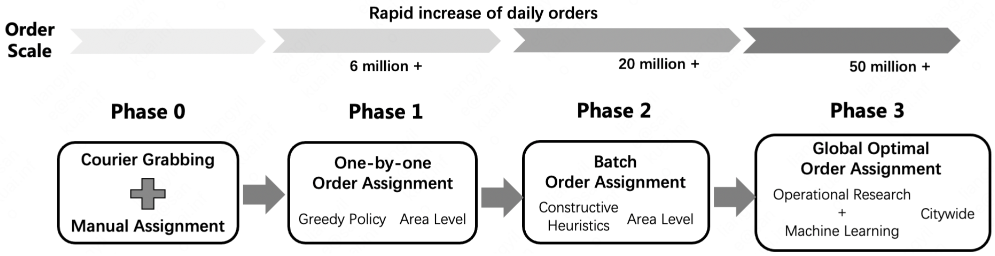
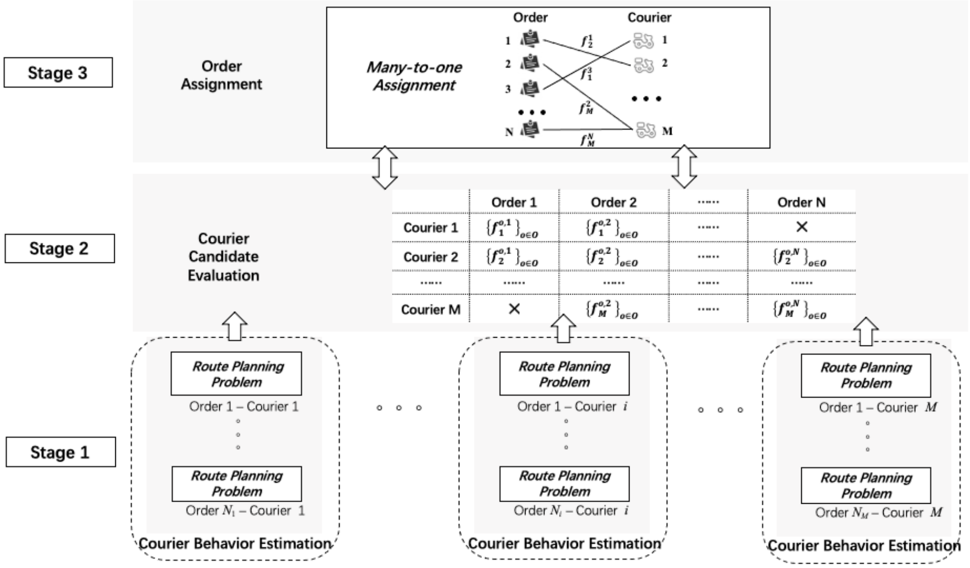


Figure 2 **Evolution of the dispatch system with order volume increase. 1) Phase 0: courier grabbing and manual assignment, 2) Phase 1: area-level one-by-one OA based on greedy policy, 3) Phase 2: area-level batch OA based on constructive heuristics, 4) Phase 3: citywide global optimal OA based on OR and ML methods.**



	Order 1	Order 2	......	Order N
Courier 1	$\{f_1^{o,1}\}_{o\in O}$	$\{f_1^{o,2}\}_{o\in O}$	......	×
Courier 2	$\{f_2^{o,1}\}_{o\in O}$	$\{f_2^{o,2}\}_{o\in O}$	......	$\{f_2^{o,N}\}_{o\in O}$
......	......	......	......	......
Courier M	×	$\{f_M^{o,2}\}_{o\in O}$	......	$\{f_M^{o,N}\}_{o\in O}$

Figure 3 **Dispatch procedure implemented every 30 seconds at a city level. 1) courier behavior estimation formulates and solves the courier's RP problem, providing an evaluation basis for the next stage. 2) courier candidate evaluation calculates the matching degree(MD) scores between the new orders and couriers based on the results of 1). 3) OA generation formulates and solves the OA problem to optimize the global MD scores, satisfying each stakeholder.**

the next stage. *courier candidate evaluation* calculates the MD scores between each new order and its available couriers, reflecting the requirements of each stakeholder. For example, "time score" represents the overtime severity 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.

The remainder of the paper is organized as follows. In *Problem and Challenge* section, we describe the assignment problem formulation, properties, and challenges we encounter. In *Technical Solution*, we introduce our methods at each stage of the dispatch procedure aiming to continuously improve assignment quality. Online A/B test and large-scale offline simulation are conducted to validate the effectiveness. In *Benefits*, the overall effects of the dispatch system are summarized from the view of the key stakeholders, business and finance, environment, as well as the role in response to COVID-19. In the last section, we introduce the extended use of the system at Meituan, and analyze the transportability of our technical solutions proposed in the system.

Problem and Challenge

Problem Formulation

In essence, the sequential dispatch process for OA across a day shown in Figure 4 can be formulated as a multi-period multi-objective combinatorial optimization problem, which is defined in (1) of the Appendix. The current dispatch decisions will affect the OA results in the next dispatch period, through changing the courier's status. Furthermore, the platform pursues global spatial-temporal optimality, i.e., maximum MD scores of all orders and the couriers assigned in a whole day, i.e., $\{\sum_{t=1}^{T} F_t^{o*}\}_{o=1}^{O}$, instead of a single or a few of dispatch periods, i.e., $\{F_t^{o*}\}_{o=1}^{O}$. Explanations for the notations can be found in the Appendix.

Property and Challenge

Modeling and solving the above multi-period multi-objective decision problem in an online fashion is no easy task. According to the properties of on-demand food delivery service and OA problems, the technological challenges we encounter are as follows:

Dynamic and Sequential Decision Process In our scenario, dispatch is a dynamic and sequential decision process, which executes every 30 seconds in a city level, shown in Figure 4. At each dispatch moment, the system collects the newly arrived orders and the status of available couriers, then decides the best matching of the orders and available couriers. Since the system pursues global spatial-temporal optimality and the matching results of the current moments will

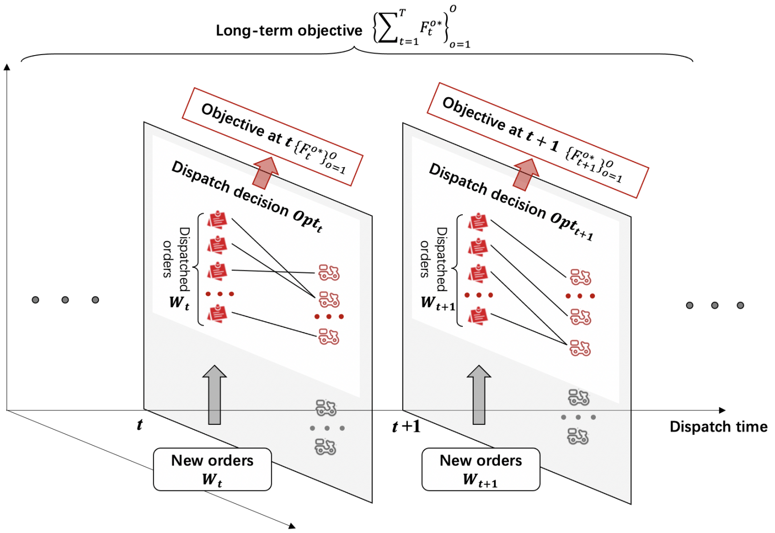

Long-term objective $\left\{\sum_{t=1}^{T} F_t^{o*}\right\}_{o=1}^{O}$
Objective at t $\{F_t^{o*}\}_{o=1}^{O}$
Dispatch decision Opt_t
Dispatched orders W_t
t
New orders W_t
Objective at $t+1$ $\{F_{t+1}^{o*}\}_{o=1}^{O}$
Dispatch decision Opt_{t+1}
Dispatched orders W_{t+1}
$t+1$
New orders W_{t+1}
Dispatch time

Figure 4 Sequential dispatch process. At each dispatch moment, the dispatch system collects the newly arrived orders and the status of available couriers, then decides the best matching of the orders and available couriers.

directly affect the results afterwards, it is necessary to consider the influence of future information in the succeeding time steps to avoid greedy decisions and achieve long-term global optima.

However, it is difficult to directly predict the actual influence of the future. On one hand, the exact distributions of orders and courier status in the future are usually stochastic. On the other hand, the behaviors of the couriers and the real-world environment are full of uncertainties, which further complicates the problem. Building a highly-precise simulation system is too expensive and difficult for our scenario.

Multi-objective Balance As described above, OA should balance the various goals of different stakeholders. Consumers want to get their food on time and intact, merchants hope their food served fresh to satisfy their consumers, couriers need to deliver enough orders with less labor, the platform pursues to operate the 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, trade-offs should be carefully made among them, to achieve long-term optimality of the whole system and all the stakeholders.

A common method to handle these multi-objective problems is to combine the multiple objectives into a single merged objective with weights, known as scalarization methods (Gunantara 2018). However, finding and calibrating the proper weights on different objectives is laborious and non-trivial. The relative scales of different objectives change dramatically during different periods of each day. Thus the weights should be adaptively updated to ensure long-term optimality. But there is a lack of theoretical methods to guide the selection of such weights, especially in such an online and dynamic setting to guarantee long-term optimality.

Full of Uncertainties Uncertainty is an inevitable characteristic of the food delivery process. Unexpected cases, such as traffic jams, extended food preparation time of the merchants, and couriers' personal preferences, will greatly affect couriers' behaviors, and the service quality of the related on-hand orders. However, these uncertainties, can neither be eliminated nor accurately sensed by the system. Moreover, the lack of adequate real-time monitoring of couriers' and merchants' status will further increase the uncertainties.

Due to the existence of uncertainties, it is impossible for the system to make accurate point estimations on couriers' behaviors, e.g., when the order will be picked up or delivered by the courier. Modeling the MD score and making dispatch decisions based on the inaccurate estimation results will weaken the effectiveness of the decision, thereby damaging the experience of consumers, couriers, and merchants. Take the food preparation time as an example, once the food is prepared slowly, the courier has to wait in the restaurant until it is finished, which may cause subsequent orders timeout and make conflicts between the couriers and the merchants. And the impacts of uncertainties will be further intensified by the couriers' on-hand workload and bad weather.

Solving Large-scale NP-hard Integer Programming in Real-time For each dispatch period, the many-to-one assignment problem is NP-hard. And it is quite different from the one in a ride-hailing platform which is widely accepted as challenging. For ride-hailing, each driver is usually assigned only one order each time, while in ours, couriers are often assigned more than one order each time, even over 5 orders at noon peak. Hence, the scale of our problem is significantly larger than the one-to-one problem. For example, the number of decision variables for a one-to-one assignment problem with N assignment objects is of the order of N^2, while the number of decision variables for a five-to-one assignment problem with N assignment objects is of the order of N^6. Furthermore, the MD scores between the couriers and orders are non-additive, namely, 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 manner, which further brings severe challenges to algorithm design.

Since couriers are usually traveling fast, the OA problem must be solved in less than 10 seconds to ensure assignment quality, keeping the consistency of the courier status during the information acquisition period and assignment generation period. Traditional supply-chain optimization problems, even though suffering from large-scale searching space, they are allowed to be solved in hours. Thus their methods can not be directly adopted in the field of on-demand delivery.

Technical Solution

Algorithm Structure of the Dispatch System

To resolve the above challenges, the dispatch system firstly decomposes the original multi-period multi-objective stochastic assignment problem into a series of single-period single-objective deterministic sub problems, which can be solved independently at each dispatch moment, shown in (2) of the Appendix, through weight and matching score design. Then it develops effective algorithms for solving the sub problem 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 *courier behavior estimation* stage, the dispatch system updates *the modeling and solution methods of the local courier's RP problem*, in order to continuously improve the route consistency rate between the planning routes and the real ones in the uncertain and time-varying environment. It provides a reliable evaluation basis for the evaluation stage.

In *courier candidate evaluation* stage, on one hand, the system adopts an *online weight adaptation mechanism* realizing the temporal decomposition and multi-objective integration. It aims to guild the system states gradually evolving into an acceptable long-term balance among each objective. On the other hand, the system develops a *robust and adaptive matching score calculation method* to resolve the problems brought by uncertainties in the delivery process. Hence, it transforms the OA problem at each dispatch moment into a single-objective and deterministic one.

In *OA generation* stage, the system upgrades the algorithm to continuously improve the solution quality and computational efficiency. In addition to traditional OR methods, we adopt ML techniques to enhance the algorithm performance. One example is that we effectively prune the searching space and greatly improve the algorithm (both solution quality and computational efficiency), via graph neural network (GNN) methods to learn effective order combinations from experienced couriers' behaviors.

The execution of the dispatch algorithm at each period is shown in Figure 5. Since we cannot compute the MD scores between arbitrary order combinations and couriers, the behavior estimation and evaluation of the courier candidates is only executed for some promising order combinations at each iteration, which are determined by the searching mechanism of the algorithm.

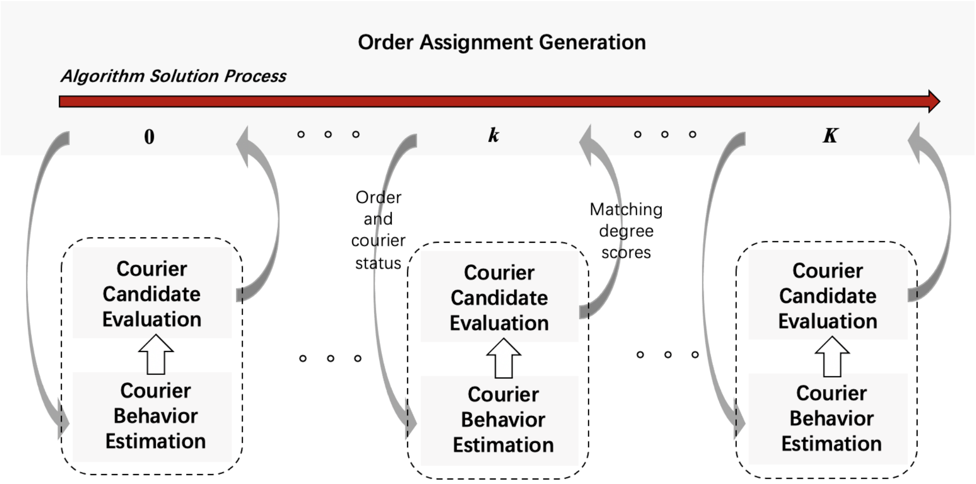

Order Assignment Generation
Algorithm Solution Process
0
k
K
Order and courier status
Matching degree scores
Courier Candidate Evaluation
Courier Behavior Estimation
Courier Candidate Evaluation
Courier Behavior Estimation
Courier Candidate Evaluation
Courier Behavior Estimation

Figure 5 Dispatch algorithm execution at each period: along with the searching and iteration of the OA algorithm, the behavior estimation and evaluation of the courier candidates for part of order combinations are executed sequentially.

The major algorithm components of the dispatch system are introduced as follows.

RP: Domain Refined Heuristics with Inverse Reinforcement Learning

Different pickup and delivery routes for the same courier, can result in different delivery distances as well as delivery time for each order. Figure 6 shows a simple example for different RP results. An inappropriate route implies further distance as well as longer time. 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 48.3 million times per minute. These together raise demanding challenges for both the quality of the planning result and the time efficiency of the algorithm.

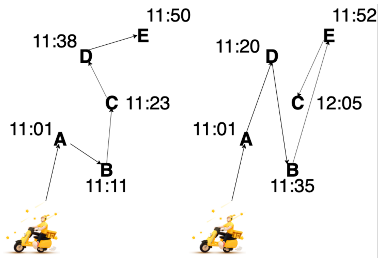


Figure 6 **An example for the RP. Different routes can result in different total delivery time (delivery ends at 11:50 and 12:05) and distances for the same courier.**

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

Heuristic Search Based RP 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. It is developed to find the optimal route which 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, precedence constraint and capacity constraint. The former means each order should be picked up before being delivered, and the latter suggests the orders should not exceed the total capacity of a courier.

The initialization stage is a greedy insertion procedure. We first sort the orders according to their ETA, 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 tempt to move delivery points with delays forward or otherwise, as illustrated in Figure 7. 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) in 97% cases within 10 milliseconds.

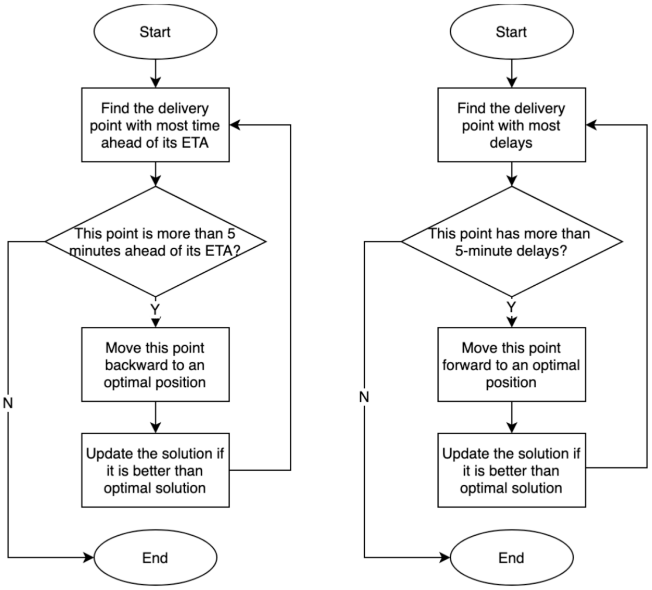

Start
Find the delivery point with most time ahead of its ETA
This point is more than 5 minutes ahead of its ETA?
Y
Move this point backward to an optimal position
N
Update the solution if it is better than optimal solution
End
Start
Find the delivery point with most delays
This point has more than 5-minute delays?
Y
Move this point forward to an optimal position
N
Update the solution if it is better than optimal solution
End

Figure 7 **The local search stage. We find the delivery point with most time ahead of its ETA and try to move it backward in order to find a better solution (Left). In the other case, we try to move the point with most delays forward (Right). The optima are updated using these local search operations.**

RP 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 over ETA (Zheng et al. 2019). However, it is obvious that the manual function can hardly depict how couriers make decisions during the

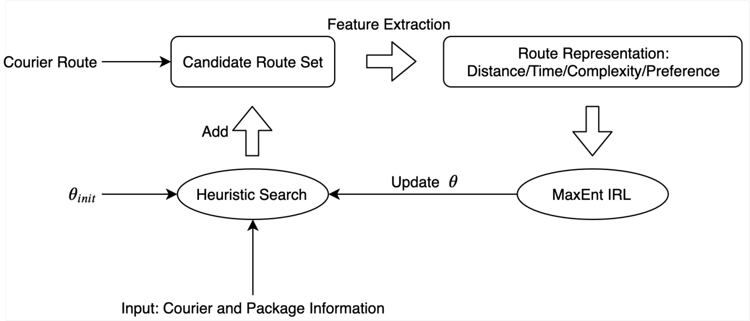


Figure 8 The IRL pipeline. The IRL pipeline is an iteration process based on the courier and package information. The heuristic search is used to generate a candidate route set R based on the cost function parameter θ_{init}. Route features are extracted and used in the following feature-based MaxEnt IRL. The main goal of the IRL is to learn the parameter θ which maximizes the probability of the route which the courier employs in the reality. Thereafter, the updated θ is used in the next iteration.

delivery process. Therefore, there is a fairly high chance that couriers will not deliver orders as expected and the planned routes are not even usable, especially for an uncertain and time-varying environment.

To ensure the consistency between planned routes and couriers' actual delivery ones, we propose to learn the objective function from couriers' historical delivery routes. Specifically, a feature-based maximum entropy (MaxEnt) IRL (Ziebart et al. 2008, Finn et al. 2016, Kuderer et al. 2015) is used to learn a reward function (objective function $g(r)$ in our case) from couriers' 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 probabilities. 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 $Z_\phi = \sum_{r \in R} e^{-g(r)}$ is the partition function and R is the set of all possible routes. Figure 8 presents an illustration for our learning pipeline. Given the courier and order information, the heuristic search can produce a set of possible delivery routes during the local search, as well as the best route with the minimum cost. In our solution, a linear cost function is used,

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

where $G(r)$ is the feature vector of the route r and θ is the feature weights to be learned. After the heuristic search is finished, 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 which the courier employs in the reality. The parameter θ is then updated and used in the next run of heuristic search. A fixed iteration number and early stopping strategy can be used in the training considering the training dataset size. In practice, two main problems still exist, namely the feature representation of the route and the rationality of using the candidate route set in IRL. For the representation of the route, we propose tens of route features based on our domain knowledge. Generally, four kinds of route features are used, including delivery distance, delivery time, route complexity and courier's preference. For example, the total navigation/line distance and the average navigation/line distance for pickup/delivery points are used in our implementation. For rationality of the candidate route set, we should sample all possible routes in an ideal IRL setting, which is, however, impossible as the number of orders increases. Instead, we take advantage of the information in the local search stage of the heuristic search. Large-scale temporary route samples are generated during the local search. This temporary route set can cover a large proportion of the routes with high probability. In that case, we treat this set as the possible route set R, and replenish it by adding new routes during the training.

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

MD Score Calculation: Adaptive Risk Decision-making through Bayesian Optimization

Considering the inevitable uncertainties of delivery process, and the impacts of resulting invalid dispatch decisions, risk management is introduced to improve decision effectiveness and robustness. One widely used tool is conditional value-at-risk (CVaR) (Tamar et al. 2015), which is a statistical measurement quantifying the worst-case risk of the performance distribution. Moreover, to evaluate the MD for the courier and orders under normal circumstances, the average performance is also important. Therefore, MD score should comprehensively consider the courier's performance in both high-risk and average states as shown in (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 difficulties in MD score evaluation. On one hand, real-time decision-making needs to be done in milliseconds, while the calculation of risk measurements involves large-scale sampling and repeated RP calculation for each sample. Effective sampling method should be adopted to reduce complexity at the same time guarantee the accuracy of the tail risk. On the other hand, the system has different risk attitudes in different scenarios. Achieving adaptive risk aversion degree with minimum trial and error cost online is also an urgent issue.

The process of matching score evaluation can be divided into two parts:

Calculation of risk measurements. Without loss of generality, we only take the mealtime uncertainty into account. As shown in Figure 9, assume a courier is assigned L orders in total, the mealtime of the l-th pickup node, denoted as t_{ml}, obeys 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 millisecond-level calculation. Next, considering that the execution time of nodes in sequence are highly correlated with each other, we integrate the point-wise mealtime samples into route samples through route conduction. 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 measurements under two states. 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 real 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

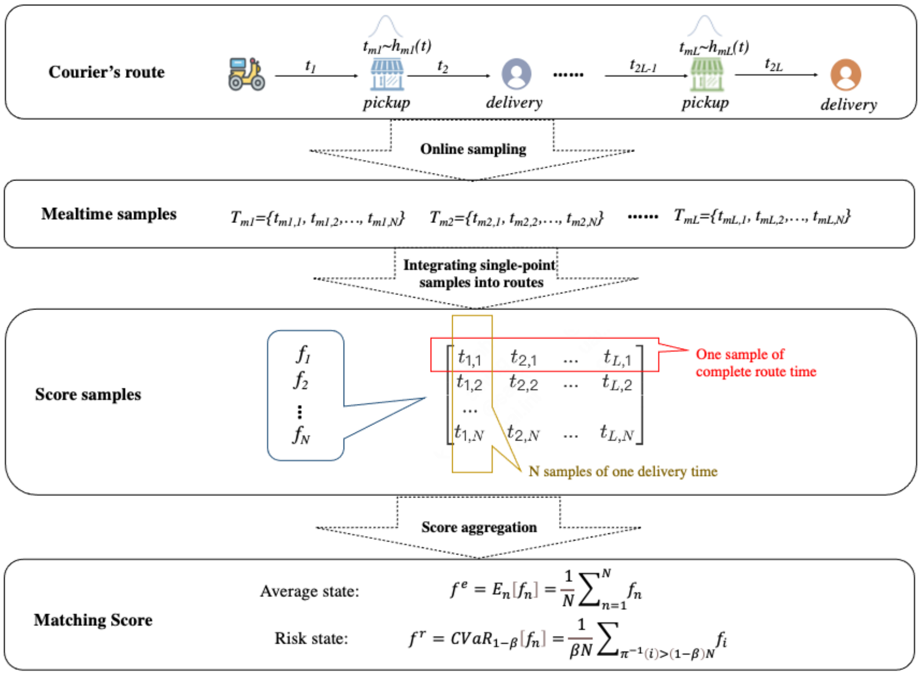

Courier's route
t_1
$t_{m1}\sim h_{m1}(t)$
pickup
t_2
delivery
......
t_{2L-1}
$t_{mL}\sim h_{mL}(t)$
pickup
t_{2L}
delivery
Online sampling
Mealtime samples
$T_{m1}=\{t_{m1,1}, t_{m1,2},\ldots, t_{m1,N}\}$
$T_{m2}=\{t_{m2,1}, t_{m2,2},\ldots, t_{m2,N}\}$
......
$T_{mL}=\{t_{mL,1}, t_{mL,2},\ldots, t_{mL,N}\}$
Integrating single-point samples into routes
Score samples
f_1 f_2 ⋮ f_N
$\begin{bmatrix} t_{1,1} & t_{2,1} & \cdots & t_{L,1} \\ t_{1,2} & t_{2,2} & \cdots & t_{L,2} \\ \cdots \\ t_{1,N} & t_{2,N} & \cdots & t_{L,N} \end{bmatrix}$
One sample of complete route time
N samples of one delivery time
Score aggregation
Matching Score
Average state:
$f^e = E_n[f_n] = \frac{1}{N}\sum_{n=1}^{N} f_n$
Risk state:
$f^r = CVaR_{1-\beta}[f_n] = \frac{1}{\beta N}\sum_{\pi^{-1}(i)>(1-\beta)N} f_i$

Figure 9 Calculation procedure of risk measurements can be divided into three steps: online sampling of mealtime, route sample integration, MD score aggregation.

specific analytical functional forms. Therefore, the idea of Bayesian optimization, namely, black-box optimization (Frazier 2018) is introduced.

As shown in Figure 10, in the offline training phase, we choose 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 variable, 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 (every 30 minutes in practice) is selected based on the upper confidence bound (UCB) (Contal et al. 2013) criterion, with the principle of 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 degree value with minimum trial and error cost, and ensure the stability and decision feasibility of the online dispatch system.

Our risk decision-making system with adaptive risk aversion degree has undergone AB testing in 9 cities across China, where the optimal degree factor of each time period fluctuates between

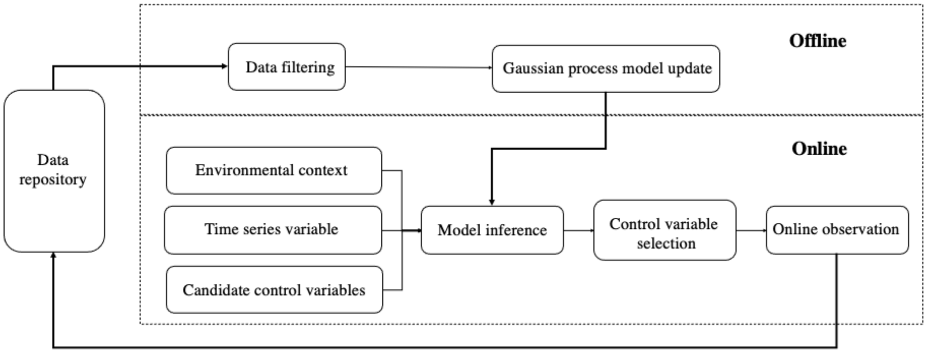


Figure 10 **Risk aversion degree adaptation consists of two iterative phases, offline model training and online decision-making.**

0.5 and 0.95. Evaluation results show that the on time rate increases by 0.27pp, and the average travel distance reduces by 1.20%.

Dynamic Objective Weight Adaptation Combining Online Decision with Offline Planning

To balance different dispatch goals, a target-oriented multi-objective balancing framework inspired by Lyu et al. (2019) is implemented to adaptively adjust the weights of different objectives during the sequential dispatch process.

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

Since the values of objectives change dramatically during the daily horizon, we divide each day into a series of dispatch periods. $U^o_{p_t}$ is the "ideal point" of objective o during 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 get weights for the next time step $t+1$, exponential smoothing is used to average the gaps from previous time steps. The weight is updated according to $\eta_{t+1}^o = \gamma(\delta_t^o)^+ + (1-\gamma)\eta_t^o$, where γ is a constant to control the smoothness.

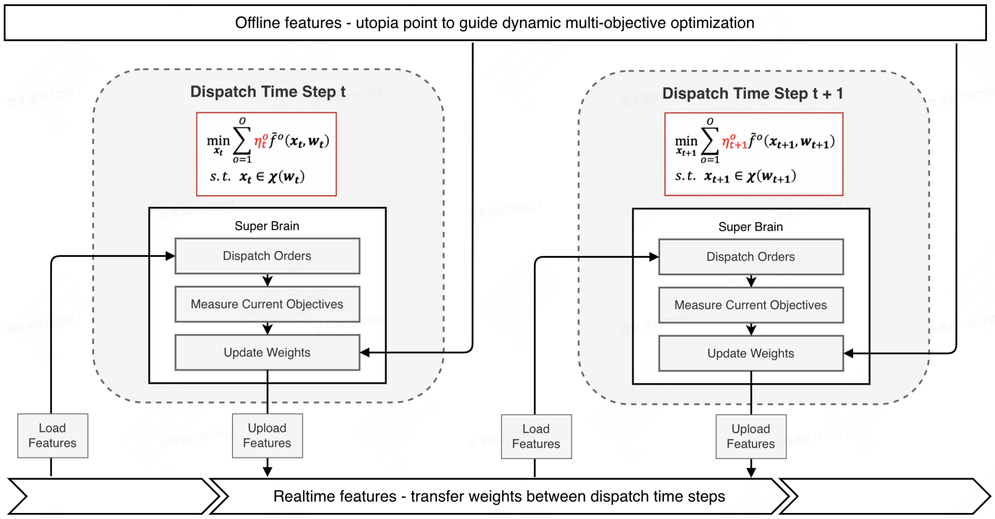

Offline features - utopia point to guide dynamic multi-objective optimization
Dispatch Time Step t
$\min_{x_t} \sum_{o=1}^{O} \eta_t^o \tilde{f}^o(x_t, w_t)$
$s.t.\ x_t \in \chi(w_t)$
Super Brain
Dispatch Orders
Measure Current Objectives
Update Weights
Dispatch Time Step t + 1
$\min_{x_{t+1}} \sum_{o=1}^{O} \eta_{t+1}^o \tilde{f}^o(x_{t+1}, w_{t+1})$
$s.t.\ x_{t+1} \in \chi(w_{t+1})$
Super Brain
Dispatch Orders
Measure Current Objectives
Update Weights
Load Features
Upload Features
Load Features
Upload Features
Realtime features - transfer weights between dispatch time steps

Figure 11 **Implementation of the multi-objective balancing framework. 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 objective weights are updated based on the gaps between current objectives and ideal points.**

Our current implementation is demonstrated in Figure 11. 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 for different goals.

This framework is implemented in the dispatch system, and AB tests are done to measure its performance. It is tested in 5 cities in China, where the on-time rate increases by 0.33pp, and the average travel distance reduces by 0.92%. During the experiment, the daily cumulative MD score of the experimental group is optimized by 3.6%-4.5%.

OA: OR Methods Combined with ML

The solution quality and computational efficiency of the OA algorithm are of great importance. The dispatch system improves traditional OR methods via ML techniques and explores several ways to optimize the algorithm performance.

Method 1: ”Divide-and-Conquer” Framework with Imitation Learning and GNN

Algorithm Framework For solving many-to-one assignment problems in a real-time manner, the dispatch system develops an enhanced ”divide-and-conquer” solution framework, i.e., decomposes the original large-scare many-to-one assignment problem at each dispatch moment into a series of one-to-one sub problems which are solved sequentially, resulting in a progressive solution procedure online. Each of the sub-problems is quite simple and can be solved in polynomial time, thus greatly reducing the computation burden. To guarantee solution quality the decomposition and iteration rounds are carefully guided and controlled by a global coordinator, which is a 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. The ”divide-and-conquer” framework is depicted in Figure 12.

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. Only limited MD scores are calculated at each iteration, thus greatly reducing the calculation volume of the previous stages. Moreover, the sub problem at each iteration degrades to a polynomial-time one and the scale is much smaller than the original one, which can be solved quite easily by traditional OR methods, such as the Kuhn-Munkras algorithm Munkres (1957).

Therefore, within this framework, choosing orders to assign at each round, namely, the way to realize decomposition, is of great importance to the solution quality. With well-designed decomposition strategies, the methods can achieve better solution quality with limited MD score calculation

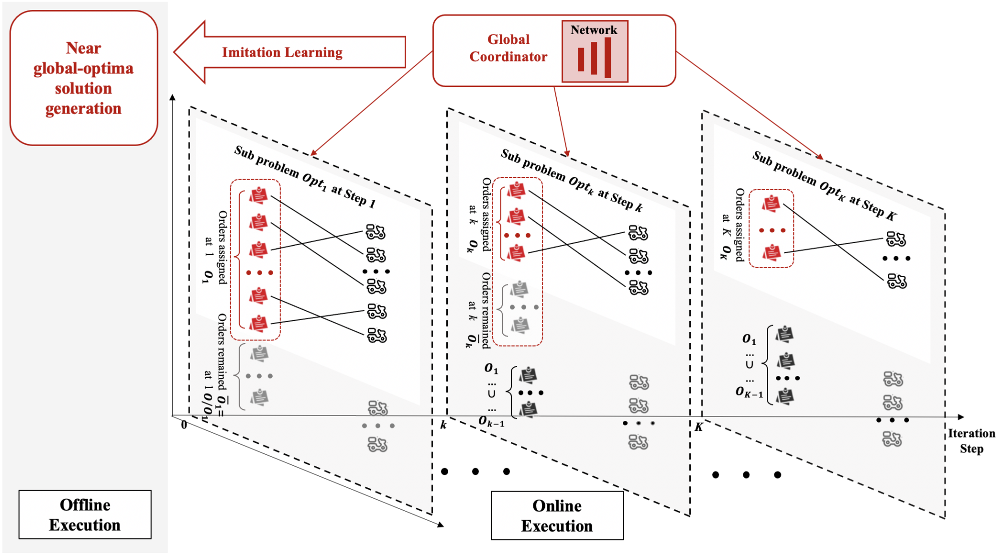

Near global-optima solution generation
Imitation Learning
Global Coordinator
Network
Sub problem Opt_1 at Step 1
Orders assigned at 1 O_1
Orders remained at 1 $\bar{O}_1 = O/O_1$
Sub problem Opt_k at Step k
Orders assigned at k O_k
Orders remained at k $\bar{O}_k$
$O_1 \cup \dots \cup O_{k-1}$
Sub problem Opt_K at Step K
Orders assigned at K O_K
$O_1 \cup \dots \cup O_{K-1}$
0
k
K
Iteration Step
Offline Execution
Online Execution

Figure 12 **"Divide-and-Conquer" framework by IL: the original many-to-one assignment problem is decomposed into a series of one-to-one sub-assignment problems which are solved sequentially. A global coordinator is trained offline by IL methods with high-quality solutions as its labels. And it controls and guides the online solution procedure.**

volume, which is polynomial-related or even linearly-related to the order scale, instead of combination explosion. Meanwhile, the algorithm for solving each sub problem is of polynomial computing 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 by IL method offline and uses expert solutions from enhanced tabu search method as its labels.

During the offline training stage, the traditional tabu search method is optimized to improve its solution quality, by adding several well-designed operators, including the double shift operator, the long chain operator, and so on. Based on the expert solutions generated by the enhanced tabu search method, we employ the IL method 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, i.e., orders with higher average optimal matching probabilities will be selected to be assigned at the current iteration. Since the coordinator extracts

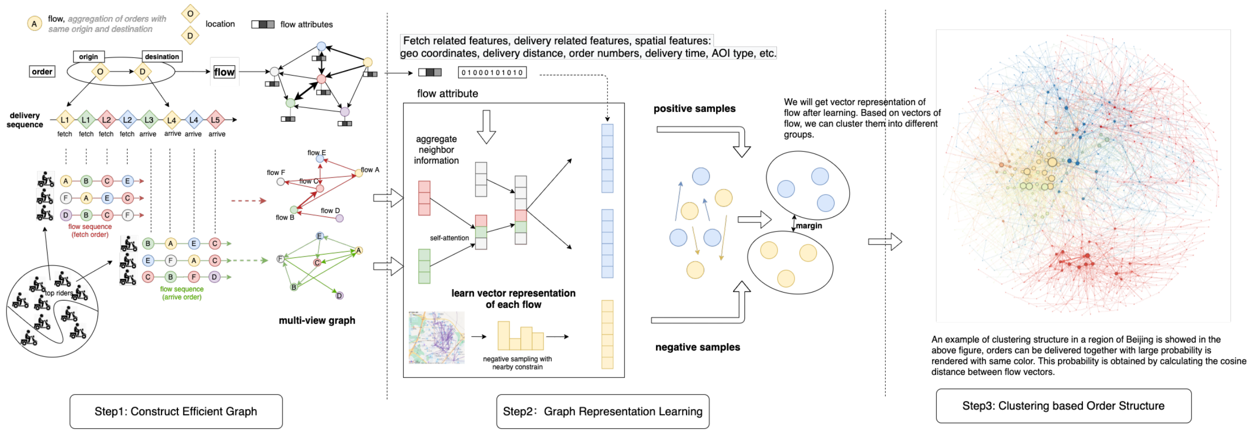

A
flow, aggregation of orders with same origin and destination
O
D
location
flow attributes
order
origin
desination
O
D
flow
delivery sequence
L1
L1
L2
L2
L3
L4
L4
L5
fetch
fetch
fetch
fetch
arrive
arrive
arrive
arrive
flow sequence (fetch order)
flow sequence (arrive order)
top riders
flow E
flow F
flow C
flow A
flow B
flow D
multi-view graph
Step1: Construct Efficient Graph
Fetch related features, delivery related features, spatial features:
geo coordinates, delivery distance, order numbers, delivery time, AOI type, etc.
01000101010
flow attribute
aggregate neighbor information
self-attention
learn vector representation of each flow
negative sampling with nearby constrain
positive samples
negative samples
margin
We will get vector representation of flow after learning. Based on vectors of flow, we can cluster them into different groups.
Step2: Graph Representation Learning
An example of clustering structure in a region of Beijing is showed in the above figure, orders can be delivered together with large probability is rendered with same color. This probability is obtained by calculating the cosine distance between flow vectors.
Step3: Clustering based Order Structure

Figure 13 **Framework of modeling delivery network via GNN. It contains three steps: 1) delivery network construction: constructing an efficient delivery graph with flow (aggregation of orders with same origin and destination) as basic unit by merging massive delivery records from experienced couriers; 2) graph representation learning: applying GNN to model the graph and extract combination patterns from it as representative vectors of each flow; 3) online order combination: using representative vectors to cluster orders into various groups to decompose assignment task, and work with following dispatch algorithm to achieve better matching results.**

global information of the expert solutions, the algorithm can effectively avoid local greed, moreover, it can search more widely and deeply of the searching space, thus closing to expert solution quality.

The proposed algorithm significantly improved assignment quality by optimizing the solutions' objective values by 5-10%, and with the growth of the order volume, the optimization degree is even higher.

Online Order Combination: learning from experienced courier via GNN In order to reduce the iteration rounds and further improve the computation efficiency without sacrificing solution quality, it is necessary to construct an efficient online order combination mechanism, making the sub-problem at each iteration generate many-to-one assignments. The mechanism should be able to recognize the complicated relationship between different orders and find effective ways to combine them as different uncoupled groups.

An intuitive solution is to group orders with nearby origin-destination and delivery time-window 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 trying to deliver orders in practice, we find that many couriers can group various orders flexibly with their personal perceptions and fruitful experiences learned during working, which is more complicated than the static grouping methods.

Thus, we further develop a GNN-based framework by directly learning knowledge from experienced couriers to guide the online order combination. With the learning-from-human mechanism, this method can discover more delivery patterns and cluster orders with effective and flexible ways of solving the assignment problem. The whole framework of the proposed method is presented in Figure 13. Detailed designs of three steps are described as follows.

Delivery Network Construction. We first group orders with the same origin and destination as flow. Following the experience from couriers, we pre-process 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 relation in sequence as a link, we can merge these flow sessions into a union graph as a delivery network. Furthermore, each flow node also owns fruitful attributes to describe its crucial character, e.g., 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. We aim to learn the vector representation of the node which contains the attributes and the topology information of it in the graph after training. We apply classic graph learning models Wang et al. (2018), Hamilton et al. (2017) to learn the basic vector representation of flow node, and some well-known tricks Bengio et al. (2009) are also applied to optimize the quality of vector representation.

Online Order Combination. After obtaining the vector representation of the flow node, we define a similarity-based mechanism to combine the orders together. Basically, we use the dot-product results of vector representations of two flow nodes as the similarity metric of combination. Similar flow nodes with large dot-product results can be regarded as a group to participate in the assignment problem. As the right part in Figure 13 shows, we use this similarity metric to cluster the historical orders into communities and validate the responsibility of clustering results by humans in various cases.

Method 2: Multi-stage Sub-problem Reformulation with ML

We also propose a multi-stage solution algorithm based on neighborhood search and ML to solve the many-to-one assignment problem at each dispatch moment. The algorithm constructs sub problems by adding constraints to the original one, which limits the variations in the solution space of the current stage sub problem compared to the previous stage. Specifically, one of the sources of the complexity of the original many-to-one assignment problem is that orders and orders can be arbitrarily batched and assigned to the same courier (the number of batched groups grows exponentially with the number of orders in the groups). To reduce complexity, the main constraints added to the sub problem are the selections of the new 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 many-stage sub problems sequentially, an approximate solution to the original problem is obtained.

One of the key aspects of the proposed many-stage 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. For the rule-based method, expert knowledge is used to generate various constraints. For example, we can have the rule that only the orders that have near pick-up points can be batched and assigned to the same courier, or that each order can only be assigned to the courier when the order's pick-up point is close to the courier's current location. Different rules will affect the construction of sub problems, which in turn affects 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, the branch-and-bound method is used to solve the many-to-one assignment problems offline and the resulting batched groups are used to train an XGBoost-based classification model Chen et al. (2015) to predict whether the batched groups will appear 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 sub problem (i.e. the computation time of $f_{t,\bar{w}}^{o,r}$) can be completed within seconds.

Although we have limited the complexity of the sub problems 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) where 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 allows for efficient searching and high-quality solutions.

An illustrative example of the proposed method is shown in Figure 14. Noted that the proposed multi-stage solution algorithm shares some merits with 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 making the solution trajectory quickly approach the optimal solution. Typically, only 3-5 stages are needed to converge to a satisfactory near-optimal solution and the total construction and solution time is within 10 seconds.

Reassginment: Technical Solutions for Abnormal Scenario

In most cases, the courier who accepts the assignment is expected to deliver the assigned order. However, since we assign orders tens of minutes before their arrival, we cannot predict all the potential incidents that couriers may encounter during the actual delivery process. If unforeseen or extremely tail-end incidents 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 takes too long, the courier has to wait for the order at the merchant, which will delay the arrival time of the multiple orders. We use the reassignment scheme to address this problem: based on the latest environmental and estimated information, we reassign orders

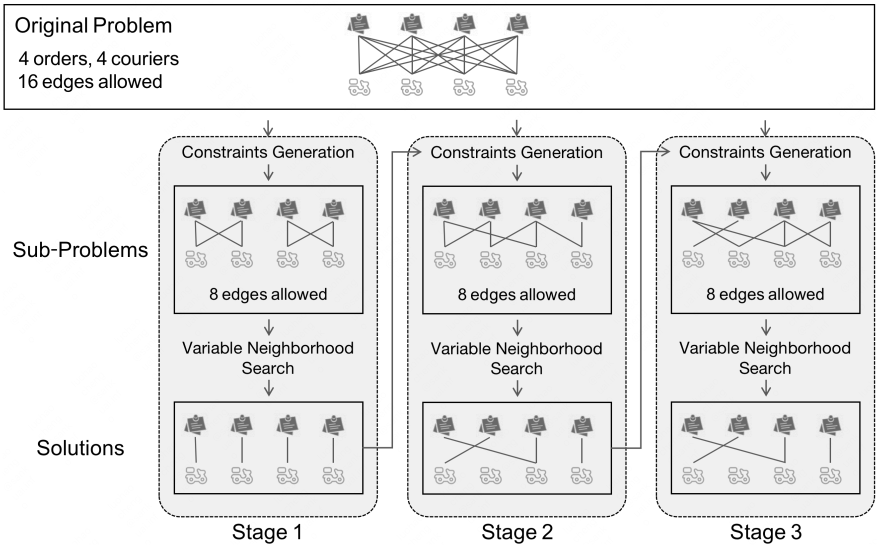


Figure 14 An illustrative example using the proposed multi-stage solution algorithm, where 4 orders and 4 couriers are considered. At each stage, the constraints generation method is utilized to add constraints to the original problem, formulating a sub problem with smaller complexity (i.e., fewer edges allowed). The constraints generation 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 contains the optimal solution if possible. The proposed variable neighborhood search algorithm is then implemented to solve the sub problem. The overall method stops when no more new sub-problems can be formulated or solutions of consecutive stages remain the same. In this example, the method stops after 3 stages.

that have not yet been picked up to other appropriate couriers if it 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 other is initiated by the courier. Both types have similar processes:

Step 1: Collecting orders for reassignment. For those initiated by the system, the orders that are worth reassigning are identified by an online algorithm that monitors the delivery process of couriers. Specifically, we run RP algorithms every minute to estimate the delivery sequences and distances of the courier in the case of completing all the assigned orders, as well as the cases where one of the orders is removed from the courier. By comparing the estimated delivery processes of the courier before and after reduction, we choose the order that is most beneficial for delivery efficiency as the order needed to be reassigned. Additionally, couriers can also request reassignment through our app if they encounter abnormal incidents that negatively affect their delivery processes.

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

Step 3 (skip for the reassignment initiated by the courier): Reach out to the courier currently handling the order through a pop-up prompt on our app to inquire if they agree to the reassignment. If they decline, we will halt the reassignment process.

Step 4: Reach out to the courier who is considered as a more appropriate one for the reassigned order through our app to inquire if they are willing to take on. If accepted, we will complete the reassignment process. If declined, the reassignment process will be stopped.

Through reassignment, about 1% of orders will be completed by new couriers found through reassignment. The A/B experiment shows that the delivery intervals for these reassigned orders are on average shortened by over 1 minute.

Benefits

The dispatch system has brought great benefits, including business development and financial benefits for the company, optimizing courier, consumer, and merchant experience, improving environmental impact, and helping respond to the COVID-19 pandemic.

Business Development

Since the company was listed in Hong Kong in 2018, Meituan, as one of the world's largest online and on-demand delivery platforms, has grown greatly. Compared to 2019, the number of cities covered increased by 40% and the number of registered users increased by 60.76%. Meanwhile, 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 increased by 100% to about 60 million.

As the number of transactions increases, the dispatch problem becomes more challenging, especially in peak hours when about 20 million orders could be assigned to about 5 million couriers.

Table 1 Variations in Different Metrics

Indicators of Different Stakeholders	After vs. Before
No. of Cities Covered	up 40.00%
No. of Active Couriers	up 32.08%
No. of Merchants	up 55.93%
No. of Consumer	up 60.76%
Daily Transactions Peak	up 100.00%
Average Delivery Time per Order	down 20.96%
Average No. of Orders Delivered by Couriers per Day	up 109.63%
Average Riding Distance per Order	down 23.77%
Average Meal Waiting Time (fetch_meal_time - push_meal_time)	down 16.00%

However, armed with the dispatch system, the assignment results can be provided within seconds at each dispatch period, i.e., every 30 seconds. And numerical experiments show that the algorithms can achieve an average optimality of 90% compared with the best-known results obtained offline with sufficient computation time.

Courier and Consumer Experience

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

For consumers, the time from placing an order to receiving the delivery decreased by 20.96% with the dispatch system deployment.

For couriers, the average number of daily delivered orders per full-time couriers has increased by more than 109% with the help of the dispatch system. In addition, because of the intelligent system, the order received by each courier is more smooth. Therefore, the couriers traveling distance per order before is about 2 km, and the distance after is about 1.5 km, reduced by about 0.5 km for delivering an order.

For merchants, the reduction in average delivery time per order is also good news. Besides, the time it takes for food to be picked up decreased by 16.0%. Therefore, the service provided to merchants is also becoming better.

Financial Benefits

Up to now, the system can contribute to a daily decrease of $0.64 million in the cost of Meituan, which amounts to about $0.23 billion cost reduction for a year.

Environmental Impact

With the application of the dispatch system, we can reduce about 532,125kg of CO2 emissions per day. The calculation formula is as follows: the CO2 reduction per day = the average delivery distance reduced per order * order num per day * CO2 emission per kilometer by e-bike. In the above formula, the first item is 0.5375km. The second item is about 60,000,000. As for the third item, the carbon footprint of an e-bike in China is about 16.5g of CO2 per kilometer, and the detailed estimation steps of the carbon footprint of an e-bike are in Appendix. Thus, we obtain the final CO2 emission reduction result, 532,125kg (CO2 emissions reduction per day) = 0.5376km/order × 60,000,000 order × 0.0165kg/km.

Response to COVID-19

The dispatch system helped the company 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 has surged, however, the supply of couriers has dropped a lot. Thus, the supply and demand ratio has become extremely tight, and a considerable portion of the orders was canceled. By adapting the system flexibly, including losing ”irresponsible refusal” and ”higher incentives” for couriers, relaxing the consideration of punctuality, and transforming the objective to achieve the best completion rate, better experiences for consumers and couriers have been achieved in many cities where severe epidemics have occurred.

Extending the Use of the Dispatch System

The vision of Meituan is to help people eat better, and live better. In this sense, the value of the dispatch system is even higher. It enables the thriving of other new business formats of the digital economy in Meituan, such as Meituan shopping, Meituan groceries, and Meituan drugs.

And with the rapid development of the digital economy in recent years, real-time assignment problems becomes more and more common and fundamental in many industrial fields. The analytics and OR techniques in the dispatch system are not just limited to the scenarios of OA problems in food delivery. They are highly portable to other industrial scenarios inside or outside of Meituan, such as computational resource dispatching, and online advertisement displaying.

Acknowledgments

We also appreciate the work of Hui Li, Xianying Fan, Sicheng Liu, Huanjia Lian, Ruoyu Zhang, Xingyu Liu, Min Wu for the promotion of the system.

References

Bengio Y, Louradour J, Collobert R, Weston J (2009) Curriculum learning. *Proceedings of the 26th International Conference on Machine Learning*, 41–48.

Bosch-Ebike (2022) `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 Transactions on Intelligent Transportation Systems* 23(10):18603–18619.

Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, Mitchell R, Cano I, Zhou T, 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. *Proceedings of Machine Learning and Knowledge Discovery in Databases: European Conference*, 225–240.

Ding H, Wang Z (2020) Layered sampling for robust optimization problems. *Proceedings of the 37th International Conference on Machine Learning*, 2556–2566.

ECF (2011) `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. *Proceedings of the 33rd International Conference on Machine Learning*, 49–58.

Frazier PI (2018) A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811* .

Gunantara N (2018) A review of multi-objective optimization: Methods and its applications. *Cogent Engineering* 5(1):1502242.

Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. *Proceedings of the 31st Advances in Neural Information Processing Systems* .

Hamilton WL (2020) Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning* 14(3):1–159.

Hansen P, Mladenović N (2001) Variable neighborhood search: Principles and applications. *European Journal of Operational Research* 130(3):449–467.

He H, Eisner J, Daume H (2012) Imitation learning by coaching. *Advances in neural information processing systems* 25.

Kuderer M, Gulati S, Burgard W (2015) Learning driving styles for autonomous vehicles from demonstration. *2015 IEEE International Conference on Robotics and Automation*, 2641–2646.

Kuhn HW (1955) The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2(1-2):83–97.

Lyu G, Cheung WC, Teo CP, Wang H (2019) Multi-objective online ride-matching. *Available at SSRN 3356823* .

Munkres J (1957) Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics* 5(1):32–38.

Pisinger D, Ropke S (2019) Large neighborhood search. *Handbook of Metaheuristics* 99–127.

Statista (2022) `https://www.statista.com/statistics/1300419/power-generation-emission-intensity-china/`.

Tamar A, Glassner Y, Mannor S (2015) Optimizing the cvar via sampling. *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, volume 29, 2993–2999.

Wang J, Huang P, Zhao H, Zhang Z, Zhao B, Lee DL (2018) Billion-scale commodity embedding for e-commerce recommendation in alibaba. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 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. *Proceedings of the Institute for Operations Research and the Management Sciences Annual Meeting*.

Ziebart BD, Maas AL, Bagnell JA, Dey AK, et al. (2008) Maximum entropy inverse reinforcement learning. *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, volume 8, 1433–1438.

The Calculation of the Carbon Footprint of Ebike

the carbon footprint of ebike (16.5g/km) = manufacturing consumption (7g/km) + electricity consumption (3.2g/km) + food consumption (6.3g/km).

1.manufacturing consumption 7g CO2 per km

The ECF (ECF 2011) estimates that an ebike has an average manufacturing carbon footprint of 134kg CO2. With a lifespan of 19,200km, the manufacturing consumption of ebike can be calculated as 7g/km=134kg/19200km.

2.electricity consumption 3.2g CO2 per km

According to Bosch's ebike range calculator (Bosch-Ebike 2022), a city ebike with a 500Wh battery will provide a range of 94km under typical commuting conditions (assuming 22km/h average speed, mountain bike tires, "sports" assistance mode and 85kg combined weight). Assuming a charging efficiency of 90 percent (not all the energy from the plug makes it into the battery), an ebike will require 5.9Wh (=500Wh/94km/90%) of electricity from the grid to travel each kilometer. Furthermore, in China, the carbon intensity of electricity in 2021 is at 0.549g of CO2 per Wh of electricity (Statista 2022). Thus, electricity use is calculated as 3.2g CO2 per km=5.9Whx0.549g/Wh.

3.food consumption 6.3g CO2 per km

The ECF(ECF 2011) assumes an average 70kg cyclist on an ebike will burn only 4.4 extra calories per kilometer over no exercising time. Thus, with the ECF's estimate for food production emissions (1.44g CO2 per calorie), we get 6.3g CO2e per km (=4.4calories/kmx1.44CO2/calories) for food production.

Notations

In this part, we list major symbols and notations used in this paper.

General Notations

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 $x_{\bar{w}}^r = 1$ means order w or order combination $\bar{w}$ 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 $o \in O$

$\bar{w}$ Order combination which contains one or more orders, $\bar{w} \in \overline{W_t}$ where $\overline{W_t}$ is the set of all possible order combinations at time t

$\bar{w}(w)$ 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

The multiperiod, multiobjective dispatch problem can be formulated as

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

where x_t is non-anticipative 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 below:

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

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

RP with IRL

$g(r)$ 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

Adaptive Risk Decision-Making Through Bayesian Optimization

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}, $T_{ml}=\{t_{ml,1},t_{ml,2},...,t_{ml,N}\}$, where N is the number of samples

f_n Matching degree score corresponding to the n-th sample

Dynamic Objective Weight Adaptation

p_t Each day is divided into a series of periods and p_t is the period containing time t

$U^o_{p_t}$ “ideal point” of objective o during period p_t