



Research article

The robust partitioned distribution problem with courier learning and demand uncertainty

Chi Zhang¹, Xue Bai^{2,*}, Guojun Sheng³ and Zhili Zhou¹

¹ School of Management, Xi'an Jiaotong University, Xi'an 710049, PR China

² School of Economics and Management, Chang'an University, Xi'an 710064, PR China

³ National Key Laboratory of Mass Customization System and Technology, COSMOPlat IoT Technology Co., Ltd., Qingdao 266103, PR China

* **Correspondence:** E-mail: baixue@chd.edu.cn; Tel: +86-17765082580.

Abstract: This study focuses on the robust partitioned distribution problem with courier learning and demand uncertainty. The research is divided into tactical (customer point zoning) and operational (courier route planning within zones) stages, for which mathematical models are established. Nonlinear learning curves are linearized via piecewise methods, and the robust partitioned distribution model for demand uncertainty is converted into a robust linear model. An improved adaptive large neighborhood search algorithm is designed; on 13 standard capacitated vehicle routing problem (CVRP) benchmarks, it achieves a 1.72% average gap to the best-known solutions and the best or tied-best gap on 12 of them, outperforming the genetic algorithm, tabu search, and general adaptive large neighborhood search (ALNS) comparators. Its effectiveness is further verified through extensive numerical experiments. First, considering couriers' learning effects reduces the total costs and concentrates route coverage. Second, the benefit of robust solutions is instance-dependent: They reduce overload costs under demand fluctuations in some cases, while excessively large uncertainty parameters lead to conservative, cost-increasing outcomes. Experimental results on real enterprise data clearly validate the superiority of the proposed model and algorithm over existing approaches.

Keywords: partitioned distribution; learning curve; robust optimization; adaptive large neighborhood search

Mathematics Subject Classification: 90C10, 90C27, 90C90

1. Introduction

Despite annual growth in the business volume and revenue of express service companies, China's logistics sector is burdened by high overall costs, where transportation takes up more than 50%.

Consequently, enhancing the efficiency and reducing the cost of last-mile delivery are of paramount importance to the sector. Distribution region partitioning is one of the most widely used last-mile delivery methods. By dividing the total delivery area into several subregions and assigning each to a specific courier, companies effectively reduce the scale of the delivery network and the complexity of the problem, thereby improving delivery efficiency.

In practice, express service companies usually use electronic maps, referring to administrative divisions and combining the experiences of relevant personnel regarding customer demand characteristics and road network information to delineate zones. However, these methods mainly depend on the personnel's experience and fail to account for real-time road network conditions. Moreover, most existing studies on distribution region partitioning do not fully consider the couriers' learning efficiency and customer demand uncertainty. These factors play an important role in actual delivery processes but are often ignored, leading to limitations in addressing the real scenarios.

This paper studies the distribution region partitioning problem considering both courier learning-related effects and customer demand uncertainty. As couriers repeatedly provide delivery services within assigned areas, their familiarity with the service region improves, leading to progressive reductions in travel time. The learning effect is modeled using learning curve theory, enabling a more accurate prediction of travel time and more effective route design within delivery zones. Meanwhile, this paper refines the sources of uncertainty to fluctuations in items' quality and volume, expanding the scenarios of customer demand uncertainty. It addresses the high complexity problem caused by the large number of customer points in actual express delivery.

The partitioned distribution problem (PDP) comprises the tactical stage and the operational stage. The tactical level determines the partitioning of customer points, and the operational level determines the specific routes for couriers within each delivery zone. The learning curve affects both stages of the PDP, influencing the partitioning of customer points and the specific route planning within delivery zones. Demand uncertainty only affects the operational level through its impact on route planning.

This paper proposes a robust partitioned distribution problem with learning curve and demand uncertainty, which jointly captures courier learning effects and customer demand uncertainty within a single two-stage framework. The problem sits at the intersection of two research streams that have, so far, developed separately. On one side, studies on learning-aware territory planning recognize the value of drivers' familiarity: Zhong et al. [1] balanced familiarity against day-to-day demand variation through core areas and flex zones, and Li et al. [2] adjusted territory plans dynamically according to predicted demand and driver experience. Both methods, however, operate under deterministic or forecast demand and offer no worst-case protection. On the other side, robust vehicle routing studies [3–5] provide such protection through uncertainty sets, but address single-level routing decisions without territory structures or learning effects. Integrating the two streams is not incidental. The partition is fixed before demand uncertainty is realized, and it is the carrier of learning: Couriers accumulate experience by repeatedly serving a stable, compact subregion. The robust routing stage must then absorb demand fluctuations within whatever territories the first stage has created. Consequently, the partition design determines the price of robustness—how much buffering cost is needed to protect against overload—and this hierarchical coupling between learning-driven partitioning and robust routing can only be captured in a joint model. The contributions are threefold.

(1) We introduce the robust PDP that combines learning-aware partitioning with robust routing under demand uncertainty, and we quantify how the price of robustness interacts with learning gains

across the two stages.

(2) We formulate a robust optimization model over a budgeted uncertainty set. Through piecewise linearization of the learning curve and dual reformulation of the worst-case conditions, the nonlinear formulation is transformed into a tractable two-stage model.

(3) We design a hybrid solution framework in which the first-stage is solved to optimality and an improved adaptive large neighborhood search (improved ALNS) metaheuristic solves the second-stage routing problem. The methodology is validated on standard benchmarks and a case study from a real express terminal network in Xi'an, China.

The remainder of this study proceeds as follows. Section 2 presents the related literature. Section 3 describes the mathematical models and presents the linearized models. Section 4 shows the details of the proposed solution approach. Section 5 presents the comparison results of the model and the algorithm. Section 6 concludes this study, offering some future research directions.

2. Literature review

This section reviews and summarizes the literature from four aspects: The delivery area partitioning problem, couriers' travel distance and time approximation, learning curves in vehicle routing problems (VRPs), and robust vehicle routing problems.

2.1. Delivery area partitioning problem

The partitioning criteria mainly include demand balance, work time balance, demand point density, administrative division, and other indicators. Fisher and Jaikumar [6] first proposed the idea of clustering first and then scheduling to solve large-scale routing problems. Considering the rapid growth of express delivery volume and urbanization in China, Huang et al. [7] proposed a city distribution area division and aggregation method based on population density. Clustering methods based on K-means [8], the genetic algorithm (GA), and intelligent hybrid algorithms [9] are commonly used methods for area partitioning. Matl et al. [10] summarized VRP problems focusing on route balance, and identified six common fairness measures. Haugland et al. [11] assigned fixed delivery areas to couriers according to postal codes, making it easier for couriers to familiarize themselves with the delivery area and randomly occurring customer groups. Khodabandeh et al. [12] studied the distribution scheduling problem of General Electric (GE), dividing customers with similar attributes into the same distribution unit based on the cities and postal codes where they are located.

Some scholars proposed adjustable area planning methods. Zhong et al. [1] studied the area planning problem with learning effects and solved it using a tabu search (TS) algorithm. In Zhong et al.'s research, the core area remains unchanged during the period, but the delivery area can be adjusted by assigning variable areas to different couriers. Li et al. [2] allowed periodic adjustment of the core area based on Zhong et al.'s research [1] and solved it using a two-stage rolling horizon (TSRH) algorithm. Schneider [13] examined the area planning problem under time window constraints and proposed a two-stage area planning method, assigning core areas and planning routes within the areas in each stage. Bender et al. [14] studied the area partitioning problem with heterogeneous couriers and vehicles, considering the conflict between always having the same courier serve each customer and daily demand fluctuations.

2.2. Courier travel distance and time approximation

Beardwood et al. [15] first proposed a distance approximation method, where for a tight convex region A containing n nodes, as $n \rightarrow \infty$, the travel length of the traveling salesman problem (TSP) gradually converges to $c\sqrt{nA}$, where c is a given constant. Stein [16] extended the parameters in Beardwood's model and estimated the optimal length for one or several vehicles. Hindle and Worthington [17] developed further distance approximations for the TSP, using the shape of the region or the distance between customers. Çavdar and Sokol [18] used customer coordinates and the standard deviation of distances between customers and the area's mean point for approximations. Larson and Odoni [19] proved the convergence of vehicle scheduling problems' estimates. Webb [20] first established a travel approximation method for the capacitated vehicle routing problem (CVRP) model, studying the correlation between path length and the distance between customers and the distribution center. Daganzo [21] expressed the approximate travel length of the CVRP as $CVRP(n) \approx 2rn/Q + 0.57\sqrt{nA}$, where n is the number of customers, r is the average distance between customers and the distribution depot, Q is the maximum number of customers that a vehicle is allowed to serve, and A is the area of the customer points. Figliozzi [22] studied the VRP with variable customer numbers, time window constraints, and demand levels. Zhong et al. [1] estimated drivers' travel time on the basis of the distances between customers, area seed points, and the delivery center. Since service times at customer units are relatively stable and predetermined by demand, the primary source of variable operational workload and uncertainty across different regions stems from travel times, making travel time approximations a robust proxy for partitioning decisions. In this study, approximate methods are used to assign delivery areas to drivers, applying the methods proposed by Zhong et al. [1] and Figliozzi [22] to estimate travel times, respectively.

2.3. Learning curves in the VRP

Sungur et al. [23] pointed out that ensuring the similarity of courier delivery routes in actual delivery processes helps improve service quality and customer satisfaction. Wright [24] first scientifically analyzed learning efficiency and quantified the learning curve based on the aircraft manufacturing industry. Jaber and Glock [25] comprehensively reviewed the learning curve models and applications, with extensive literature discussing mathematical models of learning curves by linking the number of services and delivery time, including logarithmic linear models [25] and exponential models [26].

Learning effects have been widely discussed in the contexts of manufacturing, machinery, project, and labor scheduling, but have received less attention in the VRP field. Zhong et al. [1] first combined travel time with learning effects, considering a multi-day VRP where the driver's familiarity is generated through repeated visits to service units. Kunkel and Schwind [27] considered a multi-day VRP with courier learning-related effects and used heuristic methods to solve it. Contrary to previous methods, Schneider [13] assumed that the courier's learning process had occurred in the past, treating the problem as a variant of a heterogeneous courier VRP. Hewitt et al. [28] proposed a method to linearize segmented learning curves. Chen et al. [29] studied the learning effects on service times for routing problems with experience-based service times.

This study follows the exponential learning model proposed by Chen et al. [29], traceable to De Jong's [26] learning curve. This model is chosen because it describes the practical phenomenon of limited compression in travel time.

2.4. The robust vehicle routing problem (RVRP)

The emergence of uncertainties may render the original routes highly inefficient. For example, when demands are uncertain, a vehicle may follow the original plan to visit a non-existent customer. Robust optimization can enhance the solution's immunity to the impacts of uncertainties. The goal of the RVRP is to optimize for the worst-case scenario, making the worst-case objective value minimal.

Most RVRP studies focus on uncertain demand and service time [5, 30]. Sungur et al. [3] proposed a robust optimization method for solving a CVRP with uncertain demand, assuming that all customers' demands simultaneously are at the worst case. Gounaris et al. [4] developed an adaptive metaheuristic algorithm to solve the over-conservatism in decision-making when all customers' demands simultaneously reach the worst case. Lee et al. [31] studied an RVRP with deadlines and uncertain demand and solved it using dynamic programming algorithms. Souyris et al. [32] studied a VRP with deadlines under uncertain service times, providing proof for the worst-case under independent uncertainty sets among routes. Cao et al. [33] proposed four robust strategies and an improved evolution algorithm to handle uncertain demand.

Braaten et al. [34] considered an RVRP with hard time windows, uncertain travel times, and demand budgets as polyhedral uncertainty sets, proposing an efficient heuristic method based on adaptive large neighborhood search (ALNS) for solving. Adulyasak and Jaillet [35] studied stochastic programming and robust optimization for a VRP with deadlines and uncertain travel times, showing that the latter has better immunity to uncertainties in most scenarios.

Regarding RVRP-solving algorithms, Han et al. [36] and Lu and Gzara [30] proposed a branch-and-cut algorithm for solving a VRP under uncertain travel times. Munari et al. [5] used a branch-and-price-and-cut algorithm considering set partitioning to solve RVRPs. Pureza et al. [37] and De La Vega et al. [38] combined tabu search (TS) and ant colony algorithms to solve RVRPs with time window constraints. Sim and Hart [39] proposed a two-stage hyper-heuristic method, using GA in the first stage to generate an initial solution and perturbing candidate solutions in the second stage. Braaten et al. [34] and Hu et al. [40] used ALNS and adaptive variable neighborhood search algorithms to solve the vehicle routing problem with uncertain travel times, respectively. Both algorithms obtained satisfactory optimization results efficiently.

In summary, the partitioning methods are mainly based on administrative area or postal code partitioning and K-means clustering partitioning. These partitioning methods only consider geographical factors and do not account for the number of customer points and their demand within the area, which may lead to uneven distribution of customer points within the area. In the application of learning curves in VRPs, most scholars only consider the impact of learning curves during the partitioning of customer points, while the learning effect on specific courier routes is less involved. In the research on RVRPs, most scholars focus on a single aspect, such as handling uncertain demand and/or uncertain travel/service time, or improving solving algorithms for RVRPs. Almost no scholars have considered learning effects and demand uncertainty simultaneously.

3. Mathematical models

3.1. Problem description

In the daily express delivery process in cities, couriers at the distribution center are responsible for different delivery areas. This ensures that all customers are served and avoids the overlap of courier services at the customers. Additionally, if fixed delivery areas are assigned to fixed couriers, it helps improve the couriers' familiarity with the delivery range and enhances customer satisfaction. However, as customer demand is dynamic, the distribution center should periodically calculate and adjust delivery areas to ensure the rationality of delivery areas and route planning within the areas. Periodically adjusting the delivery areas can help the distribution center monitor its operational status in real-time. This paper divides the research problem into two sub-problems: The partitioning of customer points and route planning for couriers within the areas. Figure 1 illustrates the problem's description.

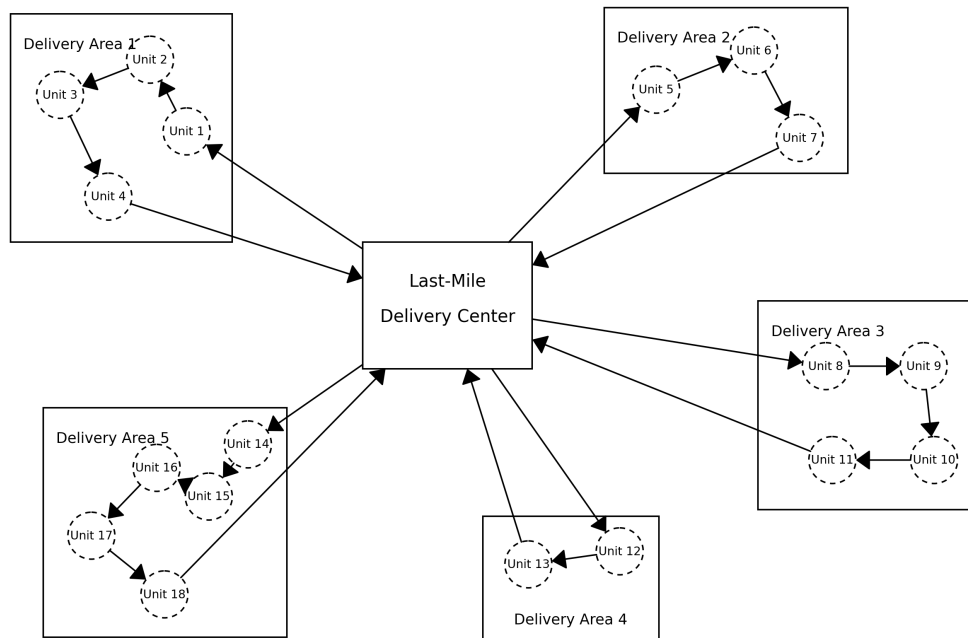


Figure 1. Schematic diagram of delivery zone partitioning and route planning.

This paper applies a two-stage approach, where the first stage aims to minimize the estimated workload of all couriers in the areas; the second stage aims to minimize the total travel time and service time costs for couriers within the areas. The mathematical description is as follows.

In a given complete graph $G = (I, A)$, the set I consists of two parts: The set of customer units and the virtual point o representing the distribution center. $A = \{(i, j) | i, j \in I; i \neq j\}$ represents the set of arcs, with the distance between nodes (i, j) , denoted as $d_{i,j}$ and the travel time is denoted as t_{ij} . The total planning horizon is the set of days T , indexed by $t \in T$, and on any given day t , the demand of customer i is w_{it} . Every h days, distribution areas are divided according to the workload of the couriers, forming the area set R . Each area $r \in R$ is exclusively assigned to one courier, who is equipped with a delivery vehicle (hereafter, couriers and vehicles are used interchangeably), forming the resource set K . Since the vehicles are subject to capacity constraints, they are allowed to return to the distribution center multiple

times, forming a set of trips M_k for the vehicle within each region. The rated load of each vehicle is denoted as μ . The assumptions of the problem are as follows:

- All vehicles start and end at the distribution center.
- All vehicles are homogeneous with rated load constraints.
- Total demand on each route cannot exceed the vehicle's capacity.
- There are no time window requirements for serving units.

With the proliferation of parcel stations and smart lockers, customers can pick up packages from parcel stations and lockers at any time, without following strict delivery schedules. This increases the flexibility and efficiency of the delivery process, as couriers are no longer constrained by time windows during deliveries.

3.2. Notation

The notation used throughout this paper is summarized in Table 1.

Table 1. Sets, parameters, and variables for the model.

Set	
T	Set of days in the total planning horizon, indexed by $t \in T$
H	Set of periods, indexed by $\tau \in H$
h	Number of days in each period
H_τ	Subset of days belonging to period τ , $H_\tau = \{h(\tau - 1) + 1, \dots, \min\{h\tau, T\}\}$
I'	Set of customer units, indexed by $i \in I'$
I	$I = I' \cup \{o\}$, where o represents the distribution center
I_k	Set of customer points served by vehicle $k \in K$
A	Set of arcs $A = \{(i, j) i, j \in I; i \neq j\}$, where an arc represents the distance between customer points i, j
A_k	Set of arcs served by vehicle k , $A_k = \{(i, j) i, j \in I_k; i \neq j\}$
K	Set of couriers (vehicles and couriers are used interchangeably)
M_k	Set of trips vehicle k passes through the distribution center, $m \in M_k$, $k \in K$
P_{mt}^k	Set of customers visited by courier k on the m -th trip on day t
Parameters	
$C_{i,k,\tau}$	Service time of vehicle k for customer i in period τ
$D_{i,k,\tau}$	Travel time of vehicle k in customer i in period τ
I_0	Minimum number of customers in a delivery zone
t_{ij}	Travel time of arc (i, j)
t_{ijmt}^k	Travel time of arc (i, j) on the m -th trip by courier k on day t adjusted by the learning factor
p_i	Service time of customer i
$p'_{i,k,t}$	Service time of courier k for customer i on day t adjusted by the learning factor
w_{it}	Actual uncertain demand of customer i on day t
$\bar{w}_{it}$	Nominal demand of customer i on day t
w'_{it}	Emergency demand of customer i on day t
γ_i	Maximum deviation of the uncertain demand of customer i
Γ_m	Uncertainty budget of the m -th trip, i.e., the upper bound on the total demand deviation

$w_{i\tau}$	Average demand for customer i on the period τ
$w'_{i\tau}$	Average emergency demand for customer i on the period τ
A_i	Area of customer i
$\rho_{i,k,t}$	Learning factor of courier k for customer i on day t
$\rho_{i,k,\tau}$	Learning factor of courier k for customer i in period τ
ρ_{ijmt}^k	Learning factor of courier k on the m -th trip traversing arc (i, j) on day t
α_k	Learning potential of courier k
β_k	Decay factor of courier k
$\omega_{11}, \omega_{12}, \omega_{13}$	Estimation coefficient for service time
$\bar{d}$	The average distance between customers and the distribution center
ot^{cost}	Unit overload cost
tt^{cost}	Unit travel time cost
ω_2	Estimation coefficient for travel time
b	Maximum working hours for a vehicle
μ	Maximum load capacity for a vehicle
θ	Estimated parameter for the minimum working hours of the vehicle
d_{i0}, d_{ik}, d_{0k}	The distances between customers and the distribution center, between units and regional centers, and between the distribution center and regional centers
d_{ij}	Distance between nodes i and j
$\bar{q}$	Maximum number of visits to any given unit
$\bar{q}_j$	Breakpoint of the piecewise linearization, $\bar{q}_j = 1 + \frac{j}{J}(\bar{q} - 1)$
$\bar{\rho}_j$	Learning factor at breakpoint $\bar{q}_j$, $\bar{\rho}_j = (1 - \alpha_k) + \alpha_k \bar{q}_j^{-\beta_k}$
J	Number of segments in the piecewise linearization
M	A sufficiently large constant used in the Big-M reformulation

Decision variables

x_{ijmt}^k	If vehicle $k \in K$ traverses arc $(i, j) \in A$ on the m -th ($m \in M_k$) trip on day $t \in T$, the value is 1; 0 otherwise
y_{imt}^k	If vehicle $k \in K$ departs from the distribution center and passes through customer $i \in I_k$ on the m -th ($m \in M_k$) trip on day $t \in T$, the value is 1; 0 otherwise
$\bar{x}_{i,k,\tau}$	Binary variable, which takes 1 if vehicle $k \in K$ serves customer $i \in I'$ in period $\tau \in H$; 0 otherwise
$s_{i,k,t}$	Time vehicle $k \in K$ arrives at customer $i \in I_k$ on day $t \in T$
ot_{mt}^k	Overload amount of courier $k \in K$ on day $t \in T$ on m -th trip ($m \in M_k$)
z_{it}	Uncertain demand deviation of customer i on day t
f_{mt}^k	Load of vehicle $k \in K$ at departure from the distribution center on the m -th trip on day $t \in T$, $m \in M_k$
f_{jmt}^k	Cumulative load of vehicle $k \in K$ after serving customer j on the m -th trip on day $t \in T$
f_{imt}^k	Cumulative load of vehicle $k \in K$ after serving customer i on the m -th trip on day $t \in T$
$q_{i,k,t}$	Number of times vehicle $k \in K$ visits unit i on day $t \in T$
$q_{i,k,\tau}$	Number of times vehicle $k \in K$ visits unit i during period $\tau \in H$
q_{ijmt}^k	Number of times courier $k \in K$ traverses arc $(i, j) \in A$ on the m -th trip on day $t \in T$
$y_{ik\tau}^n$	Continuous weight of the n -th breakpoint in the special ordered set of Type 2 (SOS2) linearization of customer i served by courier k in period τ , $n \in \{1, 2, 3, 4\}$

$z_{ik\tau}^m$	Binary variable, which takes 1 if segment m is active in the SOS2 linearization of customer i served by courier k in period τ , $m \in \{1, 2, 3\}$
u_{kt}	Binary auxiliary variable for linearizing uncertain-demand constraint of courier k on day t
v_{ikt}	Auxiliary variable equal to $x_{imt}^k z_{it}$, used in the Big-M reformulation

3.3. Two-stage partitioned distribution model

In general VRP research, the service time for each courier at the same unit and the travel time for each courier traversing arc (i, j) are fixed and do not consider the impact of experience gained from multiple visits to the same unit. In reality, the enhancement of experience reduces the service time and travel time, allows couriers to optimize their delivery routes, and increases service speed. The greater the learning ability of the courier, the greater the impact. Learning experience can be characterized using learning curves.

Let T denote the set of days in the total planning horizon, indexed by $t \in T$. This horizon is partitioned into a finite set of discrete tactical periods, represented by the set H and indexed by $\tau \in H$. Each period τ comprises a fixed duration of h days, and H_τ denotes the subset of days belonging to period τ . Daily quantities are indexed by t , whereas period-level quantities (indexed by τ) are aggregates over the days in H_τ . The relationships between the time nodes are shown in Figure 2.

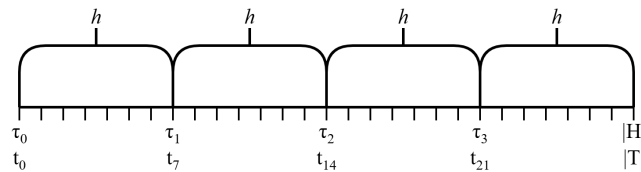


Figure 2. Time node diagram.

3.3.1. Learning curve

The service time of couriers at units and the travel time between arcs (i, j) depend on the experience couriers gain from previous services to those customers and the couriers' learning capabilities. Generally, when couriers frequently visit a unit, they spend less time at that unit. The number of visits by courier k to customer i on day t is denoted as $q_{i,k,t}$. We assume that all couriers have a certain level of basic training experience at the start of their work, which creates a threshold for basic training experience for all couriers, represented as $q_{i,k,0} = 1$.

This study adopts the learning model proposed by Chen et al. [29] to describe the changes in service time and travel time. A learning factor $\rho_{i,k,t} \in [0, 1]$ is introduced to adjust the travel time of courier k through customer i on day t . Let $\alpha_k \in [0, 1]$ denote the learning potential of courier k , which represents the compressible space, and let $\beta_k \in [0, 1]$ denote the decay factor, indicating the speed of compression as experience accumulates. The learning curve is given by Formula (1):

$$\rho_{i,k,t} = (1 - \alpha_k) + \alpha_k q_{i,k,t}^{-\beta_k} \quad (1)$$

The learning process describes the reduction in travel time and service time as couriers gain experience. The learning factor $\rho_{i,k,t}$ is used to adjust the service time within units and the travel time between units, reflecting the couriers' familiarity with the delivery area. The learning curve affects both stages of

the partitioned distribution study. During the specific route planning stage, the learning curve reduces the service time within units and the travel time between units. Consequently, the learning curve also influences the estimation of workload within the region. During the region partitioning stage, the frequency of assigning unit i to courier k in each period τ is represented as $q_{i,k,\tau}$, which increments by one per assigned period; the day-level visit count is therefore aggregated to the period level, so each assigned period still counts as one visit. It is assumed that all regions have a threshold value at the beginning of the allocation, denoted as $q_{i,k,0} = 1$, and the learning curve formula is given as follows:

$$\rho_{i,k,\tau} = (1 - \alpha_k) + \alpha_k q_{i,k,\tau}^{-\beta_k}, \quad i \in I', k \in K, \tau \in H \quad (2)$$

3.3.2. Estimating workload in delivery zones

To improve the efficiency of computing delivery zone partitioning, this study adopts the workload estimation method proposed by Figliozzi [22], which approximates the service time using the area and demand of units. Based on the courier travel time estimation method proposed by Zhong et al. [1], average distances between units are used to approximate travel time. Figliozzi [22] and Zhong et al. [1] have validated the feasibility of this method in practical applications. Using this method significantly reduces computational complexity without sacrificing accuracy, thus improving the efficiency of delivery zone partitioning computations.

It is worth noting that during the strategic regional partitioning stage, the total planning horizon is discretized into the set H of tactical periods. Consequently, customer demands are evaluated on the basis of the average customer demand calculated over the days within H_τ . Therefore, the parameters $w_{i\tau}$ and $w'_{i\tau}$ denote the average demand and emergency demand for customer i in the period τ , which is calculated as the mathematical average of the daily demands recorded within that specific period. Formulas (3) and (4) are the calculation formulas for service time within units and travel time between units, respectively.

$$C_{i,k,\tau}(\rho_{i,k,\tau}) = \rho_{i,k,\tau} \left(\omega_{11} \sqrt{A_i w_{i\tau}} + \omega_{12} \sqrt{A_i w'_{i\tau}} + \omega_{13} \frac{2\bar{d} w_{i\tau}}{\mu} \right), \quad i \in I', k \in K, \tau \in H \quad (3)$$

$$D_{i,k,\tau}(\rho_{i,k,\tau}) = \rho_{i,k,\tau} \omega_2 (d_{ik} + d_{i0} - d_{0k}), \quad i \in I', k \in K, \tau \in H \quad (4)$$

We use the area and demand of units to approximate the service time. The service time for courier k in unit i on day t with consideration of learning factor is given by Formula (3). Parameters ω_{11} , ω_{12} , and ω_{13} are the estimation coefficients for service time in routing problems. Additionally, A_i is the spatial area of customer i , $\bar{d}$ represents the average distance between customers and the distribution center, and μ is the rated load capacity of the vehicle. We use the average distance between units to estimate the travel time between units. The travel time between units i for the courier in delivery zone k on day t with consideration of the learning factor is approximated as Formula (4). Parameter ω_2 is the estimation parameter for travel time between customers, d_{ik} is the distance from unit i to the center of delivery zone k , d_{i0} is the distance from unit i to the distribution center, and d_{0k} is the distance from the center of delivery zone k to the distribution center. Figure 3 is a schematic diagram for calculating travel time between units.

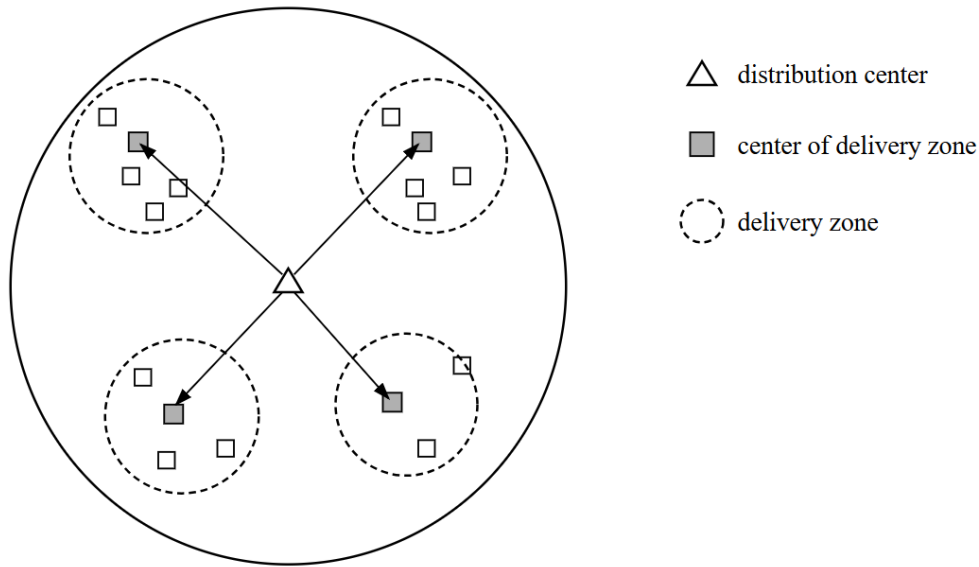


Figure 3. Estimation of travel time between units.

3.3.3. Demand uncertainty

We assumed that the uncertain customer demands between different routes are independent. The set of customers visited by courier k during the m -th trip from the distribution center on day t is denoted as P_{mt}^k , and the uncertainty set is defined as follows:

$$U = \left\{ w_{it} \mid \bar{w}_{it} - z_{it} \leq w_{it} \leq \bar{w}_{it} + z_{it}, 0 \leq z_{it} \leq \gamma_i, i \in I', t \in T, \sum_{i \in P_{mt}^k} z_{it} \leq \Gamma_m, m \in M_k \right\} \quad (5)$$

where w_{it} is the actual uncertain demand of customer i on day t , $\bar{w}_{it}$ is nominal demand, and γ_i is the maximum demand deviation of customer i . The demand of each customer fluctuates within the range of $[\bar{w}_{it} - z_{it}, \bar{w}_{it} + z_{it}]$. The demand uncertainty budget for each route handled by each courier per day is Γ_m , which denotes the upper bound on the total demand deviation for all customers on the route. Γ_m is used to control the level of demand uncertainty on each route. When $\Gamma_m = 0$, demand uncertainty is not considered.

Souyris et al. [32] proved that the worst case of the route load on day t occurs when the sum of customer demand deviations in P_{mt}^k reaches its maximum value. This condition can be described as follows:

$$\sum_{i \in P_{mt}^k} z_{it} = \min \left\{ \Gamma_m, \sum_{i \in P_{mt}^k} \gamma_i \right\}, \quad \forall k \in K, m \in M_k, t \in T \quad (6)$$

To study the impact of demand uncertainty on courier routes, this study does not consider the service time at customer points. If the load of the courier on any route exceeds its maximum load capacity, a penalty cost due to overloading will be incurred. Let ot_{mt}^k be a non-negative auxiliary variable representing courier overloading, which is calculated as follows:

$$ot_{mt}^k = \max\{0, f_{mt}^k - \mu\}, \quad \forall m \in M_k, k \in K, t \in T \quad (7)$$

If the load of courier on the m -th trip from the distribution center on day t is less than its maximum load μ , then $ot_{mt}^k = 0$; if it exceeds μ , then $ot_{mt}^k = f_{mt}^k - \mu$, and a corresponding penalty cost $ot^{cost} ot_{mt}^k$ will be incurred.

3.3.4. Model establishment

The partitioned distribution model discussed in this paper comprises two stages. The first stage involves estimating the workload in delivery zones using unit demand and travel distance between units, thereby partitioning the units. The objective of this stage is to determine the most suitable number and range of delivery zones from all delivery units. The second stage assigns the optimal delivery routes to couriers within each delivery zone. The synergy of these two stages constitutes a complete solution to the PDP, effectively partitioning delivery zones and planning the optimal delivery routes for couriers, thereby reducing the total delivery cost. The model for the first stage can be represented as follows:

$$\min \sum_{i \in I'} \sum_{k \in K} \sum_{\tau \in H} (C_{i,k,\tau} + D_{i,k,\tau}) \bar{x}_{i,k,\tau} \quad (8)$$

subject to (s.t.)

$$\sum_{i \in I'} (C_{i,k,\tau} + D_{i,k,\tau}) \bar{x}_{i,k,\tau} \leq b, \quad k \in K, \tau \in H \quad (9)$$

$$\sum_{i \in I'} (C_{i,k,\tau} + D_{i,k,\tau}) \bar{x}_{i,k,\tau} \geq \theta b, \quad k \in K, \tau \in H \quad (10)$$

$$\sum_{k \in K} \bar{x}_{i,k,\tau} = 1, \quad i \in I', \tau \in H \quad (11)$$

$$\sum_{i \in I'} \bar{x}_{i,k,\tau} \geq I_0, \quad k \in K, \tau \in H \quad (12)$$

$$C_{i,k,\tau}(\rho_{i,k,\tau}) = \rho_{i,k,\tau} \left(\omega_{11} \sqrt{A_i w_{i\tau}} + \omega_{12} \sqrt{A_i w'_{i\tau}} + \omega_{13} \frac{2\bar{d}w_{i\tau}}{\mu} \right), \quad i \in I', k \in K, \tau \in H \quad (13)$$

$$D_{i,k,\tau}(\rho_{i,k,\tau}) = \rho_{i,k,\tau} \omega_2 (d_{ik} + d_{i0} - d_{0k}), \quad i \in I', k \in K, \tau \in H \quad (14)$$

$$\rho_{i,k,\tau} = (1 - \alpha_k) + \alpha_k q_{i,k,\tau}^{-\beta_k}, \quad i \in I', k \in K, \tau \in H \quad (15)$$

$$q_{i,k,\tau+1} = q_{i,k,\tau} + \bar{x}_{i,k,\tau}, \quad i \in I', k \in K, \tau \in H \quad (16)$$

$$\bar{x}_{i,k,\tau} \in \{0, 1\}, \quad i \in I', k \in K, \tau \in H \quad (17)$$

The objective function (8) minimizes the service time within units and the travel time between units; Constraint (9) is the maximum working time constraint within the region; Constraint (10) is the minimum working time constraint within the region; Constraint (11) ensures that each unit is assigned to only one region; Constraint (12) ensures that the number of units in each region is not less than a specific value; Constraints (13) and (14) are the estimated values for service time within units and travel time

between units, respectively; Constraint (15) represents the learning process; Constraint (16) represents the calculation process for the number of visits; and Constraint (17) is the 0-1 variable constraint.

The first stage aims to determine the boundaries of each delivery zone, while the second stage needs to plan the optimal delivery route within each delivery zone. The objective of this stage is to minimize the travel time cost and overloading cost of all couriers while ensuring reasonable coverage of all service points. The model for the second stage can be represented as follows:

$$\min \quad tt^{cost} \sum_{k \in K} \sum_{m \in M_k} \sum_{(i,j) \in A} \sum_{t \in T} t_{ij} x_{ijmt}^k + ot^{cost} \sum_{k \in K} \sum_{m \in M_k} \sum_{t \in T} ot_{mt}^k \quad (18)$$

s.t.

$$\sum_{j \in I_k} x_{ijmt}^k = y_{imt}^k, \quad \forall i \in I_k, k \in K, m \in M_k, t \in T \quad (19)$$

$$\sum_{m \in M_k} y_{imt}^k = \bar{x}_{i,k,\tau}, \quad \forall i \in I_k, k \in K, t \in H_\tau, \tau \in H \quad (20)$$

$$\sum_{j \in I_k} x_{0jmt}^k = \sum_{j \in I_k} x_{j0mt}^k, \quad \forall k \in K, m \in M_k, t \in T \quad (21)$$

$$\sum_{j \in I_k} x_{ijmt}^k = \sum_{j \in I_k} x_{jimt}^k, \quad \forall i \in I_k, k \in K, m \in M_k, t \in T \quad (22)$$

$$f_{mt}^k - ot_{mt}^k \leq \mu, \quad \forall m \in M_k, k \in K, t \in T \quad (23)$$

$$s_{i,k,t} + t'_{ijmt}^k + p'_{i,k,t} - s_{j,k,t} \leq (1 - x_{ijmt}^k) M, \quad \forall i, j \in I_k, k \in K, m \in M_k, t \in T \quad (24)$$

$$t_{0j} - s_{j,k,t} \leq (1 - x_{0jmt}^k) M, \quad \forall j \in I_k, k \in K, m \in M_k, t \in T \quad (25)$$

$$p'_{i,k,t} = \rho_{i,k,t} p_i, \quad \forall i \in I_k, k \in K, t \in T \quad (26)$$

$$t'_{ijmt}^k = \rho_{ijmt}^k t_{ij}, \quad \forall i, j \in I_k, k \in K, m \in M_k, t \in T \quad (27)$$

$$\rho_{ijmt}^k = (1 - \alpha_k) + \alpha_k (q_{ijmt}^k)^{-\beta_k}, \quad \forall i, j \in I_k, k \in K, m \in M_k, t \in T \quad (28)$$

$$q_{ijmt+1}^k = q_{ijmt}^k + x_{ijmt}^k, \quad \forall i, j \in I_k, k \in K, m \in M_k, t \in T \quad (29)$$

$$q_{i,k,t+1} = q_{i,k,t} + \bar{x}_{i,k,\tau}, \quad \forall i \in I_k, k \in K, t \in H_\tau, \tau \in H \quad (30)$$

$$f_{mt}^k = \sum_{(i,j) \in A_k} (\bar{w}_{it} + z_{it}) x_{ijmt}^k, \quad m \in M_k, k \in K, t \in T \quad (31)$$

$$\sum_{i \in I_k} \sum_{j \in I_k} x_{ijmt}^k z_{it} \geq \min \left\{ \Gamma_m, \sum_{i \in I_k} \sum_{j \in I_k} x_{ijmt}^k \gamma_i \right\}, \quad \forall k \in K, m \in M_k, t \in T \quad (32)$$

$$f_{jmt}^k - \bar{w}_{jt} - z_{jt} - f_{imt}^k \leq (1 - x_{ijmt}^k) M, \quad \forall i, j \in I_k, m \in M_k, k \in K, t \in T \quad (33)$$

$$y_{imt}^k \in \{0, 1\}, \quad \forall i \in I_k, m \in M_k, k \in K, t \in T \quad (34)$$

$$x_{ijmt}^k \in \{0, 1\}, \quad \forall (i, j) \in A, m \in M_k, k \in K, t \in T \quad (35)$$

The objective function (18) minimized the travel time cost and overload cost of all vehicles; Constraints (19) and (20) indicate that couriers can only serve the units within their allocated region from the first stage; Constraints (21) and (22) ensure the flow balance constraints for visiting customer points; Constraint (23) represents the vehicle capacity constraint; Constraint (24) ensures that the service for each customer point starts only after the previous customer's service ends; Constraint (25) represents the starting service time constraint for the first customer on each route; Constraint (26) represents the service time for customer points considering the learning curve; Constraint (27) represents the travel time considering the learning curve; Constraints (28) and (29) represent the learning process, where the number of visits to the arc increases by one when the courier passes through the arc; Constraint (30) updates the day-level customer visit count $q_{i,k,t}$ by one per day whenever unit i is assigned to courier k in the corresponding period; Constraint (31) initializes the load of the vehicle at the distribution center to the total demand of all customers on the route; Constraint (32) ensures the maximization of uncertain demand deviation on each route; Constraint (33) propagates the cumulative load from node to node, so that the cumulative load of the vehicle passing through the later points is greater than that at the earlier points, and the load at the first customer of a trip equals that customer's demand, and the cumulative load is increased by the demand of each customer just served along the trip; and Constraints (34) and (35) represent the 0-1 variable constraints.

3.4. Model linearization

3.4.1. Linearization of the learning curve

Given the nonlinear characteristics of the learning curve equation, this study converts the nonlinear Constraints (15) into linear constraints through piecewise linear approximation. Figure 4 shows the learning curve divided into three segments.

Let $\bar{q}$ represent the maximum number of visits to any given unit, and q represent the number of visits, and $\bar{q}_j = 1 + \frac{j}{J}(\bar{q} - 1)$, $\forall j \in \{1, 2, \dots, J\}$. For the learning curve function. We can replace the exponential learning function constraints with the following J additional linear constraints:

$$\rho = \frac{\bar{\rho}_1 - 1}{\bar{q}_1 - 1} (q - 1) + 1, \quad \forall q \in [1, \bar{q}_1] \quad (36)$$

$$\rho = \frac{\bar{\rho}_2 - \bar{\rho}_1}{\bar{q}_2 - \bar{q}_1} (q - \bar{q}_1) + \bar{\rho}_1, \quad \forall q \in [\bar{q}_1, \bar{q}_2] \quad (37)$$

$$\rho = \frac{\bar{\rho}_3 - \bar{\rho}_2}{\bar{q}_3 - \bar{q}_2} (q - \bar{q}_2) + \bar{\rho}_2, \quad \forall q \in [\bar{q}_2, \bar{q}_3] \quad (38)$$

...

$$\rho = \frac{\bar{\rho}_J - \bar{\rho}_{J-1}}{\bar{q}_J - \bar{q}_{J-1}} (q - \bar{q}_{J-1}) + \bar{\rho}_{J-1}, \quad \forall q \in [\bar{q}_{J-1}, \bar{q}_J] \quad (39)$$

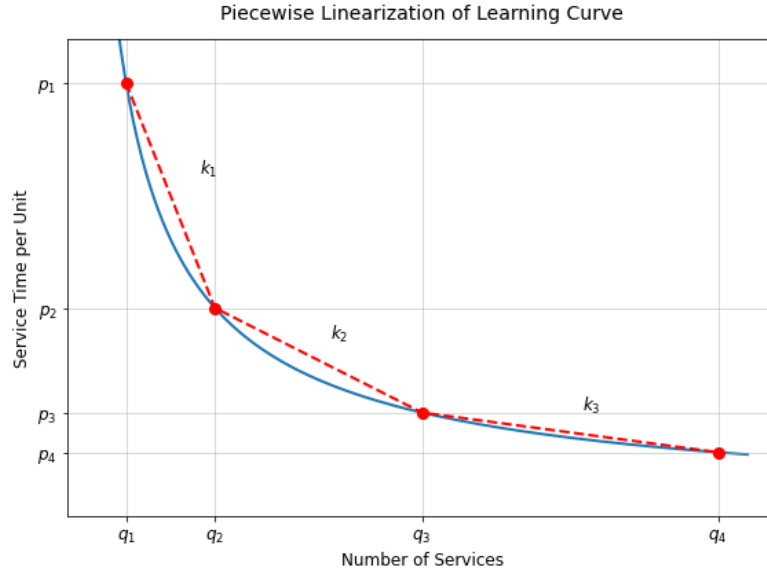


Figure 4. Piecewise linearization of learning curve.

The number of segments in this type of learning-curve linearization is commonly chosen based on practical solvability; for instance, Qin et al. [41] linearize a nonlinear worker's learning curve using a similar approach. Since learning curves are typically concave, with most curvature concentrated in the early stage and a nearly flat tail near steady-state performance, a small number of segments is generally sufficient to capture their essential shape. Three segments are therefore a natural candidate, corresponding to the three characteristic stages of the learning curve — rapid improvement, transition, and near-steady-state — while keeping the number of additional binary variables manageable.

With $J = 3$, the breakpoints are placed uniformly at $\bar{q}_j = 1 + \frac{j}{3}(\bar{q} - 1)$, i.e., $\bar{q}_1 = 1 + (\bar{q} - 1)/3$, $\bar{q}_2 = 1 + 2(\bar{q} - 1)/3$, and $\bar{q}_3 = \bar{q}$, with the corresponding learning factors $\bar{\rho}_j = (1 - \alpha_k) + \alpha_k \bar{q}_j^{-\beta_k}$. For the first (partitioning) stage, $\bar{q} = 4$, so the breakpoints are $\{1, 2, 3, 4\}$; since the period-level visit count $q_{i,k,\tau}$ in (16) only takes the integer values $\{1, 2, 3, 4\}$, these breakpoints coincide with every feasible visit count and the approximation is exact at all feasible points. For the second (routing) stage, $\bar{q} = 28$, so the breakpoints are $\{1, 10, 19, 28\}$; under the baseline $\alpha_k = 0.1$ and $\beta_k = 0.2$ used in Section 5, the maximum absolute approximation error is 0.012 (at most 1.2% relative), concentrated in the first segment where the curvature of the learning curve is largest, and it is 0.096, that is, under the stronger learning setting $\alpha_k = 0.3, \beta_k = 0.6$ used in the learning-and-demand-uncertainty experiment. Table 2 reports a sensitivity analysis over $J = 2, 3, 4, 5$: increasing J to 4 or 5 changes the first-stage objective by less than 0.15% while leaving the partition unchanged, and reduces the second-stage error only marginally, at the cost of 800 additional binary variables per segment; the error can also be reduced by using non-uniform breakpoints denser near the origin. We therefore set $J = 3$ as the best accuracy–cost trade-off.

The linearized expression of constraints (15) in the model is as follows, where $\bar{q}_j, \forall j \in J$ is the partition point of the piecewise function.

Table 2. Sensitivity of the number of segments J (baseline $\alpha_k = 0.1, \beta_k = 0.2$).

J	Stage-1 max. error	Stage-2 max. error	First-stage objective	Time (s)
2	0.0018	0.0153	11891.44	0.15
3	0	0.0120	11882.84	0.33
4	0.0003	0.0098	11884.50	0.28
5	0.0002	0.0083	11884.01	0.30

$$q_{i,k,\tau} = y_{ik\tau}^1 \cdot 1 + y_{ik\tau}^2 \bar{q}_1 + y_{ik\tau}^3 \bar{q}_2 + y_{ik\tau}^4 \bar{q}_3, \quad \forall i \in I', k \in K, \tau \in H \quad (40)$$

$$\rho_{i,k,\tau} = y_{ik\tau}^1 \cdot 1 + y_{ik\tau}^2 \bar{\rho}_1 + y_{ik\tau}^3 \bar{\rho}_2 + y_{ik\tau}^4 \bar{\rho}_3, \quad \forall i \in I', k \in K, \tau \in H \quad (41)$$

$$y_{ik\tau}^1 + y_{ik\tau}^2 + y_{ik\tau}^3 + y_{ik\tau}^4 = 1, \quad \forall i \in I', k \in K, \tau \in H \quad (42)$$

$$y_{ik\tau}^1 \leq z_{ik\tau}^1, \quad \forall i \in I', k \in K, \tau \in H \quad (43)$$

$$y_{ik\tau}^2 \leq z_{ik\tau}^1 + z_{ik\tau}^2, \quad \forall i \in I', k \in K, \tau \in H \quad (44)$$

$$y_{ik\tau}^3 \leq z_{ik\tau}^2 + z_{ik\tau}^3, \quad \forall i \in I', k \in K, \tau \in H \quad (45)$$

$$y_{ik\tau}^4 \leq z_{ik\tau}^3, \quad \forall i \in I', k \in K, \tau \in H \quad (46)$$

$$z_{ik\tau}^1 + z_{ik\tau}^2 + z_{ik\tau}^3 = 1, \quad \forall i \in I', k \in K, \tau \in H \quad (47)$$

$$z_{ik\tau}^m \in \{0, 1\}, \quad \forall i \in I', k \in K, \tau \in H, m \in \{1, 2, 3\} \quad (48)$$

$$y_{ik\tau}^n \geq 0, \quad \forall i \in I', k \in K, \tau \in H, n \in \{1, 2, 3, 4\} \quad (49)$$

Constraints (42)–(49) form a standard SOS2 (special ordered set of type 2) formulation: the equality in (47) activates exactly one segment of the learning curve, and the upper bounds (43)–(46) force the positive weights $y_{ik\tau}^1, \dots, y_{ik\tau}^4$ to lie at the two adjacent breakpoints of the active segment only. Interpolation is therefore confined to a single linear segment, so the piecewise-linear approximation is well defined over $q \in [1, \bar{q}]$. Because the breakpoint values satisfy $\bar{\rho}_j = \rho(\bar{q}_j)$, the surrogate coincides with the true learning curve at every breakpoint and is continuous across segment boundaries. Moreover, Constraints (15) and (28) both evaluate the same univariate learning function $\rho(q) = (1 - \alpha) + \alpha q^{-\beta}$, differing only in the visit-count variable ($q_{i,k,\tau}$ and $q_{ijm\tau}^k$, respectively). Equations (40)–(49) are stated for the first (partitioning) stage with period index τ ; the second (routing) stage uses the identical template with τ replaced by the day index t and $q_{i,k,\tau}$ replaced by $q_{ijm\tau}^k$. Accordingly, the same SOS2 template is used to linearize both of them. It is worth noting that the piecewise-linear surrogate and its approximation error bound therefore carry over directly from the partitioning stage to the routing stage, so the accuracy of the linearization is guaranteed uniformly across both stages.

3.4.2. Linearization of uncertain constraints

The total uncertain demand deviation on a single route can be described as follows:

$$\sum_{i \in I_k} x_{ijmt}^k z_{it} \geq \min\{\Gamma_m, \sum_{i \in I_k} x_{ijmt}^k \gamma_i\}, \quad \forall k \in K, m \in M_k, j \in I_k, t \in T \quad (50)$$

Let us introduce a binary variable u_{kt} to indicate the relationship between Γ_m and $\sum_{i \in I_k} x_{ijmt}^k \gamma_i$. We can classify u_{kt} as follows:

$$u_{kt} = 0 \Rightarrow \begin{cases} \Gamma_m \leq \sum_{i \in I_k} x_{ijmt}^k \gamma_i \Rightarrow \Gamma_m - \sum_{i \in I_k} x_{ijmt}^k \gamma_i \leq 0 \Rightarrow \Gamma_m - \sum_{i \in I_k} x_{ijmt}^k \gamma_i - Mu_{kt} \leq 0 \\ \sum_{i \in I_k} x_{ijmt}^k z_{it} \geq \min\{\Gamma_m, \sum_{i \in I_k} x_{ijmt}^k \gamma_i\} \Rightarrow \Gamma_m - \sum_{i \in I_k} x_{ijmt}^k z_{it} - Mu_{kt} \leq 0 \end{cases} \quad (51)$$

$$u_{kt} = 1 \Rightarrow \begin{cases} \Gamma_m \geq \sum_{i \in I_k} x_{ijmt}^k \gamma_i \Rightarrow \sum_{i \in I_k} x_{ijmt}^k \gamma_i - \Gamma_m \leq 0 \Rightarrow \sum_{i \in I_k} x_{ijmt}^k \gamma_i - \Gamma_m \leq M(1 - u_{kt}) \\ \sum_{i \in I_k} x_{ijmt}^k z_{it} \geq \min\{\Gamma_m, \sum_{i \in I_k} x_{ijmt}^k \gamma_i\} \Rightarrow \sum_{i \in I_k} x_{ijmt}^k z_{it} - \sum_{i \in I_k} x_{ijmt}^k \gamma_i \leq M(1 - u_{kt}) \end{cases} \quad (52)$$

Let us introduce an auxiliary variable $v_{ikt} = x_{ijmt}^k z_{it}$, and we

$$\begin{cases} x_{ijmt}^k = 0 \Rightarrow v_{ikt} = 0 \Rightarrow \begin{cases} v_{ikt} \geq 0 \\ v_{ikt} \leq 0 \end{cases} \Rightarrow \begin{cases} x_{ijmt}^k = 0, -v_{ikt} \leq 0 \\ x_{ijmt}^k = 0, v_{ikt} \leq 0 \end{cases} \\ x_{ijmt}^k = 1 \Rightarrow v_{ikt} = z_{it} \Rightarrow \begin{cases} v_{ikt} \geq z_{it} \\ v_{ikt} \leq z_{it} \end{cases} \Rightarrow \begin{cases} 1 - x_{ijmt}^k = 0, -v_{ikt} \leq -z_{it} \Rightarrow z_{it} - v_{ikt} \leq 0 \\ 1 - x_{ijmt}^k = 0, v_{ikt} \leq z_{it} \Rightarrow v_{ikt} - z_{it} \leq 0 \end{cases} \end{cases} \quad (53)$$

According to Constraints (53), we

$$\begin{cases} -v_{ikt} - Mx_{ijmt}^k \leq 0 \\ v_{ikt} - Mx_{ijmt}^k \leq 0 \\ z_{it} - v_{ikt} - M(1 - x_{ijmt}^k) \leq 0, \quad \forall i \in I_k, k \in K, t \in T \\ v_{ikt} - z_{it} - M(1 - x_{ijmt}^k) \leq 0 \\ v_{ikt} \geq 0 \end{cases} \quad (54)$$

Finally, we

$$\begin{cases} \Gamma_m - \sum_{i \in I_k} x_{ijmt}^k \gamma_i - Mu_{kt} \leq 0 \\ \Gamma_m - \sum_{i \in I_k} v_{ikt} - Mu_{kt} \leq 0 \\ \sum_{i \in I_k} x_{ijmt}^k \gamma_i - \Gamma_m - M(1 - u_{kt}) \leq 0 \\ \sum_{i \in I_k} x_{ijmt}^k \gamma_i - \sum_{i \in I_k} v_{ikt} - M(1 - u_{kt}) \leq 0 \end{cases}, \quad \forall k \in K, t \in T \quad (55)$$

3.4.3. Linearized two-stage robust model

By executing the piecewise linearization and big-M reformulations detailed above, the original mathematically intractable non-linear models are successfully transformed into a structured, tractable two-stage optimization framework. This transformation effectively purges all nonlinearity constraints by replacing the original problematic expressions with sets of standard linear inequalities.

In the first partitioning stage, the highly non-linear courier learning curve encapsulated in the original Constraint (15) is completely linearized. It is replaced by a set of piecewise linear segments and auxiliary variables (Equations (40)–(49)). Consequently, the reformulated first-stage model settles into a standard mixed-integer quadratic programming formulation (MIQP). In this framework, all constraints are strictly linear, while the objective function preserves quadratic terms arising from the bilinear products of assignment decisions. The MIQP model of the first stage can be expressed as follows:

$$\min \sum_{i \in I'} \sum_{k \in K} \sum_{\tau \in H} (C_{i,k,\tau} + D_{i,k,\tau}) \bar{x}_{i,k,\tau} \quad (56)$$

s.t. Constraints (9)–(14), (16)–(17), and (40)–(49).

In the second routing stage, the mathematical complexity escalates as the system faces the dual impacts of couriers' learning effects and demand uncertainty. The original non-linear behaviors embedded in Constraints (28) and (32) are thoroughly linearized. Specifically, the non-linear Constraint (28) is replaced by a set of piecewise linear segments and auxiliary variables (Equations (40)–(49)). The robust uncertainty constraint, Constraint (32), is reformulated into standard linear inequalities via big-M dual transformations and auxiliary workload tracking variables (Equations (54) and (55)). The mixed-integer linear programming model of the second stage can be expressed as follows:

$$\min \quad tt^{cost} \sum_{k \in K} \sum_{m \in M_k} \sum_{(i,j) \in A_k} t_{ij} x_{ijmt}^k + ot^{cost} \sum_{k \in K} \sum_{m \in M_k} ot_{mt}^k \quad (57)$$

s.t. Constraints (19)–(27), (29)–(31), (33)–(35), (40)–(49), and (54)–(55).

It is worth noting in the second stage, Constraints (40)–(49) and (54)–(55) are written with the day index t and the arc-level visit count q_{ijmt}^k instead of the period index τ and the customer-level visit count $q_{i,k,\tau}$.

4. Algorithm design

4.1. Algorithm framework

The solution framework designed in this paper is divided into two main stages: The customer point partitioning stage and the route planning stage within the delivery area. In the customer point partitioning stage, the mathematical model is converted into a standard MIQP (shown in Section 3.4.3), which can be accepted and efficiently solved by Gurobi's quadratic programming with quadratic complementarity problem (QPQCP) environment using the interface *gurobipy* in Python. The purpose of this step is to determine the optimal customer point partitioning scheme, providing a solid foundation for subsequent route planning.

In the route planning stage within the delivery area, an improved ALNS algorithm is used. The VRP has been proven to be a type of NP-hard problem. When considering learning curves and demand

uncertainty, the VRP introduces additional constraints, further increasing its complexity. The improved ALNS aimed at efficiently solving such PDPs. This algorithm seeks effective solutions through a series of operations such as initial solution generation, design of the destruction and repair operators, and the local search process. The termination condition of the algorithm is set to achieve the predetermined maximum number of iterations. The pseudocode of the algorithm details the implementation process of these steps. The pseudocode is shown in Table 3.

Table 3. Improved ALNS algorithm pseudocode.

Algorithm 1: Improved ALNS algorithm pseudocode	
1	Input: number of cluster centers, number of units, unit demands, and other basic parameters.
2	Output: delivery areas and area route maps
3	Generate regional center points using the K-means clustering algorithm.
4	For day $t = 0$ to T do
5	If $(t \bmod h) == 0$ then
6	$\tau = t/h$
7	Predict the average demand for the next few days
8	Solve for the new vehicle delivery areas y^* for the next stage
9	Update the delivery area center points
10	Else
11	For $k = 0$ to K do
12	//ALNS
13	Generate several candidate initial routes (greedy insertion, nearest-neighbor, sweep, and random) and keep the best
14	Apply intra-route 2-opt and inter-route 2-opt to refine the initial solution
15	While Iteration limit not reached
16	While Inner iteration limit not reached
17	Roulette wheel selection for destruction and repair operators
18	Execute the destruction operator
19	Execute the repair operator and calculate the objective value
20	Apply intra-route 2-opt to the candidate solution
21	If $\text{new_sol.obj} < \text{sol.obj}$:
22	If $\text{new_sol.obj} < \text{best_sol.obj}$
23	$\text{best_sol.obj} = \text{new_sol.obj}$
24	Update the scores of the destruction and repair operators
25	Else
26	Update the scores of the destruction and repair operators
27	End
28	End
29	If the best solution has stagnated for several iterations:
30	Restart from a newly constructed initial solution
31	End
32	Save the optimal route
33	End

34	Update operator weights
35	Save the optimal route
36	End
37	Apply intra-route 2-opt and inter-route 2-opt to best solution
38	Plot the delivery areas and the delivery route map
39	End
40	End
41	End

4.2. Process implementation

4.2.1. Partitioning process

(1) Initialization of regional center points

The selection of initial center points has a significant impact on the partitioning of areas. If the center points are too densely clustered, it may lead to uneven and overlapping regions; conversely, if they are too dispersed, it may result in excessively large delivery areas for the couriers. Therefore, this study uses K-means algorithm-based method to generate the initial regional center points, aiming for a more uniform and reasonable area allocation.

(2) Solving the model using Gurobi solver

First, calculate the service time within units and the travel time between units, representing these as variables. Next, define the objective function to minimize the travel time cost and service time cost within the calculation period τ . Finally, use the QPQCP model from the *gurobipy* library in Python to solve the problem. When considering the learning curve, introduce a learning factor to adjust the travel time between units and the service time within units, more accurately reflecting the impact of the courier's learning effect on the travel time cost. This optimization model can obtain the optimal partitioning scheme while considering the learning effect.

4.2.2. Improved ALNS

Through the first stage of computation, multiple delivery areas are determined, meaning the overall problem can be transformed into multiple VRPs. In these VRPs, it is necessary to determine the specific delivery routes for couriers within each region to minimize overall costs and ensure that the couriers can effectively cover customers within their delivery areas. The specific steps are as follows:

(1) Initial solution construction

The initial solution is constructed using a greedy algorithm, which is a classic optimization algorithm typically used for solving combinatorial optimization problems. When solving problems with a large search space, the greedy algorithm constructs solutions by selecting the locally optimal choice at each step in the current state, making it suitable for constructing the initial solution. The greedy algorithm only selects the best result in the current state, so each greedy choice can refer to the previous choices made to iteratively approach the optimal solution. However, each choice never depends on future choices or the solutions to subproblems. To reduce the dependence on a single starting point, the construction is repeated from several candidate orders greedy insertion above, nearest-neighbor orders from randomly chosen starting customers, an angular sweep around the depot, and random permutations and the candidate with the smallest objective value is kept. The pseudocode for constructing the initial

solution is shown in Table 4.

Table 4. Construction of the initial solution pseudocode.

Algorithm 2: Construction of Initial Solution Pseudocode

```

1      Input: Basic information such as unit demand, unit coordinates, etc.
2      Output: Initial optimal route
3      Minimum distance = infinity
4      Insertion position = -1
5      Optimal route = empty list
6      For a unit in remove_list:
7          For the index in range(Current route length + 1):
8              New_sol = now_sol
9              New_sol (index).append(Unit)
10             If new_sol.obj < best_sol.obj:
11                 best_sol.obj = new_sol.obj
12                 Update insertion position to current position
13             End
14         End
15         Insert the current unit at the insertion position in the optimal route
16         Reset the minimum distance to infinity
17     End
18     Repeat from the nearest-neighbor, sweep, and random orders; keep the best
19     Split the sequence by dynamic programming into at most  $K$  routes
20     Return the resulting initial route

```

The split step decodes a unit sequence into at most K capacity-feasible routes through dynamic programming, and minimizing the travel distance plus capacity-violation penalties. This replaces a greedy fill, which can mis-evaluate sequences on tightly loaded instances by charging an unfavorable unit order to the sequence rather than to the filling.

The initial solution algorithm for the robust model considering customer demand uncertainty has the same basic steps as the learning effect model, with just a few adjustments: when splitting the route, the demand uncertainty needs to be considered.

(2) Destruction operator and repair operator

Based on the problem's characteristics, two sets of destruction operators and three sets of insertion operators were designed.

- Destroy operators

Random destroy: Randomly select a certain proportion of demand nodes from the current solution and remove them. The purpose of this operator is to disrupt the structure of the current solution by randomly removing some demand nodes, encouraging the search process to escape local optima, and providing a better optimization space for the subsequent repair operator.

Worst destroy: Select a certain proportion of demand nodes from the current solution that cause a significant increase in the objective function and remove them. This operator is similar to the random destroy operator, but it selects the nodes to be removed on the basis of the magnitude of

the change in the objective function. This aims to more specifically disrupt the structure of the current solution, thereby promoting a more effective search process.

- Repair operators

Random repair: Randomly insert the removed demand nodes into any position in the sequence of assigned nodes. This insertion strategy is simple and straightforward, without considering the impact of the insertion position on the objective function. It merely inserts the nodes into the existing sequence randomly, aiming to partially restore the original solution's structure.

Greedy repair: Insert the removed demand nodes into each possible position in the assigned node sequence according to the increase in the objective function. Sequentially select the combination of demand node and insertion position that results in the smallest increase in the objective function. This insertion strategy considers the impact of the insertion position on the objective function and reconstructs the original solution's structure by greedily choosing the combination with the smallest objective function increment.

Regret repair: Calculate the sum of the differences between the objective function values of several suboptimal positions and the optimal position when inserting a removed node back into the assigned node sequence. Then select the demand node and its optimal position with the largest sum of differences. This insertion strategy considers the relative merits of the insertion positions and prioritizes the insertion of the node and its optimal position that have the greatest impact on the solution structure, aiming to more effectively optimize the current solution's structure.

When the greedy and regret repair operators evaluate candidate insertion positions, the cost of a position is estimated by the marginal distance increment it induces, which is computable in constant time; the exact objective of the repaired solution is computed once after the insertion is fixed. This keeps the repair step inexpensive while preserving the selection logic described above.

(3) Operator selection strategy

Using the roulette wheel algorithm to select the appropriate operator based on their weights involves the following steps:

Step1: Use the equation to calculate the selection probability P_k for each operator, where w_k is the weight of operator k ;

Step2: Use the equation to calculate the cumulative probability Q_k for each operator;

Step3: Generate a random number p within the interval $(0, 1]$. If $Q_{k-1} < p < Q_k$, then select the operator k .

$$P_k = \frac{w_k}{\sum_{i=1}^n w_i} \quad (58)$$

$$Q_k = \sum_{i=1}^k P_i \quad (59)$$

(4) Embedded local search

To further improve the solution's quality, the improved ALNS incorporates intra-route 2-opt and inter-route 2-opt as embedded local search operators. Intra-route 2-opt reverses a segment of consecutive customers within a single route to reduce its travel distance, while inter-route 2-opt exchanges the tail segments between two routes, enabling structural reorganization across route boundaries subject to capacity feasibility.

Three refinement phases are applied throughout the algorithm. Before the main loop, the constructed initial solution is refined by sequentially applying both operators, ensuring that the search begins from a locally optimal starting point. During the main loop, intra-route 2-opt is applied to each candidate solution immediately after repair and before the acceptance criterion; the objective is recomputed as total route distance plus a capacity violation penalty per overloaded route, ensuring that acceptance decisions are always made on locally refined, feasibility-aware candidates and preventing infeasible solutions from being incorrectly retained as the incumbent.

After the main loop, a post-processing phase alternately applies intra-route 2-opt, inter-route 2-opt, and or-opt (which relocates a segment of up to three customers) to the best solution for up to three passes, stopping once no further improvement is obtained.

(5) Acceptance criteria

Due to the presence of capacity constraints and maximum working time constraints, there are many infeasible solutions within the solution space. The acceptance criterion for new solutions in the neighborhood follows the threshold acceptance (TA) rule shown in Figure 5, where suboptimal solutions are selectively accepted with a certain probability.

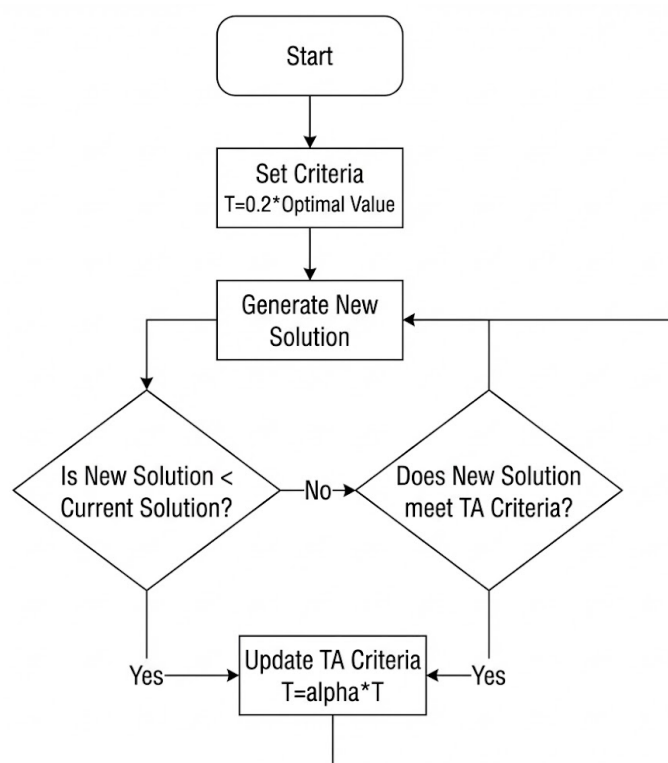


Figure 5. Acceptance criteria chart.

The TA rule is expressed as follows: When the objective function value of the new solution minus the objective function value of the current solution is less than a threshold T , the new solution is accepted. The threshold T is attenuated according to a coefficient ϕ , reducing the probability of accepting worse solutions over time, thereby guiding the algorithm towards more optimal solutions. The general ALNS sets the initial threshold T to 10% of the objective function value of the current solution, whereas the improved ALNS tightens the rule to an improvement-only (strict) acceptance, i.e., $T \equiv 0$. Since the

embedded local search already refines every candidate solution, admitting only improving candidates yields the best performance, as confirmed by the ablation study in Section 5.1.1. Escape from local optima is instead provided by a stagnation-triggered restart: whenever the best solution fails to improve for a number of iterations, the search is restarted from a newly constructed initial solution.

(6) Weight update strategy

Let δ be the weight update cycle, w_k be the weight of operator k , a_k be the cumulative score of operator k over δ iterations, b_k be the number of times operator k is used over δ iterations, and ρ be the weight update coefficient. The initial weight of all operators is 1. If the current iteration count reaches δ , update the weight of all operators using Formula (60) and reset the iteration count to 0; otherwise, use Formulas (60) and (61) to update the cumulative score of the operator for this iteration.

$$w_k = \begin{cases} (1 - \rho) w_k + \rho \frac{a_k}{b_k}, & b_k \neq 0 \\ w_k, & b_k = 0 \end{cases} \quad (60)$$

$$a_k = a_k + \begin{cases} r_1, & \text{new solution better than optimal} \\ r_2, & \text{new solution better than current} \\ r_3, & \text{new solution worse but accepted} \end{cases} \quad (61)$$

5. Numerical experiments

Considering there are no standard instances for the proposed problem, the performance of the proposed improved ALNS is evaluated on the basis of benchmark instances set of Christofides and Eilon [42]. All instance experiments are conducted on a personal computer with an Intel(R) Core(TM) i5-10500 central processing unit (CPU) @ 3.10GHz with 16 GB of random access memory (RAM). The mathematical models were implemented using Python 3.7.9 with Gurobi 12.0.1. All instances were run 10 times and the average results are reported. The performance of the improved ALNS is analyzed in Section 5.1. The effects of the learning curves and demand uncertainty are described in Sections 5.2, 5.3, and 5.4. The proposed heuristic is evaluated in practice in Section 5.5.

5.1. Performance of improved ALNS

In our two-stage framework, the first-stage MIQP model is solved exactly by Gurobi at every instance size; only the second-stage routing problem is solved by the ALNS heuristic. There is consequently no heuristic counterpart to the first-stage solution to compare against. The results of first stage are used as input to the second-stage routing. This section first evaluates the performance of the improved ALNS. Since the improved ALNS is used only to solve the second-stage vehicle routing problem, we compare the proposed algorithm against three benchmark heuristics on the instances of Christofides and Eilon [42]. The instance set contains 13 instances. The number of customers ranges from 12 to 100, and the number of vehicles from 3 to 14. The competitors are GA, TS, and a general ALNS. All methods were run 10 times under an identical per-instance time budget, from 12 s on the smallest instance to 150 s on the largest. The benchmark heuristics are described as follows.

The three benchmarks are configured as follows. GA encodes a solution as a permutation chromosome overall customer nodes and decodes it into capacity-feasible routes via a split procedure; the population

size is $\min(80, \max(30, 2N))$ (with three elite individuals preserved per generation, where N denotes the number of customer units), selection uses tournament selection with 5, crossover uses order crossover with probability 0.80, and mutation is applied with probability 0.25, drawing randomly from segment reversal, random swap, and node insertion. TS defines its neighborhood by node relocation, sampling at most $\min(200, 5N)$ candidate moves per iteration; the tabu list records recent (node, source route) pairs with a tenure of $\max(7, \lfloor N/4 \rfloor)$, an aspiration criterion accepts any move that improves on the historical best, the search restarts from the historical best with a cross-route random perturbation after $\max(80, 2N)$ non-improving iterations, and all moves are checked against the capacity constraint. The general ALNS shares the same destroy-and-repair operator set, adaptive weight-updating mechanism (decay coefficient ρ), and threshold acceptance (TA) criterion. The improved ALNS differs from the general ALNS through three components: (i) an embedded local search, labeled as (LS), that applies intra-route 2-opt, inter-route 2-opt, and Or-opt, (ii) an improvement-only (strict) acceptance criterion (AC) that admits only improving candidates (versus the threshold acceptance with an initial threshold of 10% in the general ALNS), and (iii) a final post-processing (PP) phase.

Table 5. Comparison results with three benchmark heuristics.

Instance	N	BKS	GA		TS		ALNS		Improved ALNS		Time (s)
			Cost	Gap	Cost	Gap	Cost	Gap	Cost	Gap	
E-n13-k4	12	247	249	0.65%	247	0.00%	247	0.00%	247	0.00%	12s
E-n22-k4	21	375	375	0.00%	375	0.00%	377	0.53%	375	0.00%	30s
E-n23-k3	22	569	570	0.14%	569	0.00%	570	0.21%	569	0.00%	30s
E-n30-k3	29	534	558	4.40%	534	0.00%	547	2.42%	537	0.52%	30s
E-n31-k7	30	379	489	28.92%	415	9.50%	536	41.35%	390	3.01%	60s
E-n33-k4	32	835	869	4.12%	835	0.00%	850	1.81%	835	0.00%	60s
E-n51-k5	50	521	580	11.38%	619	18.75%	544	4.41%	525	0.83%	60s
E-n76-k7	75	682	710	4.11%	708	3.77%	727	6.64%	687	0.76%	120s
E-n76-k8	75	735	791	7.55%	772	5.03%	785	6.78%	746	1.50%	120s
E-n76-k10	75	830	930	12.01%	1049	26.40%	899	8.31%	867	4.41%	120s
E-n76-k14	75	1021	1133	10.95%	1311	28.35%	1748	71.17%	1097	7.48%	120s
E-n101-k8	100	817	854	4.52%	851	4.16%	869	6.33%	822	0.64%	150s
E-n101-k14	100	1071	1158	8.10%	1147	7.07%	1154	7.71%	1106	3.23%	150s
Average				7.45%		7.93%		12.13%		1.72%	

Table 5 reports the results on the 13 instances, with solution quality measured as the percentage gap to the best-known solution (BKS). The improved ALNS achieves the lowest average gap at 1.72%, followed by GA at 7.45%, TS at 7.93%, and the general ALNS at 12.13%. The improved ALNS achieves the best or tied-best gap on 12 of the 13 instances and matches the BKS in every run on four of them (E-n13-k4, E-n22-k4, E-n23-k3, and E-n33-k4); the only exception is E-n30-k3, where TS is marginally better. The advantage is most pronounced on the six largest instances ($N \geq 75$), where the improved ALNS averages a 3.00% gap against 7.87% for GA, 17.82% for the general ALNS, and 12.46% for TS, and its gap never exceeds 7.5% on these instances while the baselines degrade sharply on E-n76-k10 and E-n76-k14. On E-n31-k7, where a large fleet serves few customers, the improved ALNS limits the

gap to 3.01% (with one run matching the BKS of 379), well below TS (9.50%), GA (28.92%), and the general ALNS (41.35%). Notably, the general ALNS alone does not guarantee competitive performance, as it is outperformed by GA and TS as well as by the improved ALNS; the gains therefore stem from the proposed enhancements. Overall, the results support the effectiveness and scalability of the improved ALNS for the second-stage routing problem.

5.1.1. Ablation study

To isolate the contribution of each individual component in the improved ALNS, we conducted an ablation study on the same 13 CVRP instances. Table 6 reports the gap (%) to the best-known solution (BKS) under five configurations: the full improved ALNS (LS + AC + PP), three variants each removing one component (without LS, without AC, and without PP), and the general ALNS that omits all three enhancements. The last three columns (Δ LS, Δ AC, Δ PP) directly measure the degradation incurred by dropping each component. All five configurations share the same destroy-and-repair operators, adaptive weight-updating mechanism, initial-solution construction, restart mechanism, and time budget, so the observed differences are attributable solely to LS, AC, and PP. Each configuration is evaluated over 10 independent runs, and the reported gaps are the per-instance averages across these runs. The instance-level Δ values are computed from those averages; a few entries remain slightly negative because the mean of one configuration can still fall below another on a single instance, while the averages across all 13 instances capture the consistent component-level trend.

Table 6. Ablation study: gap (%) to BKS under five algorithm configurations.

Instance	Full	w/o LS	w/o AC	w/o PP	ALNS	Δ LS	Δ AC	Δ PP
E-n13-k4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E-n22-k4	0.00	0.11	0.00	0.00	0.53	0.11	0.00	0.00
E-n23-k3	0.00	0.18	0.00	0.00	0.21	0.18	0.00	0.00
E-n30-k3	0.52	1.48	0.51	0.39	2.42	0.96	-0.02	-0.13
E-n31-k7	3.01	18.18	8.94	4.20	41.35	15.17	5.94	1.19
E-n33-k4	0.00	0.85	0.02	0.00	1.81	0.85	0.02	0.00
E-n51-k5	0.83	2.03	1.59	0.88	4.41	1.21	0.77	0.06
E-n76-k7	0.76	3.65	1.39	0.70	6.64	2.89	0.63	-0.06
E-n76-k8	1.50	5.56	2.08	1.39	6.78	4.07	0.59	-0.11
E-n76-k10	4.41	6.05	4.30	4.34	8.31	1.64	-0.11	-0.07
E-n76-k14	7.48	23.99	9.52	8.67	71.17	16.50	2.04	1.19
E-n101-k8	0.64	2.73	1.15	0.89	6.33	2.09	0.51	0.26
E-n101-k14	3.23	4.65	3.39	3.00	7.71	1.42	0.16	-0.23
Avg.	1.72	5.34	2.53	1.88	12.13	3.62	0.81	0.16

On average, the full configuration achieves a 1.72% gap. Removing LS causes the largest degradation (+3.62 percentage points (pp), to 5.34%), with severe drops on E-n76-k14 (+16.50 pp) and E-n31-k7 (+15.17 pp); the degradation is positive on every instance (paired comparison across the 13 instances, $p = 0.03$), indicating that local search is the primary driver of the solution's quality. Replacing the strict acceptance with the 10% threshold acceptance of the general ALNS degrades the average gap by +0.81 pp (to 2.53%), again with the largest losses on E-n31-k7 (+5.94 pp) and E-n76-k14 (+2.04 pp);

threshold acceptance at 5% or 10% performs comparably (2.60% vs 2.53%), and both are clearly worse than the strict rule (1.72%). Removing PP has a modest average impact (+0.16 pp), concentrated on the harder instances (E-n31-k7 and E-n76-k14, each +1.19 pp). The general ALNS, which drops all three components, reaches a 12.13% average gap, 10.41 pp above the full configuration, with positive degradation on every instance. Overall, LS is the dominant contributor, the strict acceptance provides a consistent secondary gain, and PP offers complementary improvement on harder problems; all three components have a positive average degradation, so none is harmful.

5.2. PDP considering learning curves

5.2.1. Performance of reformulated MIQP

This subsection evaluates the effects of learning on the results. Four representative cases ($N=31, 50, 75, 100$) are selected for subsequent in-depth analysis. The instances are extended to cover $T = 28$ days and $H = 4$ periods, with demand subject to a maximum 20% fluctuation everyday. Initially, the learning potential parameter α and the decay factor β are set to 0.1 and 0.2, respectively.

Table 7 reports the exact solutions to the first-stage MIQP partitioning model, obtained directly by Gurobi on four small-scale instances ($N = 31, 50, 75$, and 100 customer unit), each evaluated over four periods. In all cases, Gurobi solved the model to global optimality in well under half a second (0.22–0.36 s), confirming that the reformulated MIQP model is indeed directly and efficiently solvable. The Obj-MIQP represents the first-stage objective, namely, the minimization of within-unit service time and between-unit travel time, when the learning effect is taken into account.

Table 7. Results of the reformulated MIQP for first stage.

N	Partition	Period	Obj-MIQP	Time (s)
31	2	4	1722.89	0.22
50	2	4	2390.06	0.29
75	3	4	2573.01	0.34
100	4	4	3013.86	0.36

Figure 6 reports the customer-to-driver partitions for instance $N = 100$ across its four planning periods. Dark-outlined dots mark customers reassigned to a different driver relative to the previous period. Assignments are not perfectly stationary: Five, seven, and seven customers switch drivers between consecutive periods (periods 1–2, 2–3, and 3–4, respectively), while cluster sizes fluctuate mildly around 25 customers per driver (28/27/19/26 $\rightarrow$ 28/25/18/29 $\rightarrow$ 29/22/20/29 $\rightarrow$ 27/22/19/32). This reassignment reflects the learning-effect dynamics embedded in the model: as couriers accumulate experience within a service area, the marginal cost of serving that area declines, shifting the relative cost-efficiency of alternative partitions across periods. The optimal partition thus evolves incrementally to capture these learning gains, rather than remaining fixed — consistent with the model's premise that partition design and courier learning are jointly coupled.

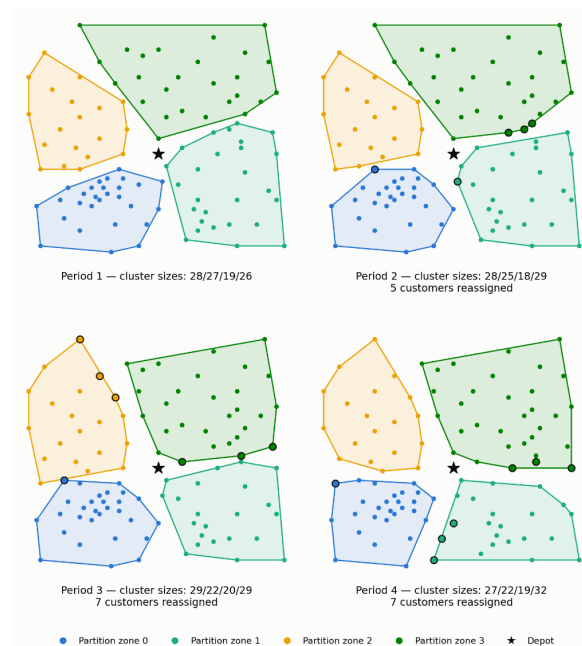


Figure 6. Customer-to-driver partitions for instance $N = 100$.

5.2.2. Sensitivity analysis

To study the impact of the learning potential parameter α and the decay factor β , this study categorizes each parameter into high, medium, and low. Specifically, $\alpha = 0.1, 0.3, 0.5$ represent low, medium, and high learning potentials, respectively; and $\beta = 0.2, 0.4, 0.6$ represent low, medium, and high decay factors, respectively. The computational results at different levels of learning potential and decay factors are shown in Table 8.

In this table, the “PDP” represents the known optimal solution for partitioned distribution without considering learning curves, the “PDP-LC” represents the sum of the travel time cost and service time cost per day on average during the period when considering learning curves, the “GAP” represents the percentage improvement of the optimal value considering learning curves compared to the best-known solution, and the “Time” represents the average time.

It shows that the average values of all four cases considering learning curves outperform the best-known solution without them, with the improvement becoming more significant as customer point scale increases. This confirms the critical role of learning curves in VRPs: ignoring them may miss better path opportunities. Thus, decision-makers should incorporate learning curves into route planning to optimize routes, reduce costs, and enhance efficiency, and practical routing models should integrate learning effects.

We further use the $N = 31$ instance to isolate the effects of the learning potential and decay parameters on courier travel time, as shown in Figure 7.

In Figure 7(a), the horizontal axis represents the number of days, and the vertical axis represents travel time. Assuming all couriers have low learning potential and medium learning efficiency (i.e., $\alpha = 0.1, \beta = 0.6$), it can be seen that the travel time is relatively less affected by the learning effect when the learning potential is low.

Table 8. Results of PDP considering learning curves.

(α, β)	N	Capacity	PDP	PDP-LC	GAP(%)	Time(s)
(0.1, 0.2)	31	100	788	746.52	5.26	95.25
	50	160	561	510.70	8.97	899.43
	75	220	725	676.74	6.66	922.46
	100	200	894	853.92	4.48	828.61
(0.3, 0.2)	31	100	788	736.36	6.55	72.77
	50	160	561	490.25	12.61	746.49
	75	220	725	667.35	7.95	795.89
	100	200	894	835.30	6.57	995.32
(0.5, 0.2)	31	100	788	694.73	11.84	74.92
	50	160	561	492.20	12.26	856.07
	75	220	725	648.66	10.53	727.79
	100	200	894	827.15	7.48	812.26
(0.1, 0.4)	31	100	788	749.31	4.91	69.63
	50	160	561	509.16	9.24	922.80
	75	220	725	693.15	4.39	993.63
	100	200	894	854.76	4.39	962.26
(0.3, 0.4)	31	100	788	713.23	9.49	67.64
	50	160	561	477.94	14.81	609.53
	75	220	725	666.88	8.02	838.29
	100	200	894	833.74	6.74	961.04
(0.5, 0.4)	31	100	788	672.82	14.62	64.99
	50	160	561	495.44	11.69	623.14
	75	220	725	614.40	15.26	761.04
	100	200	894	808.36	9.58	957.75
(0.1, 0.6)	31	100	788	755.58	4.11	89.97
	50	160	561	509.51	9.18	681.62
	75	220	725	687.65	5.15	903.81
	100	200	894	853.73	4.50	1126.33
(0.3, 0.6)	31	100	788	701.73	10.95	78.32
	50	160	561	503.17	10.31	646.20
	75	220	725	651.46	10.14	855.53
	100	200	894	826.28	7.57	1022.71
(0.5, 0.6)	31	100	788	668.16	15.21	68.22
	50	160	561	457.20	18.50	725.53
	75	220	725	635.22	12.38	865.65
	100	200	894	799.23	10.61	1147.97

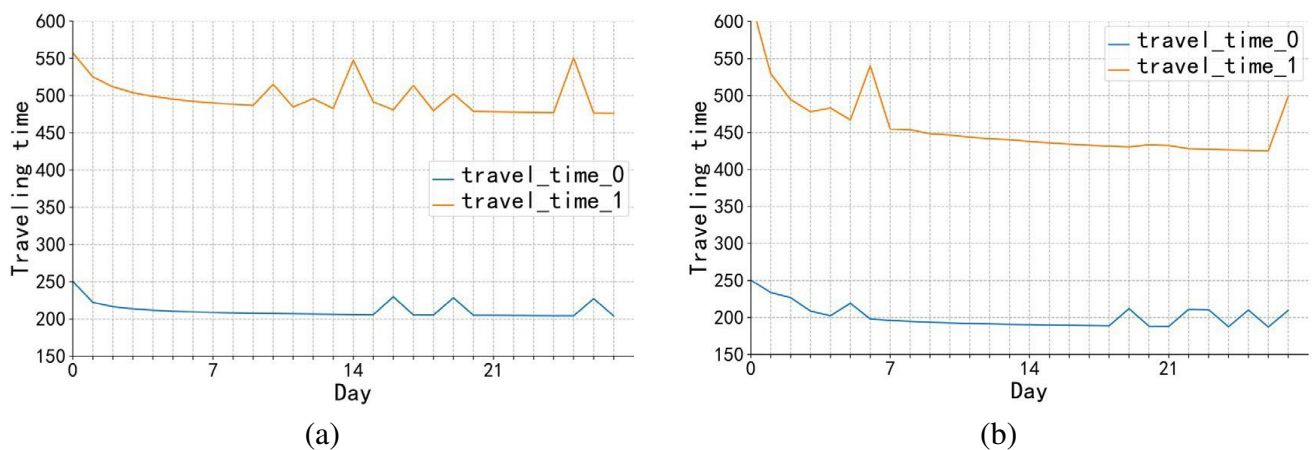


Figure 7. Travel time variation under different learning potentials.

Low learning potential in couriers may lead to three key impacts: first, poor adaptation to routes and customer demands, causing longer adaptation time, frequent route adjustments, and increased travel time uncertainty, which gradually stabilizes with familiarity; second, slow absorption of new skills/knowledge, hindering effective response to complex scenarios during route execution and prolonging adaptation periods; third, conservative route choices (avoiding new routes) to reduce short-term time fluctuations but missing more efficient options. In summary, low learning potential limits couriers' adaptability and increases travel time variability, though experience accumulation mitigates these issues. Thus, couriers' learning potential should be integrated into route planning to optimize travel time and route efficiency.

When couriers have high learning potential (i.e., $\alpha = 0.5$, $\beta = 0.6$), Figure 7(b) shows smoother travel time changes with smaller variance than Figure 7(a), indicating that higher courier learning potential mitigates travel time variability. Medium learning potential correlates with moderate travel time, and increased learning potential enhances couriers' adaptability to routes/demands, ability to master new skills, and willingness to explore efficient new routes—all contributing to more stable and predictable travel times. In contrast, lower learning potential prolongs adaptation periods and increases time variance. In conclusion, learning potential significantly impacts couriers' travel performance, providing valuable insights for optimizing travel time.

Next, we examine the impact of decay factors on travel time. Assuming all couriers have medium learning potential ($\alpha = 0.3$), Figure 8 shows the optimal travel time under different decay factors.

Figure 8 shows that the decay factor significantly impacts couriers' travel time: with a low decay factor ($\beta = 0.2$), travel time decreases slowly and linearly (Figure 8(a)), as couriers adopt cautious, conservative driving to ensure stability; with a high decay factor ($\beta = 0.6$), travel time drops sharply (Figure 8(b)), due to couriers' enhanced adaptability to routes/demands and improved driving efficiency. This is because lower β induces conservative behavior, while higher β accelerates skill mastery and efficiency. In summary, higher decay factors speed up travel time reduction, boosting overall productivity, which provides a theoretical basis for optimizing couriers' driving behavior.

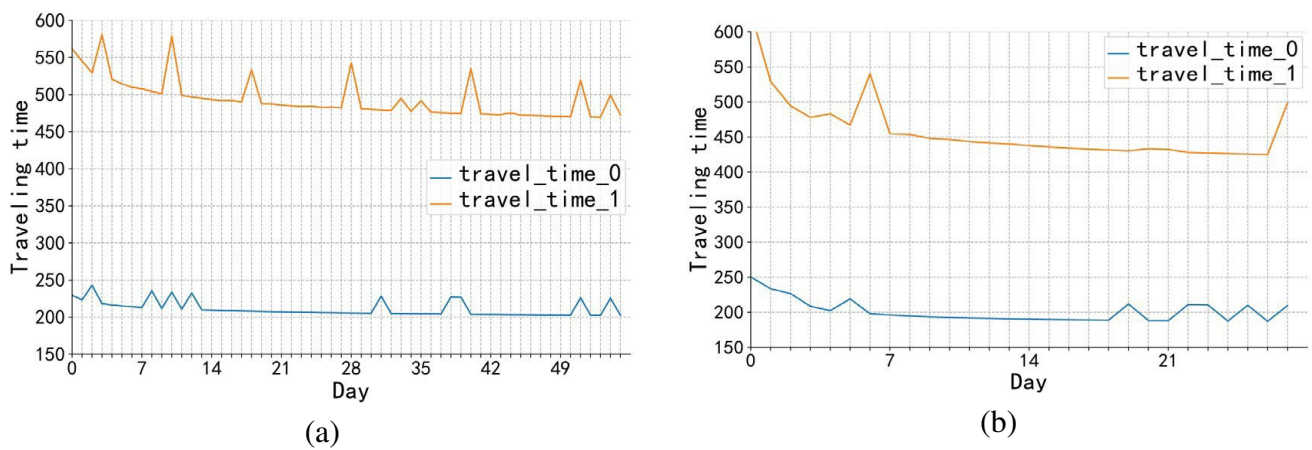


Figure 8. Travel time variation under different decay factors.

5.3. PDP considering demand uncertainty

5.3.1. Example design

The following example set variant is still based on the proposed benchmark set. The deterministic model retains the coordinate positions, demand, and vehicle capacity from the benchmark set, while the robust model assumes uncertain customer demand with a maximum deviation of $\gamma_i = 0.2\bar{w}_{it}$ for individual customers. The uncertain demand budget on a single route handled by each courier per day is $\Gamma_m = \sqrt{N_k/|M_k|} \gamma_i$, where N_k is the number of customers allocated to region k per day, and $|M_k|$ is the number of times the courier in region k departs from the distribution center. Vehicles are subject to maximum load constraints, with overload permitted only for the last departure from the distribution center (incurring overload costs). Travel time is calculated as regular working hours, so the unit travel time cost is set to 1 times the wage. Given significant safety hazards of overload, the unit overload cost is set to 10 times the wage.

5.3.2. Optimization results and Monte Carlo simulation evaluation

To evaluate the operational performance and inherent resilience of our robust framework, this subsection consolidates the optimization and simulation results under demand uncertainty into a unified analysis. First, we investigate the mathematical performance across four distinct instance scales ($N = 31, 50, 75$, and 100). For each scale, the maximum deviation of uncertain demand for a single customer is $\gamma_i = 0.2\bar{w}_{it}$, and the uncertain demand budget on a single route handled by each courier per day is $\Gamma_m = \sqrt{N_k/|M_k|} \gamma_i$. We then solve and compare the optimization results under both deterministic scenarios and robust uncertainty sets. The average results over 10 runs are shown in Table 9.

It shows that solutions under deterministic customer demand have lower total costs, while robust analysis (accounting for worst-case scenarios like demand fluctuations) leads to a significant cost increase. This is because deterministic solutions—based on known demand data—are precise, feasible, and cost-optimized, whereas robust solutions require additional resources to mitigate potential risks. The phenomenon reflects the trade-off between certainty and robustness: deterministic solutions are reliable but lack anti-uncertainty capability, while robust solutions cover more scenarios at the cost of higher expenses. Thus, in practice, a balance between the two should be struck based on specific needs.

to achieve optimal results.

Subsequently, to test the real-world execution stability of the generated strategies, a Monte Carlo simulation framework is deployed to evaluate the true robustness of both the robust optimal solutions and their deterministic baselines. Here, it is assumed that customer demand follows a normal distribution $w_{it} \sim N(\bar{w}_{it}, (0.2\bar{w}_{it})^2)$, where the second argument denotes the variance, i.e., the standard deviation of w_{it} is $0.2\bar{w}_{it}$, generating 1000 random customer demand scenarios based on this. Sampled demands are truncated at zero to ensure nonnegativity; given the coefficient of variation of 0.2, negative draws occur with negligible probability and truncation does not materially affect the evaluation. The results indicate that considering robustness may improve or degrade the solution's quality—robust solutions are not universally optimal and require scenario-specific analysis. For instance, with 31 customer points (Figure 9(a) and (b)), robustness leads to longer paths, excessive capacity surplus, and higher total costs than deterministic solutions; however, with 50 customer points (Figure 9(c) and (d)), robust solutions outperform deterministic ones by reducing overload costs. In summary, there is no one-size-fits-all relationship between the two solution types. In practice, the choice between robust and deterministic solutions should be guided by the instances' characteristics such as customer density and the cost of capacity violations.

Table 9. Results of Monte Carlo simulation.

N	Vehicle Capacity	Type	Optimal Cost	Evaluation Value
31	100	Deterministic	794	826
	100	Robust	854	854
50	160	Deterministic	635	666
	160	Robust	665	659
75	220	Deterministic	730	754
	220	Robust	886	762
100	200	Deterministic	899	997
	200	Robust	1061	1015

5.3.3. Sensitivity analysis

The robustness of the solution is closely related to the uncertainty parameters in the uncertainty set. Therefore, this study conducts sensitivity analysis on the maximum deviation of uncertain demand for a single customer γ_i and the uncertain customer demand budget Γ_m on each route handled by each courier per day.

Maximum Deviation of Uncertain Demand for a Single Customer γ_i . Set the uncertain demand budget on each route handled by each courier to $\Gamma_m = 2$, with $\gamma_i = \xi \bar{w}_{it}$. We study the changes in evaluation values as the coefficient ξ varies between 0 and 0.3. The results are shown in Figure 10. It can be seen that as ξ changes from 0 to 0.1, the evaluation value decreases, indicating that robust optimization can improve the performance of the solution under uncertain demand. However, as ξ changes from 0.1 to 0.3, the evaluation value increases, suggesting that being overly conservative can lead to poorer solution quality.

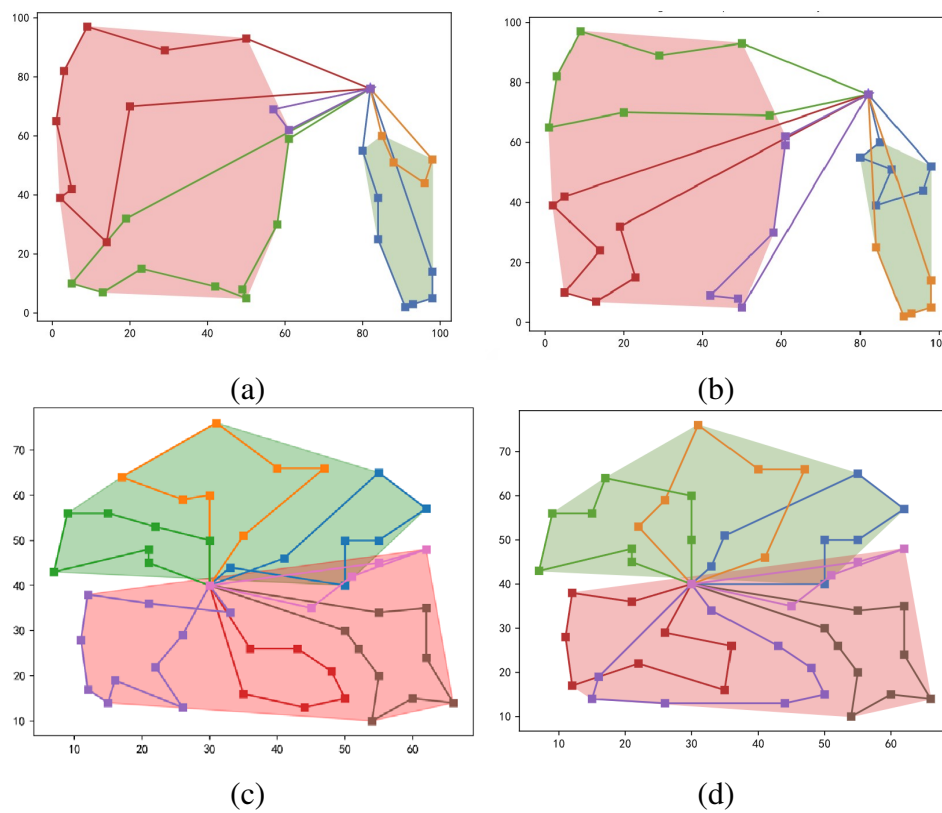


Figure 9. Comparison of robust and deterministic solution routes.

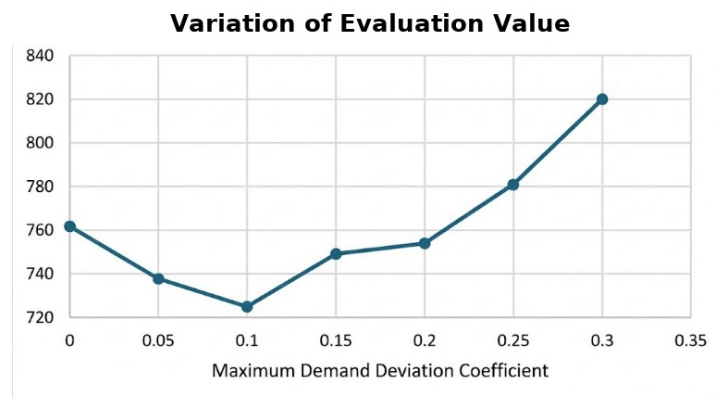


Figure 10. Variation of evaluation values with maximum deviation coefficient.

The figure shows that solution quality varies with the evaluation value: in the range of 0–0.1, quality decreases, indicating that robust optimization enhances the solution's stability and reliability under uncertain demand as robustness increases; in 0.1–0.3, quality improves, meaning excessive conservatism (overemphasizing uncertainty) leads to poorer optimization due to additional costs or overly cautious strategies. This reflects the trade-off in robust optimization: moderate robustness boosts adaptability to uncertainty, while excessive conservatism limits optimization space. Thus, balancing uncertainty impact with solutions' feasibility and cost-effectiveness is crucial for optimal results in robust optimization.

Uncertain Demand Budget on Each Route Handled by Each Courier Per Day Γ_m . Set the maximum deviation of uncertain demand for a single customer γ_i to $\gamma_i = 0.2\bar{w}_{it}$, and let $\Gamma_m = \lambda \sqrt{N_k/|M_k|} \gamma_i$. We study the changes in evaluation values as the coefficient λ varies between 0 and 0.5. The results are shown in Figure 11.

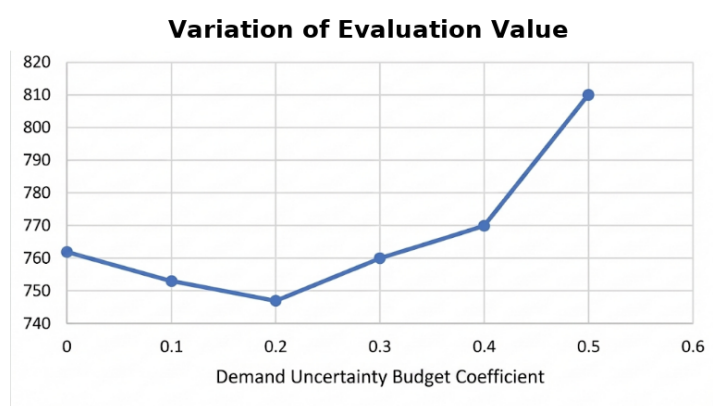


Figure 11. Variation of evaluation values with uncertain demand budget coefficient.

Figure 11 shows the evaluation value decreases first and then increases over time, indicating that excessive conservatism in robust optimization impairs solutions' quality. This trend highlights the critical need to balance risk and reward: a moderate level of conservatism enhances the solution's performance under uncertainty (reflected by the initial decrease in evaluation value), while excessive conservatism leads to missed optimization opportunities and poorer quality (evidenced by the subsequent increase). Thus, decision-makers must comprehensively consider uncertainty factors, strike a balance between robustness and optimization, and ensure the solution is both resilient to uncertainty and maximizes the objective function.

5.4. PDP considering learning curves and demand uncertainty

In this section, the variant of the example set used below is still based on the proposed benchmark example set. We also use 75 customer points, and the results are averaged over 10 runs. In the model considering learning curves and customer demand uncertainty, to facilitate the observation of the impact of learning efficiency, this section assumes a learning potential factor $\alpha = 0.3$ and a decay factor $\beta = 0.6$. Customer demand is uncertain, with the maximum deviation of uncertain demand for a single customer being $\gamma_i = 0.2\bar{w}_{it}$; the uncertain demand budget on each route handled by each courier per day is $\Gamma_m = \sqrt{N_k/|M_k|} \gamma_i$. The computational results are presented in Table 10.

Table 10. Results of PDP considering learning curves and demand uncertainty over the period.

Date	Region 1	Region 2	Region 3	Total
1	281.55	270.84	231.89	784.28
2	264.52	265.65	214.05	744.22
3	259.41	255.04	220.13	734.57
4	247.59	288.19	194.61	730.39
5	260.90	255.52	224.92	741.35
6	267.45	245.92	234.23	747.60
7	252.69	257.21	230.98	740.89
8	248.43	248.95	220.34	717.73
9	269.89	262.31	212.46	744.66
10	248.51	236.54	200.46	685.50
11	266.01	251.35	201.55	718.91
12	244.35	237.41	195.15	676.91
13	234.84	265.56	213.16	713.57
14	232.38	254.30	215.73	702.42
15	240.11	236.14	204.15	680.40
16	249.18	247.90	201.03	698.12
17	235.93	243.97	202.46	682.36
18	244.31	235.10	209.37	688.79
19	250.67	256.33	223.28	730.29
20	248.48	245.74	209.42	703.64
21	228.92	229.50	205.94	664.36
22	227.52	268.49	195.72	691.74
23	224.78	240.36	192.12	657.26
24	246.25	254.44	202.74	703.44
25	228.39	232.99	183.59	644.98
26	241.89	237.87	198.48	678.23
27	230.14	245.71	199.82	675.67
28	225.01	232.19	195.01	652.21

Combining robustness and learning effects results in a total cost slightly higher than that with only the learning effect. This is because robustness introduces system stability and reliability, typically involving extra costs (e.g., redundant equipment, backup resources) that partially offset the long-term cost reduction from learning effects. However, these additional costs act as risk insurance, boosting the system's resilience to uncertainties and ensuring long-term stability and sustainability of the production process.

5.5. Case study

5.5.1. Case description

This paper constructs the case using actual data from Xi'an Y Logistics Courier Company, which covers Xi'an's districts and counties with around 301 courier points. For research convenience, only one

distribution center is assumed. A Python web crawler was written to obtain the latitude and longitude of courier points from the Baidu Maps Web API, and convert them into actual distances (including inter-point distances and distances between points and the distribution center). Partial coordinate data is presented in Table 11, with the default demand for each courier point set to 1.

Table 11. Unit distance information.

Node Information	Node Dimension	Node Longitude	Node X Coordinate	Node Y Coordinate	Node Demand
Distribution Center	108.9478	34.26803	80.00	40.00	0
Unit 1	108.9258	34.34972	98.86	36.65	1
Unit 2	108.9812	34.38322	96.4	29.64	1
Unit 3	108.9417	34.29836	102.57	26.77	1
Unit 4	108.917	34.35813	98.17	34.05	1
Unit 5	109.0044	34.32623	95.42	28.92	1
Unit 6	108.977	34.28956	105.15	31.66	1
Unit 7	108.9108	34.30081	102.1	34.8	1
Unit 8	109.019	34.36629	94.73	33.84	1
Unit 9	108.9588	34.33891	106.79	28.22	1
Unit 10	108.9216	34.34062	100.08	30.57	1
Unit 11	108.8253	34.31719	95.93	30.42	1
⋮	⋮	⋮	⋮	⋮	⋮
Unit 293	108.9775	34.41585	102.16	23.97	1
Unit 294	109.2412	34.65504	131.53	3.44	1
Unit 295	108.9413	34.22766	98.13	40.1	1
Unit 296	108.8913	34.1736	92.56	44.73	1
Unit 297	108.8939	34.16315	92.85	45.63	1
Unit 298	108.0964	34.18142	4.03	44.06	1
Unit 299	109.2147	34.63071	128.58	5.53	1
Unit 300	108.9417	34.18611	98.17	43.66	1
Unit 301	108.078	34.12899	1.98	48.55	1

5.5.2. Distribution unit division and route optimization results for Y Courier Company

(1) Distribution Area Division

Given the distance matrix between units and the unit demand, the first stage of the proposed algorithm can be used for partitioning. In the first stage, parameter values are set as follows: Based on the research results of Stein et al. [16], the parameters for calculating the service time within a unit are $\omega_{11} = 0.77, \omega_{12} = 0, \omega_{13} = 0.88$; the minimum number of units in each distribution area is $I_0 = 50$; the maximum working hours in each distribution area are $b = 3000$, and the minimum working hours parameter are $\theta = 0.1$; The total planning horizon is $T = 28$, with one distribution area division every $h = 7$ days. The partition of regions is presented in Figure 12.

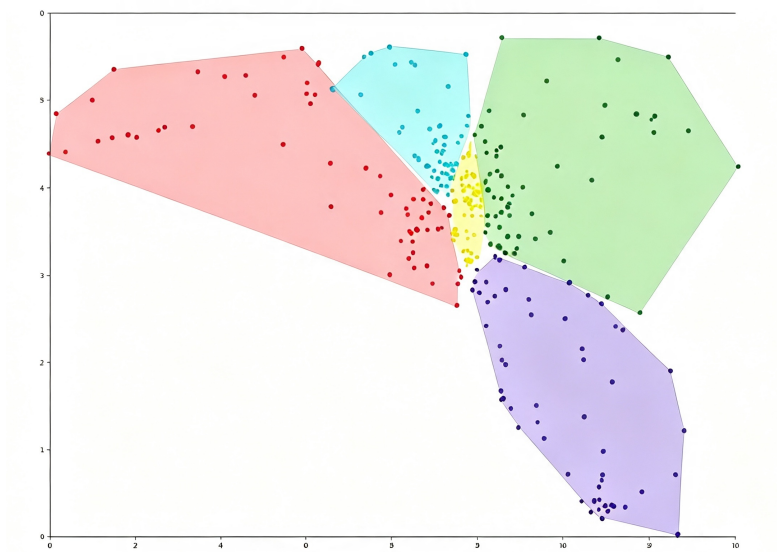


Figure 12. Distribution area division map.

(2) Route Planning within Distribution Areas

After solving for the distribution areas and their included unit points in the first stage, the second stage involves specific route planning for each area. In the second stage, the parameter values for the improved ALNS are set as follows: the lower limit of random destruction is 0.2, the upper limit is 0.5, the lower limit of the worst destruction is 1, the upper limit is 6, the number of suboptimal positions is 3, operator rewards are 1.5, 1.0, 0.5, operator weight decay factor is 0.1, threshold decay factor is 0.7, inner iteration count is 3, and outer iteration count is 200. With a vehicle capacity of 15, all distribution units are divided into five distribution areas, and the corresponding routes are shown in Figure 13. The total cost is 1904.97.

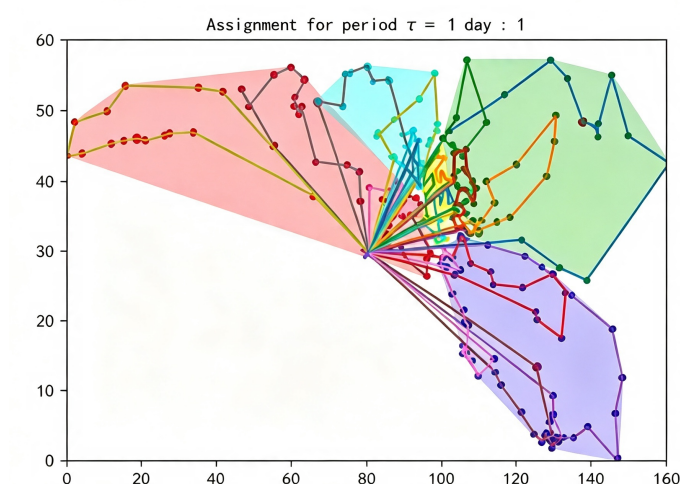


Figure 13. Route map for distribution areas.

5.5.3. Performance evaluation

To evaluate the performance of the proposed model and algorithm, the two-stage algorithm combining the K-means algorithm and the improved ALNS is used for the distribution unit division and route optimization of Y Courier Company. The total cost is 2069.27, and the final routes are presented in Figure 14. The proposed model and algorithm perform better than the results obtained by using the K-means algorithm alone for clustering.

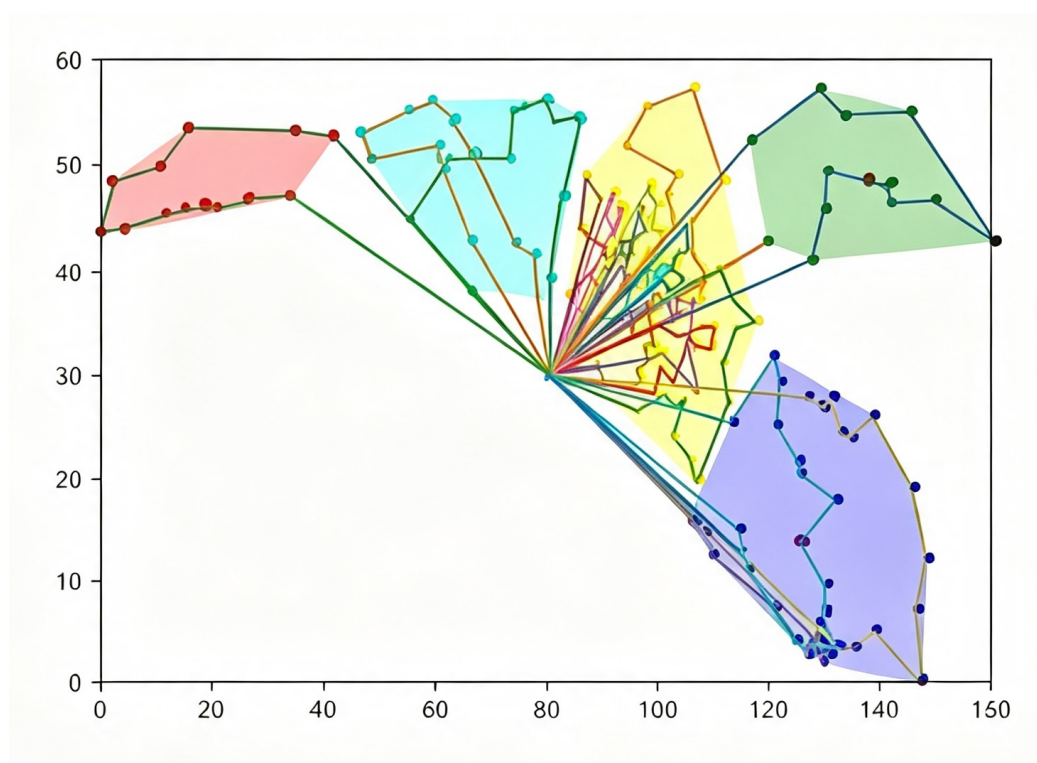


Figure 14. Final distribution route map of Y Courier Company.

6. Conclusion

This paper investigates a robust PDP that integrates courier learning-related effects and customer demand uncertainty into a two-stage optimization framework. The learning curve introduces nonlinearity, and the robust formulation adds further complexity. We address both through piecewise linearization and big-M dual reformulation. The resulting model consists of a first-stage MIQP for region partitioning and a second-stage mixed-integer linear program for routing. Since the problem is NP-hard, we design a hybrid solution framework: the tactical first-stage is solved to optimality by Gurobi, while an improved ALNS solves the operational second-stage routing problem.

Extensive experiments on standard CVRP benchmarks and a case study using real data from Y Logistics Courier Company in Xi'an validate the proposed methodology. Under identical time budgets, the improved ALNS finds better solutions with shorter travel distances than the benchmark heuristics, obtaining a 1.72% average gap to the BKSs on 13 standard CVRP benchmarks and the best or tied-best gap on 12 of them; an ablation study confirms that the embedded local search is the dominant

contributor. The joint optimization further reveals a balancing mechanism within the distribution network. The learning curve shortens service and travel times and produces compact, non-overlapping delivery territories. The robust uncertainty set, in turn, introduces a mild buffering cost under nominal conditions to protect the system from recourse penalties caused by courier overloads during demand surges. A 1,000-scenario Monte Carlo simulation shows that robust solutions improve reliability and cost stability under demand fluctuations, though the benefit is instance-dependent. The accompanying sensitivity analysis offers managers a practical tool for calibrating risk aversion against total cost.

Future research can expand in two key directions: (1) focus on dynamic modeling and forecasting of multiple real-world uncertainties such as traffic, weather and customer demand changes; and (2) explore robust strategies such as flexible route planning and dynamic scheduling to enhance distribution systems' adaptability and resilience.

Author contributions

Chi Zhang: Methodology; Writing – original draft; Xue Bai: Methodology; Writing – original draft; Funding acquisition; Guojun Sheng: Writing – review and editing; Funding acquisition; Zhili Zhou: Supervision; Funding acquisition.

Acknowledgements

This work was supported by the Key R&D Program of Shandong Province, China (No. 23-7-2-qljh-2-gx); the National Natural Science Foundation of China (NSFC) (72571205); the China Scholarship Council (202406560113), and the 2021 Annual Transportation Scientific Research Project of Shaanxi Provincial Department of Transportation (21-19R).

Use of generative-AI tools declaration

The authors used DeepSeek to improve the readability and language of the manuscript. All scientific content, methodology, analysis, and conclusions were developed and verified by the authors, who take full responsibility for the content of this article.

Conflict of interest

The authors declare that they have no competing interests.

Data availability statement

Data available on request from the authors.

References

1. H. Zhong, R. W. Hall, M. Dessouky, Territory planning and vehicle dispatching with driver learning, *Transp. Sci.*, **41** (2007), 74–89. <https://doi.org/10.1287/trsc.1060.0167>

2. Y. Li, C. Zhou, P. Yuan, T. T. A. Ngo, Experience-based territory planning and driver assignment with predicted demand and driver present condition, *Transp. Res. Part E*, **171** (2023), 103036. <https://doi.org/10.1016/j.tre.2023.103036>
3. I. Sungur, F. Ordóñez, M. Dessouky, A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty, *IIE Trans.*, **40** (2008), 509–523. <https://doi.org/10.1080/07408170701745378>
4. C. E. Gounaris, W. Wiesemann, C. A. Floudas, The robust capacitated vehicle routing problem under demand uncertainty, *Oper. Res.*, **61** (2013), 677–693. <https://doi.org/10.1287/opre.1120.1136>
5. P. Munari, A. Moreno, J. De La Vega, D. Alem, J. Gondzio, R. Morabito, The robust vehicle routing problem with time windows: Compact formulation and branch-price-and-cut method, *Transp. Sci.*, **53** (2019), 1043–1066. <https://doi.org/10.1287/trsc.2018.0886>
6. M. L. Fisher, R. Jaikumar, A generalized assignment heuristic for vehicle routing, *Networks*, **11** (1981), 109–124. <https://doi.org/10.1002/net.3230110205>
7. Y. Huang, M. Savelsbergh, L. Zhao, Designing logistics systems for home delivery in densely populated urban areas, *Transp. Res. Part B*, **115** (2018), 95–125. <https://doi.org/10.1016/j.trb.2018.07.006>
8. B. Zheng, F. Tang, H. Yang, Tobacco distribution based on improved K-means algorithm, in: Proceedings of the 2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, IEEE, 2009, 724–728.
9. Y. Wang, X. Ma, M. Xu, Y. Liu, Y. Wang, Two-echelon logistics distribution region partitioning problem based on a hybrid particle swarm optimization–genetic algorithm, *Expert Syst. Appl.*, **42** (2015), 5019–5031. <https://doi.org/10.1016/j.eswa.2015.02.058>
10. P. Matl, R. F. Hartl, T. Vidal, Workload equity in vehicle routing problems: A survey and analysis, *Transp. Sci.*, **52** (2018), 239–260. <https://doi.org/10.1287/trsc.2017.0744>
11. D. Haugland, S. C. Ho, G. Laporte, Designing delivery districts for the vehicle routing problem with stochastic demands, *Eur. J. Oper. Res.*, **180** (2007), 997–1010. <https://doi.org/10.1016/j.ejor.2005.11.070>
12. E. Khodabandeh, L. Bai, S. S. Heragu, G. W. Evans, T. Elrod, M. Shirkness, Modelling and solution of a large-scale vehicle routing problem at GE appliances & lighting, *Int. J. Prod. Res.*, **55** (2017), 1100–1116. <https://doi.org/10.1080/00207543.2016.1220685>
13. M. Schneider, The vehicle-routing problem with time windows and driver-specific times, *Eur. J. Oper. Res.*, **250** (2016), 101–119. <https://doi.org/10.1016/j.ejor.2015.09.015>
14. M. Bender, A. Meyer, J. Kalcsics, S. Nickel, The multi-period service territory design problem—an introduction, a model and a heuristic approach, *Transp. Res. Part E*, **96** (2016), 135–157. <https://doi.org/10.1016/j.tre.2016.09.007>
15. J. Beardwood, J. H. Halton, J. M. Hammersley, The shortest path through many points, *Proc. Cambridge Philos. Soc.*, **55** (1959), 299–327. <https://doi.org/10.1017/S0305004100034095>
16. D. M. Stein, An asymptotic, probabilistic analysis of a routing problem, *Math. Oper. Res.*, **3** (1978), 89–101. <https://doi.org/10.1287/moor.3.2.89>

17. A. Hindle, D. Worthington, Models to estimate average route lengths in different geographical environments, *J. Oper. Res. Soc.*, **55** (2004), 662–666. <https://doi.org/10.1057/palgrave.jors.2601751>
18. B. Çavdar, J. Sokol, A distribution-free TSP tour length estimation model for random graphs, *Eur. J. Oper. Res.*, **243** (2015), 588–598. <https://doi.org/10.1016/j.ejor.2014.12.020>
19. R. C. Larson, A. R. Odoni, Urban Operations Research, Prentice-Hall, Englewood Cliffs, 1981.
20. M. Webb, Cost functions in the location of depots for multiple-delivery journeys, *J. Oper. Res. Soc.*, **19** (1968), 311–320. <https://doi.org/10.1057/jors.1968.74>
21. C. F. Daganzo, The distance traveled to visit N points with a maximum of C stops per vehicle: An analytic model and an application, *Transp. Sci.*, **18** (1984), 331–350. <https://doi.org/10.1287/trsc.18.4.331>
22. M. A. Figliozzi, Planning approximations to the average length of vehicle routing problems with time window constraints, *Transp. Res. Part B*, **43** (2009), 438–447. <https://doi.org/10.1016/j.trb.2008.08.004>
23. I. Sungur, Y. Ren, F. Ordóñez, M. Dessouky, H. Zhong, A model and algorithm for the courier delivery problem with uncertainty, *Transp. Sci.*, **44** (2010), 193–205. <https://doi.org/10.1287/trsc.1090.0303>
24. T. P. Wright, Factors affecting the cost of airplanes, *J. Aeronaut. Sci.*, **3** (1936), 122–128. <https://doi.org/10.2514/8.155>
25. M. Y. Jaber, C. H. Glock, A learning curve for tasks with cognitive and motor elements, *Comput. Ind. Eng.*, **64** (2013), 866–871. <https://doi.org/10.1016/j.cie.2012.12.005>
26. J. R. De Jong, The effects of increasing skill on cycle time and its consequences for time standards, *Ergonomics*, **1** (1957), 51–60. <https://doi.org/10.1080/00140135708964571>
27. M. Kunkel, M. Schwind, Vehicle routing with driver learning for real world CEP problems, in: Proceedings of the 2012 45th Hawaii International Conference on System Sciences, IEEE, 2012, 1315–1322. <https://doi.org/10.1109/HICSS.2012.633>
28. M. Hewitt, A. Chacosky, S. E. Grasman, B. W. Thomas, Integer programming techniques for solving non-linear workforce planning models with learning, *Eur. J. Oper. Res.*, **242** (2015), 942–950. <https://doi.org/10.1016/j.ejor.2014.10.060>
29. X. Chen, B. W. Thomas, M. Hewitt, Multi-period technician scheduling with experience-based service times and stochastic customers, *Comput. Oper. Res.*, **82** (2017), 1–14. <https://doi.org/10.1016/j.cor.2016.12.026>
30. D. Lu, F. Gzara, The robust vehicle routing problem with time windows: Solution by branch and price and cut, *Eur. J. Oper. Res.*, **275** (2019), 925–938. <https://doi.org/10.1016/j.ejor.2018.12.019>
31. C. Lee, K. Lee, S. Park, Robust vehicle routing problem with deadlines and travel time/demand uncertainty, *J. Oper. Res. Soc.*, **63** (2012), 1294–1306. <https://doi.org/10.1057/jors.2011.136>
32. S. Souyris, C. E. Cortés, F. Ordóñez, A. Weintraub, A robust optimization approach to dispatching technicians under stochastic service times, *Optim. Lett.*, **7** (2013), 1549–1568. <https://doi.org/10.1007/s11590-012-0557-6>
33. E. Cao, M. Lai, H. Yang, Open vehicle routing problem with demand uncertainty and its robust strategies, *Expert Syst. Appl.*, **41** (2014), 3569–3575. <https://doi.org/10.1016/j.eswa.2013.11.004>

34. S. Braaten, O. Gjønnnes, L. M. Hvattum, G. Tirado, Heuristics for the robust vehicle routing problem with time windows, *Expert Syst. Appl.*, **77** (2017), 136–147. <https://doi.org/10.1016/j.eswa.2017.01.038>
35. Y. Adulyasak, P. Jaillet, Models and algorithms for stochastic and robust vehicle routing with deadlines, *Transp. Sci.*, **50** (2016), 608–626. <https://doi.org/10.1287/trsc.2014.0581>
36. J. Han, C. Lee, S. Park, A robust scenario approach for the vehicle routing problem with uncertain travel times, *Transp. Sci.*, **48** (2014), 373–390. <https://doi.org/10.1287/trsc.2013.0476>
37. V. Pureza, R. Morabito, M. Reimann, Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW, *Eur. J. Oper. Res.*, **218** (2012), 636–647. <https://doi.org/10.1016/j.ejor.2011.12.005>
38. J. De La Vega, P. Munari, R. Morabito, Robust optimization for the vehicle routing problem with multiple deliverymen, *Cent. Eur. J. Oper. Res.*, **27** (2019), 905–936. <https://doi.org/10.1007/s10100-017-0511-x>
39. K. Sim, E. Hart, A combined generative and selective hyper-heuristic for the vehicle routing problem, in: Proceedings of the Genetic and Evolutionary Computation Conference 2016, ACM, 2016, 1093–1100. <https://doi.org/10.1145/2908812.2908942>
40. C. Hu, J. Lu, X. Liu, G. Zhang, Robust vehicle routing problem with hard time windows under demand and travel time uncertainty, *Comput. Oper. Res.*, **94** (2018), 139–153. <https://doi.org/10.1016/j.cor.2018.02.006>
41. S. Qin, S. Liu, H. Kuang, Piecewise linear model for multiskilled workforce scheduling problems considering learning effect and project quality, *Math. Probl. Eng.*, **2016** (2016), 3728934. <https://doi.org/10.1155/2016/3728934>
42. N. Christofides, S. Eilon, An algorithm for the vehicle-dispatching problem, *J. Oper. Res. Soc.*, **20** (1969), 309–318. <https://doi.org/10.1057/jors.1969.75>



AIMS Press

©2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)