sustainability

MDPI

Article

Targeted Allocation and Caps for Food Delivery Promise Extensions Under Protective Operating Requirements

Zirui Xu [1,*], Mengchen Sun [2,*], Yujia Zhang [3], Peiyao Wu [2] and Pengpeng Hao [4]

[1] School of Emergency Management and Safety Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China
[2] College of Emergency Technology and Command, University of Emergency Management, Langfang 065201, China
[3] School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
[4] School of Management and Engineering, Capital University of Economics and Business, Beijing 100071, China
* Correspondence: cumtbxuzirui@163.com (Z.X.); smcfuture@outlook.com (M.S.)

Abstract

Food-delivery platforms promise a delivery time for each order, but several orders may share one rider and one route. When their deadlines conflict, a dispatcher must decide which promises to extend and by how much. We measure the smallest common extension that permits a pickup-and-delivery route, distribute a fixed allowance among the orders, and check the route against the revised promises. In a simulation of 13,772 cohorts from the Meituan–INFORMS records, each cohort received three order-minutes per order, with three minutes reserved before departure. Allocating minutes by individual order need reduced the share of cohorts with a certified residual conflict from 6.06% to 4.10% under a six-minute per-order cap. Using lateness on a joint route to set priorities reduced that share to 3.72%. A dispatcher can use the residual conflict to decide whether to change the visit sequence, redistribute the allowance, or review the budget and cap.

Keywords: food-delivery platforms; delivery–promise management; time–budget allocation; per-order caps; service performance; protective operating requirements

1. Introduction

Food-delivery platforms promise individual arrival times, yet concurrent orders share one rider's route and operating time. Orders that are feasible individually can become jointly infeasible when their pickups and deliveries compete for that time. Equal extensions can allocate minutes to orders with sufficient slack while leaving tighter tasks unresolved; concentrating the allowance can impose large promise changes on individual customers. Research connects food-delivery growth with service organization and examines the economic viability of platform business models [1,2]. We examine how a platform can distribute a finite promise-extension allowance across already-assigned orders while reserving specified operating time and limiting each customer's promise change.

A promised time sets the horizon within which a rider must complete the assigned work. Studies of delivery work relate time pressure and platform oversight to risky riding, burnout, route preferences, and perceived algorithmic management [3–6]. Heat exposure and monitoring can also change the conditions in which couriers work [7–10]. These findings motivate a practical design choice. A platform can reserve time for travel, service, or a break before it determines which delivery promises can be met.

Check for updates

Academic Editor: Lei Zhang

Received: 26 October 2026
Revised: 15 November 2026
Accepted: 24 November 2026
Published: 29 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Promise changes also affect the service offered to customers. Research on delay compensation identifies heterogeneous preferences for delivery-delay thresholds and compensation arrangements [11]. Last-mile studies distinguish stakeholder concerns and show that customer-preferred service models can conflict with broader sustainability objectives [12,13]. System safety treats operating constraints as design elements, and risk assessment connects model outputs to the decisions they inform [14,15]. We express the promise-management decision through three quantities: a total extension budget, the minutes assigned to each order, and a maximum extension per order. Together, they show how a feasible task plan distributes promise changes across customers.

Existing delivery optimization determines service capacity [16], routes, and responses to changing requests [17,18]. Meal-delivery models incorporate uncertain preparation times and deadlines [19]. Batching and assignment determine which orders travel together [20,21], while workforce scheduling determines rider availability [22]. Our decision follows acceptance and assignment: the orders and rider remain fixed, and the platform adjusts the existing promises. A complete current task set is termed a cohort. This fixed-workload comparison isolates how the placement of the same cohort allowance changes joint feasibility.

Technology can change the resources available for a delivery. A University of Calabria case study examines routes for drone meal delivery [23]. Reviews of last-mile drones identify efficiency, accessibility, and environmental outcomes [24] and the implementation priorities of senders, recipients, and regulators [25]. Drone resupply can also coordinate a rider's route with aerial replenishment [26]. Here, the rider and accepted orders are given. The available decision is how many minutes to add to each promise when the orders share a route.

The framework builds on established routing and allocation concepts. Time-window routing defines feasible visit sequences [27], while soft windows and time-window relaxation represent lateness and its cost [28,29]. Dynamic pickup-and-delivery research addresses information updates and request handling [30–32]. City logistics places these calculations within operating networks [33]. Recent analysis of the same Meituan release studies delivery delays and their propagation across stages [34]. We combine a common-extension calculation, bounded allocation rule, and exact small-instance resource benchmark to evaluate a specific promise-management decision: distributing a fixed allowance among current customer promises. Table 1 compares the decisions and resources addressed by these approaches.

Table 1. Decisions addressed by related delivery and routing models.

Research	Decision and Treatment of Time	Resource Open to Change
Meal delivery with uncertain readiness [19]	Choose dispatch and pickup routes under order deadlines	Route and dispatch timing
Soft time windows and relaxations [28,29]	Penalize or relax time-window violations during route search	Route sequence and search feasibility
Campus drone delivery [23] and drone resupply [26]	Plan aerial delivery routes or synchronize rider movement and resupply	Vehicle mix, routes and resupply schedule
Present study	Reassign a fixed number of promise minutes and test the shared route under a customer cap	Per-order promises and visit order within an assigned cohort

The open question is how to distribute a limited number of promise minutes after the rider and orders have been assigned. One order may have ample time when considered alone

but share a tight route with several others. A useful decision therefore needs three outputs: the order-level extensions, a bound on the largest customer change, and a route that meets the revised deadlines or quantifies the remaining conflict.

We connect these outputs in one fixed-workload comparison. The joint route check identifies which conflicts can be removed by reordering visits. Two priority rules then place the same allowance using either individual completion times or lateness on a common route. For small cohorts, exhaustive enumeration measures how much extension any route would need under the customer cap.

2. Materials and Methods: Decision Framework for Delivery–Promise Management

2.1. Current Orders, Promise Extensions, and Time Budgets

At decision time t, $J_w(t)$ contains all orders accepted by rider w but not yet delivered. Each order retains its recorded promise d_j and pickup status. An uncollected order must visit its pickup location before its delivery location. A state represents the rider's complete current task set, termed a closed cohort in policy comparisons. All alternatives retain the same orders. The decision variable b_j is the extension assigned to order j, giving the revised promise $d'_j = d_j + b_j$. Thus, $b_j = 3$ min changes a promise of 12:10 to 12:13.

Measure the minimum common extension G on the current route problem. Allocate the cohort allowance among its orders subject to the individual cap. Recheck the route bounds under the revised promises. Figure 1 illustrates the three decisions for a six-order cohort.

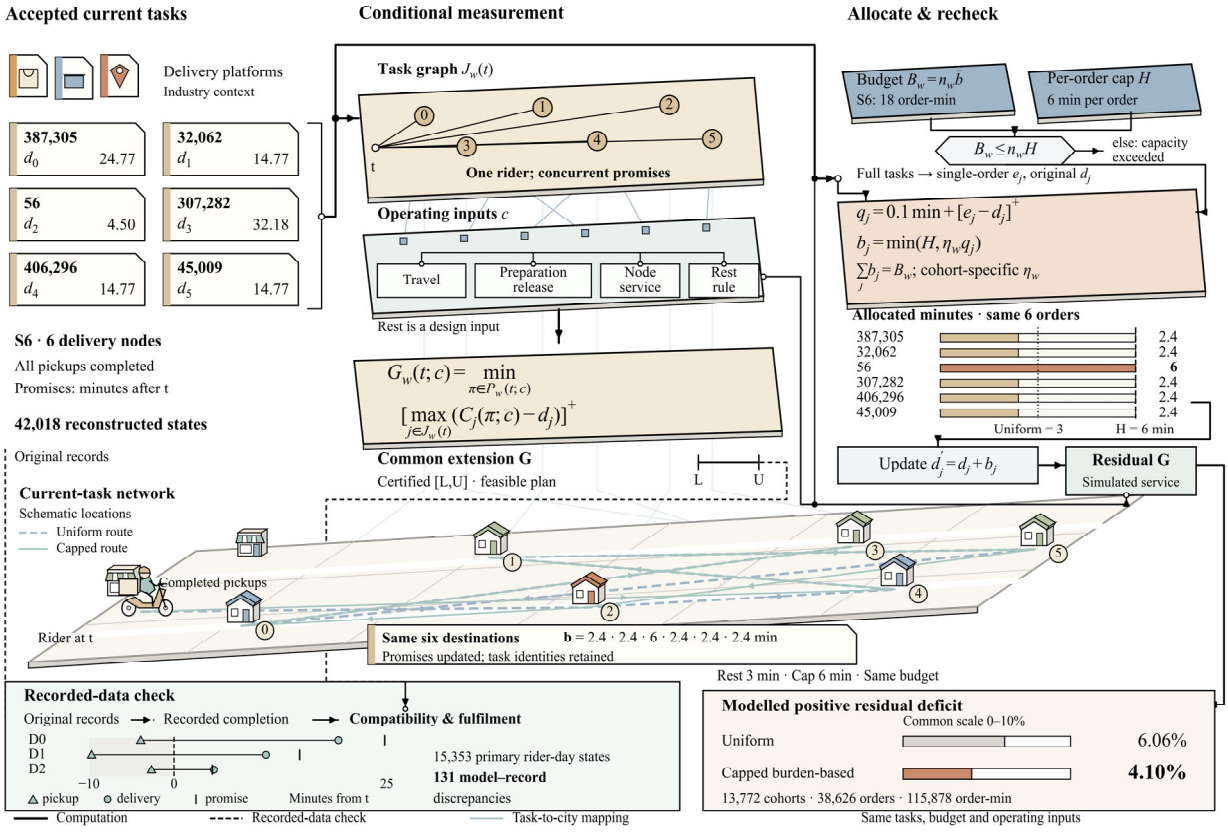


Figure 1. Computational links between full-task measurement and capped cohort allocation. Six orders illustrate the calculation. Dashed blue and solid teal lines show simulated uniform and capped routes on a schematic map. The bars allocate 18 order-minutes, with 3 min per order under uniform allocation and a 6-min cap. Dashed connectors lead to recorded outcomes. The final percentages refer to all 13,772 policy cohorts at a 3-min initial rest and a positive lower bound above 0.02 min.

The mean allowance $\bar{b}$ sets the cohort budget B_w, and H caps each order's extension. The budget is measured in order-minutes: extending four promises by 3 min each uses 12 order-minutes. Order-minutes account for changes in customer commitments; initial rest r and route duration are measured separately. Research on workload equity and allocation distinguishes resource totals, distributions, and individual limits [35–37]. Here, $\bar{b}$ controls the total relaxation of promises, H limits its concentration on any order, and b_j specifies each customer's change. Holding the cohort budget fixed makes the comparison a test of how effectively the same allowance is distributed.

2.2. Joint Task Completion and the Minimum Common Extension

A route is a sequence of remaining pickup-and-delivery nodes, $\pi = (v_1, \ldots, v_m)$. Travel time is τ, the earliest service time is a_v, and pickup or handover service takes s_v. Service starts when both the rider and the order are available. Setting $T_0 = r$ reserves r minutes of initial rest before travel begins. Node completion times follow:

$$T_0 = r, \qquad T_k = \max\{T_{k-1} + \tau_{v_{k-1},v_k}, a_{v_k}\} + s_{v_k} \tag{1}$$

The recursion accounts for travel, waiting for food preparation, and pickup or delivery service. A collected order retains only its delivery node; an uncollected order retains both nodes and their precedence. Reference conditions use 15 km/h travel, 0.5 min of service per node, and the released preparation estimates. Distances use projected relative coordinates. General pickup-and-delivery models define these task relations [38], while insertion heuristics and neighborhood search provide established route-construction methods [39,40]. Comparing the resulting completion times with the recorded promises determines whether the full task set can meet them together.

Let $P_w(t;c)$ contain all routes completing the current tasks while satisfying pickup precedence and operating conditions c. A route's largest positive lateness is the common extension needed to make its promises feasible. Minimizing this quantity over the feasible route set defines G:

$$G_w(t;c) = \min_{\pi \in P_w(t;c)} \left[\max_{j \in J_w(t)} \{C_j(\pi;c) - d_j\} \right]^+. \tag{2}$$

Here, $[x]^+ = \max(x,0)$. G is the smallest common shift of all current promises that permits joint completion under the specified conditions. For example, $G = 4$ min means that a route becomes feasible when every promise receives 4 min; for four orders, this uses 16 order-minutes. The objective follows established time-window and soft-window formulations [27–29]. Its role here is to express a shared-route conflict in the same minute unit as a promise adjustment. $G = 0$ identifies a plan meeting the recorded promises; positive G quantifies the common allowance still required. Nonuniform allocation then tests whether directing a fixed budget to different orders improves joint feasibility.

For a trial common shift g, the subset dynamic program stores the earliest completion time for each visited set S and final node v [41,42]. A delivery can enter the set once its pickup has been visited or was already complete at the snapshot. The update adds travel time, waits until release if necessary, adds service, and rejects a completion after the shifted deadline. An earlier completion can wait and follow any continuation available to a later one. An individual-order lower bound and three feasible route heuristics bracket G. The solver tests the lower bound and bisects the remaining interval until its width is at most 0.02 min. For m nodes, one feasibility test takes $O(m^2 2^m)$ time and $O(m2^m)$ memory. Total time scales with the number of trial shifts.

$$A = \max_{j \in J_w(t)} \left[e_j - d_j \right]^+, \qquad A \le L \le G \le U. \tag{3}$$

The quantity e_j is the earliest completion of order j considered separately under the same conditions. Resolved G values are displayed as $(L + U)/2$, but classification uses the bounds: $L > \varepsilon$ is certified positive, $U \leq \varepsilon$ is feasible within tolerance, and $L \leq \varepsilon < U$ is undetermined. The tolerance $\varepsilon = 0.02$ min (1.2 s) governs classification; exact zeros are retained separately. The designated solution limit is 14 remaining nodes, and unconverged problems retain their intervals. Vehicle routing admits many objectives and algorithms [43,44]. Keeping the common-extension objective fixed makes the requirement before and after promise adjustment directly comparable.

An interval of $[2.98, 3.00]$ min certifies that at least about 2.98 min of common extension is needed and that a route exists with 3.00 min. Revised promises are checked against the feasible upper bound and its route. Interval separation determines whether differences between alternatives exceed numerical tolerance, so near-threshold cases remain explicit in the decision.

2.3. Cohort Budgets and Capped Order-Level Allocation

For n_w orders, the budget is $B_w = n_w\bar{b}$; uniform allocation assigns $\bar{b}$ to each order. The burden-based rule compares each individual earliest completion e_j with its recorded promise d_j. Its priority q_j adds the positive individual deficit to a 0.1-min floor, giving larger weights to orders with greater individual time shortfalls. The term burden denotes this modelled task-time requirement. Joint G subsequently checks the complete allocation, including interactions between orders that the individual priority does not capture. Priority construction and joint feasibility assessment therefore serve distinct functions:

$$B_w = n_w\bar{b}, \qquad q_j = 0.1 \text{ min} + \left[e_j - d_j\right]^+. \tag{4}$$

The algorithm distributes the remaining budget in proportion to the active priorities. In each pass, it fixes every provisional share above the cap (numerical allowance 10^{-10} min), subtracts the fixed amounts and recomputes the common multiplier for the remaining orders [45,46]. Simultaneous cap hits are handled in the same pass. At least one order leaves the active set in each capping pass. A zero budget assigns zero throughout. At total capacity, every order receives the cap. An over-capacity request is flagged before route evaluation.

$$b_j = \min\{H, \eta_w q_j\}, \qquad \sum_{j \in J_w(t)} b_j = B_w, \qquad 0 \leq b_j \leq H. \tag{5}$$

Full allocation requires $B_w \leq n_w H$. A larger budget exceeds the combined extension capacity of the orders and is reported as a capacity shortfall. Each cohort has its own η_w; when all orders reach H, the smallest saturating value is recorded. These checks enforce both individual caps and conservation of the cohort budget. Revised promises and the residual common extension are:

$$d_j' = d_j + b_j, \qquad G_w^{\mathbf{b}} = G_w(t; c; \mathbf{d} + \mathbf{b}) \tag{6}$$

All policy comparisons retain the same orders, rider assignments, operating parameters, and initial rest. A uniform extension of b minutes per order satisfies:

$$G_w(t; c; \mathbf{d} + b\mathbf{1}) = [G_w(t; c; \mathbf{d}) - b]^+. \tag{7}$$

A common deadline shift reduces each route's maximum positive lateness by the same amount, truncated at zero. This identity checks consistency between the two solves. Nonuniform extensions alter relative deadlines and require re-solving. Figure 1 separates joint G measurement from the calculation of e_j and q_j; both use the full task input. The budget B_w, cap H, and priorities q_j determine b_j, after which the revised promises return to the joint checker. The six-order example allocates 18 order-minutes. Solid lines denote computation, while subsequent records enter a separate fulfilment-assessment branch. This separation makes each promise change traceable from its input requirement to its cohort outcome.

2.4. Minimum-Total-Extension Benchmark and Service Criterion

For small task sets, enumerating every legal pickup-and-delivery sequence establishes whether the budget can support joint feasibility. On route π, order j needs at least $h_j(\pi) = [C_j(\pi;c) - d_j]^+$ minutes. We retain routes whose individual requirements do not exceed H and minimize their sum. The resulting minimum total extension is B^*:

$$B^*(H) = \min_{\pi \in P_w(t;c):\, h_j(\pi) \le H\ (\forall j)} \sum_{j \in J_w(t)} h_j(\pi). \tag{8}$$

B^* measures the least total investment of order-minutes needed by the task set; G measures its least common deadline shift. B^* therefore tests budget sufficiency, while G measures the residual requirement before and after allocation. With a fixed route set and completion times, and an attained minimum, a fully used budget can satisfy all tasks when:

$$B^*(H) \le B_w \le n_w H. \tag{9}$$

If $B_w < B^*$, the budget is insufficient. If $B_w > n_w H$, it exceeds the orders' combined capacity. Between these bounds, assign the minimum extensions on the benchmark route and distribute $B_w - B^*$ within the remaining capacity. This construction establishes the resource-feasibility condition for the stated model and distinguishes a shortage from a failure of the allocation rule. Numerical experiments also use $h_j^{\varepsilon} = [C_j - d_j - \varepsilon]^+$ for the tolerance-matched benchmark. Strict-zero and ε-tolerance results are evaluated separately so that resource sufficiency and feasibility use the same criterion. The benchmark gives managers a diagnostic for deciding whether additional allowance is necessary.

For the tolerance-based exact comparison, enumerate the legal routes. On each route, sum the positive excess of each completion over its recorded deadline plus 0.02 min. Discard routes requiring any individual extension above the cap. The smallest remaining sum is the exact total extension at this tolerance.

For each allocation, we run the feasibility dynamic program at its own upper bound U and select the legal route with the earliest final completion [47]. Strict improvements replace a stored label. Equal times retain the first predecessor in ascending subset and node order, and the first minimum endpoint is selected. This secondary criterion applies to every rule. Simulated lateness is scored against both the recorded customer promise and the revised promise.

2.5. Recorded Fulfilment and Model Compatibility

Subsequently, recorded arrival times A_j define fulfilment. $Y_w = 1$ when at least one current order arrives after its recorded promise, and ${Y_w}^{\max}$ is the largest nonnegative lateness among those orders. Both outcomes retain the decision-time task set and exclude later orders. Recorded task fields enter the requirement calculation; subsequent arrivals construct the outcomes.

$$Y_w = \mathbb{I}\{\exists j \in J_w(t) : A_j > d_j\}, \qquad Y_w^{\max} = \max_j \left[A_j - d_j\right]^+. \tag{10}$$

When actual execution is represented by the inputs and operating conditions in $P_w(t;c)$, G cannot exceed its maximum positive lateness. States satisfying $L > {Y_w}^{\max} + \varepsilon$ therefore enter a model–record compatibility review of coordinates, preparation estimates and service counts. Verification and validation distinguish computational implementation, input applicability and intended use [48,49]. Accordingly, we combine independent enumeration, recorded-execution comparisons and operating-condition changes, interpreting outputs against the observations and stated conditions [50]. The review identifies which task inputs need examination before their calculated deficits guide promise changes.

3. Data, Experimental Design, and Statistical Analysis

3.1. Public Records and Analysis Cohorts

Operational records come from the four tables and documentation of the official Meituan–INFORMS TSL Research Challenge [51]. Dates and time-of-day variables use Asia/Shanghai; timestamps are converted to minutes relative to the decision time. Anonymous order and rider keys link accepted, unfinished tasks. Checks cover acceptance–pickup–delivery chronology and current-task consistency. The release contains 654,343 waybill rows, 568,546 unique accepted orders, 4955 rider identifiers, and 23 anonymous zones. We use three selected lunch-peak snapshots per day from 17 to 24 October 2022.

Consistency checks retain 42,018 task states from 3822 riders. The fulfilment sample requires a resolved reference problem and unexpired current promises. Selecting the earliest eligible state per rider-day gives 15,353 states from 3818 riders. The policy sample uses the first common daily snapshot and at most 14 remaining nodes, yielding 13,772 cohorts, 3691 riders, and 38,626 orders. These branches assess subsequent lateness and extension allocation, respectively. Retained order, state, rider, and cohort identifiers keep each outcome linked to its analytical unit and denominator.

The challenge dictionary defines estimate_arrived_time as the time promised to the customer (Table 2 of the Meituan data descriptor [51], p. 6). We use the value attached to the released order record. The snapshot lists the rider's unfinished orders, and the order table supplies one promise value per accepted order. A live implementation would require timestamped promise and preparation-forecast versions. Relative longitude and latitude use 111.32 $\cos(40°)$ and 110.57 km per degree. Travel speed (15 km/h), service (0.5 min per node), and initial rest (3 min in the policy comparison) are scenario inputs. The provider supplies the preparation forecast and the subsequent pickup and arrival timestamps. Studies of future requests, time slots, and scheduling address complementary service decisions [52–54]. Table 2 defines the notation, and Table 3 summarizes the data, computational settings, and analysis populations.

Table 2. Main notation and its engineering meaning.

Symbol	Definition and Unit	Engineering Object or Operation
$J_w(t)$, n_w	Current order set and order count	All accepted, unfinished orders held by one rider at time t.
d_j, C_j	Original promise and planned completion time (min)	Compare each customer's deadline with its route-based arrival time.
c, r	Operating conditions; initial rest (min)	State travel, waiting, node service, and initial-rest requirements.
G, L, U	Minimum common extension and bounds (min)	Shift all promises together; retain the solution interval $[L,U]$.
e_j, q_j	Individual earliest completion and allocation priority (min)	Weight extensions by individual task completion requirements.
$\bar{b}$, B_w	Mean allowance (min/order); total budget (order-min)	Set the total promise allowance for the current cohort.
H, b_j, η_w	Per-order cap and extension (min); cohort scaling factor	Limit the longest extension and conserve the cohort budget.
B^*, ε	Minimum total extension (order-min); numerical tolerance (min)	Check resource sufficiency and compare feasibility at matching precision.

Table 3. Data coverage, computational settings and analytical uses.

Item	Fixed Setting	Analytical Use
Operational records	17–24 October 2022; 23 anonymous zones; three lunch-peak snapshots per day	Current-task reconstruction and date-based partitioning.
Consistent states	42,018 states; 3822 riders	Joint requirements, input checks and numerical verification.
Primary fulfilment sample	15,353 rider-day states; 3818 riders	Subsequent fulfilment, compatibility and prediction.
Policy cohorts	13,772 cohorts; 38,626 unique orders	Allocation comparison on the same orders and budgets.
Reference conditions	15 km/h; 0.5 min/node; released preparation estimates	Conditional feasibility calculation.
Policy operating conditions	3-min initial rest; other reference settings retained	Allocate promises while reserving the specified initial time.
Budget and cap	$\bar{b} = 1, 3, 5$ min/order; $H = 3, 6, \infty$ min	Three budgets × three caps × two rules give 18 requests. Two exceed capacity, leaving 16 implementable combinations.
Numerical and exact checks	$\varepsilon = 0.02$ min; 48 small instances; 400 translation checks	Classification, resource benchmarking and code verification.
Independent behavioral context	423 public-survey respondents	Perceived algorithmic management and timeliness-related self-reports.

The first recorded epoch is near 11:25 local time, followed by the released epochs two and five minutes later. Using one common epoch per day gives each order a single allocation. Supplementary Table S1 compares workload, time to the tightest promise, and joint bounds between the resulting populations.

3.2. Policy Comparisons, Operating Settings and Exact Instances

Uniform and burden-based allocation share mean budgets $\bar{b} = 1$, 3, and 5 min per order and caps $H = 3$ min, 6 min, and unbounded. Of 18 requested combinations, the two with $\bar{b} = 5$ and $H = 3$ exceed capacity. For the remaining 16 combinations, we retain order extensions, budgets, residual G, and service outcomes. Uniform allocation is a constructed equal-extension reference policy. We evaluate both policies computationally on the recorded tasks and use observed arrivals for a separate fulfilment check. The main comparison uses $\bar{b} = 3$ and $H = 6$ to isolate the effect of targeting at equal budget. The uncapped comparison isolates the restriction imposed by H, while $H = \bar{b} = 3$ forces uniform allocation. Each cohort is checked before aggregation.

We vary the longitude projection, speed, service duration, preparation waiting, shared-stop service, and initial rest. Each scenario recomputes individual completion times, priorities, allocations, and joint route bounds on the same cohorts. For shared-stop service, consecutive pickups at one merchant incur one service duration after the latest applicable release. A return visit incurs service again. A separate comparison restricts all three policies to cohorts with recorded nonzero pending-pickup forecasts. The latitude-only calculation remains a relaxed diagnostic for the historical fulfilment comparison (Supplementary Figure S3).

Exact benchmarks use policy states with two to four orders and at most eight remaining nodes, stratified by date, order count, and positive-deficit status with 3 min of initial rest. Selecting the smallest state ID in each stratum gives 48 instances. All legal pickup-and-

delivery sequences are enumerated under strict-zero and ε-tolerance criteria. Separately, 100 small states are recomputed after common extensions of 0, 1, 3, and 5 min at 0.005-min precision, giving 400 translation checks. Preparation organization and batching affect meal-delivery routing [55]. Holding tasks fixed while changing one input at a time isolates its contribution to the computed requirement.

A slack-ranked rule allocates minutes first to orders with the smallest difference between the recorded deadline and earliest individual completion, dividing the available share equally among ties. A joint-route rule uses a common route computed before any extension, selecting the earliest finish at its feasible upper bound. Its weights equal 0.1 min plus each order's positive lateness on that route. Both rules use the same cohort budget and per-order cap. For every proposed allocation, the route check again allows the visit order to change.

3.3. Fulfilment Prediction and Statistical Uncertainty

Date-based partitioning assigns 9621 states from 17 to 21 October to training and 1971 states from 22 October to parameter selection. The historical test contains 3761 states from 2648 riders on 23–24 October. Evaluation on later dates preserves the temporal structure of the records [56], while separate selection and evaluation address model-selection bias [57]. Parameters are chosen on the validation date, and test predictions remain fixed. The partitions evaluate temporal screening performance within the same operational release. The two test dates had already been inspected during earlier analyses. The supplementary comparison protocol was recorded after the original analyses reproduced the historical results.

Base features are current order count, pending pickup count, minimum and mean remaining promise time, mean elapsed time, maximum node radius, prebook share, and sine and cosine encodings of decision time. Linear logistic regression standardizes inputs and uses $C = 1$; its augmented version adds $\log(1 + G/\text{min})$. Spline logistic regression uses cubic splines, three prespecified interactions, and four training-quantile knots for six continuous inputs. Both spline versions select $C = 10$ from 0.1, 1, and 10. Boosted trees use 200 iterations, a 0.05 learning rate, an L2 penalty of 1, at least 30 samples per leaf, and no early stopping. Both versions select 15 maximum leaf nodes from 7, 15, and 31. The methods and software follow Friedman [58] and Pedregosa et al. [59]. All 17 candidate fits and nine selected models are retained.

AUC assesses whether late states rank above on-time states [60]. Average precision (AP) summarizes precision and recall when higher-scored states receive review priority [61,62]. Brier score and log loss assess probability error; lower values indicate closer agreement with outcomes [63,64]. Calibration compares predicted and observed late-state shares within each model's probability deciles [65,66]. Together, the measures assess both case ranking and the accuracy of the reported probabilities. Analyses used Python 3.13.5, NumPy 2.1.3, pandas 2.2.3, Numba 0.61.0, and scikit-learn 1.6.1. Figures were generated with Matplotlib 3.10.0. End-to-end timing used one serial process on an AMD Ryzen 7 9700X processor (Advanced Micro Devices, Inc., Santa Clara, CA, USA).

For the main contrasts, we drew riders with replacement 1000 times (seed 20261111), including all of each selected rider's observations with the sampled multiplicity [67,68]. Each paired policy and decision threshold uses the same draws. Fixed predictions are compared within a separate rider resample of the historical test cohort. Percentile intervals are pointwise. Budget settings, zones, dates, and prediction increments are exploratory comparisons without familywise adjustment. Groups with fewer than two riders contribute a point estimate. AUC requires both outcome classes, average precision requires a positive

case, and Brier score and log loss use either class. Supplementary Tables S3 and S4 give the main estimates and resampling counts.

P95 and P99 describe extension levels below which approximately 95% and 99% of orders fall; the maximum checks the most extreme individual promise change. The median, interquartile range and P5–P95 summarize the center, central 50% and central 90% of the distribution. These quantities describe variation across orders or cohorts, whereas a 95% confidence interval for a mean describes estimation uncertainty. Joint plots and marginals retain the same samples. We record display transformations, numerical criteria, source mappings, and hashes for data, selection, computation, and plotting to support reproducibility [69–71].

4. Results and Operational Implications

4.1. Joint Task Checks Locate Extension Needs and Routing Opportunities

Joint checks distinguish a genuine need for promise extension under the model from a poor visit sequence. Among 42,018 states, 2803 have lower bounds above 0.02 min (6.67%). The solver reaches the required precision for 41,973 states; 45 retain wider intervals. Figure 2a illustrates the joint requirement in state 238, with four orders and one pending pickup. Circles denote individual earliest completion, triangles joint-route completion, and black marks recorded promises. Every order is feasible in isolation, but the joint route P2 → D1 → D2 → D3 → D4 requires about 0.673 min (40 s) of common extension. Checking the complete queue therefore detects a conflict that individual-order slack misses.

Figure 2b separates 39,143 exact-zero values from 2830 resolved nonzero values, including cases near the tolerance threshold. The separate strip retains interval widths for the 45 unresolved states. The numerical status guides the response: retain promises when a route is feasible within tolerance, review additional time when the lower bound exceeds tolerance, and continue solving when the interval remains undetermined. Retaining numerical status with G prevents unresolved calculations from being treated as definite shortfalls.

Figure 2c relates the joint requirement G to minimum remaining promise time. Negative horizontal values indicate an expired promise; cells and marginals show sample counts. At 0–5 min of remaining time, positive-deficit shares are about 35%, 35% and 41% for one order, two to three orders and at least four orders, respectively. Near-deadline cohorts therefore warrant early review. Order count and pickup status can prioritize checks, while joint solving determines each selected task set's extension requirement.

Changing the declared operating inputs changes the joint requirement for the same tasks (Figure 2d). The relaxed setting retains 1564 certified-positive states; among reference-positive states, 1234 become exactly zero and five become nonzero but within tolerance. Figure 2e distinguishes two reasons to review a promise decision. In 952 states, the certified joint lower bound exceeds the individual analytic bound, showing where isolated-order checks understate the requirement. In 474 states, an inexpensive route requires an extension, but optimization finds a plan within tolerance; 473 become exact zero. These cases warrant a route change before a promise extension. The full nonzero tails contain 975 joint-minus-individual values and 1487 heuristic-minus-joint values; the certification criteria select the two reported subsets.

Figure 2f retains the separate populations used for these decisions. Positive counts are 2803/42,018 for all states, 2801/41,973 for resolved states, 1824/40,996 for resolved states with unexpired promises, 628/15,353 for earliest eligible rider-day states, and 1564/42,018 under relaxed conditions. Horizontal intervals are rider-cluster estimates. Operationally, individual bounds and inexpensive routes can first bracket the requirement, with full-cohort solving used to resolve uncertain cases. This sequence separates verified joint shortfalls from extension requirements caused by an inefficient initial route.

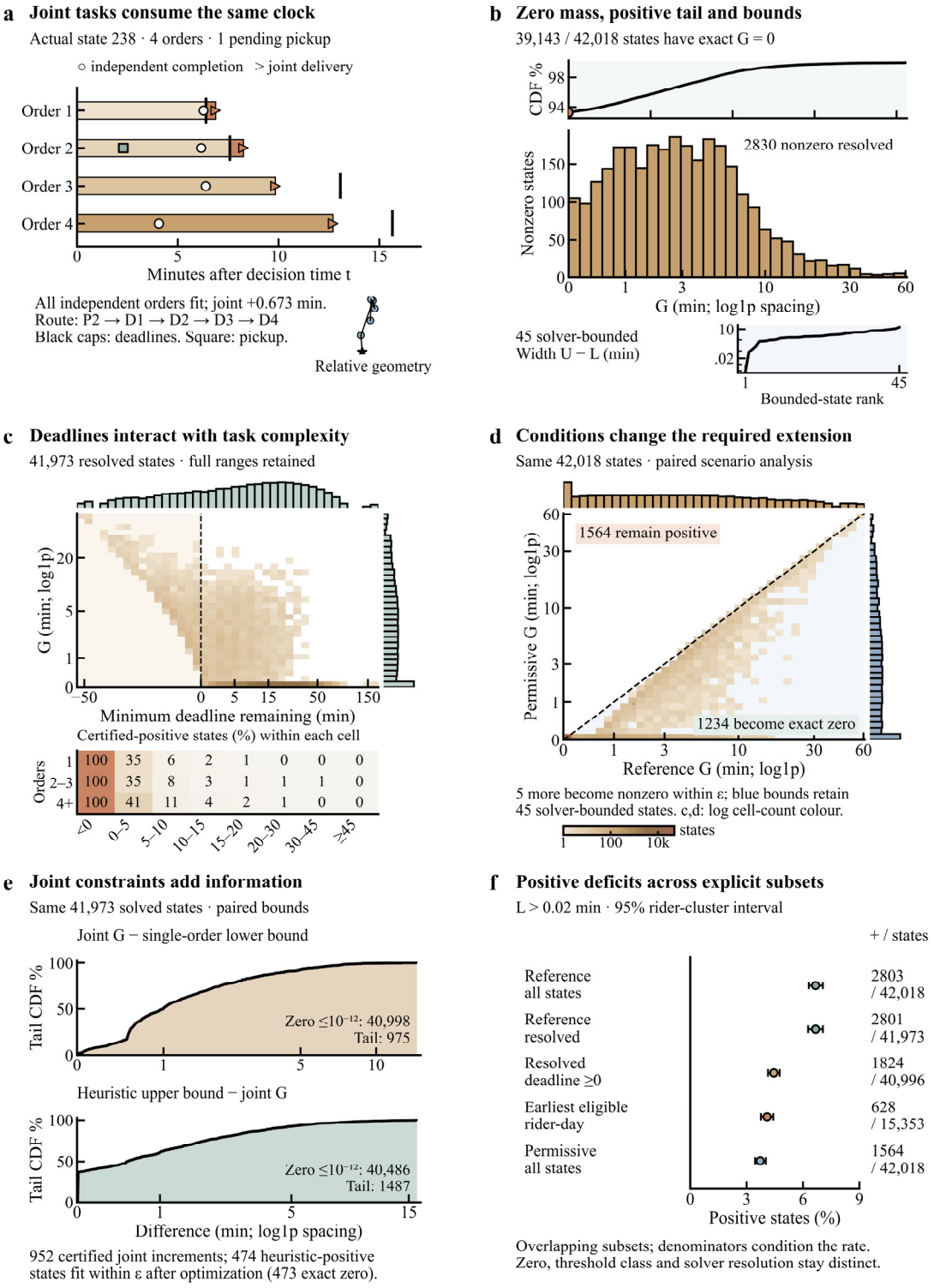


Figure 2. Joint time requirements and conditional feasibility of current orders. (**a**) Individual and joint completions relative to deadlines for one four-order state. (**b**) Zero and positive requirements among 42,018 states. (**c**) Counts by remaining time and joint deficit. (**d**) Reference and relaxed inputs for paired states. (**e**) Joint deficit minus the individual-order lower bound (upper strip) and the heuristic upper bound minus joint deficit (lower strip). (**f**) Certified-positive shares by analysis population. Continuous values are interval midpoints in minutes. Zero is shown separately from the positive tail.

4.2. Recorded Fulfilment Provides Task-Level Checks on Time Requirements

Subsequent recorded fulfilment provides a task-level check on the calculated requirements. All 15,353 primary rider-day states precede their current deadlines. Figure 3a pairs

resolved G with maximum recorded nonnegative lateness, retaining 14,714 zero-G and 9966 zero-lateness observations. Circles identify 131 states whose lower bounds exceed recorded lateness by more than 0.02 min. Among certified-positive states, 46 subsequently complete every order on time. These discrepancies provide a task-level review list for preparation estimates, node service, and geometry before the calculated deficits are used to revise promises.

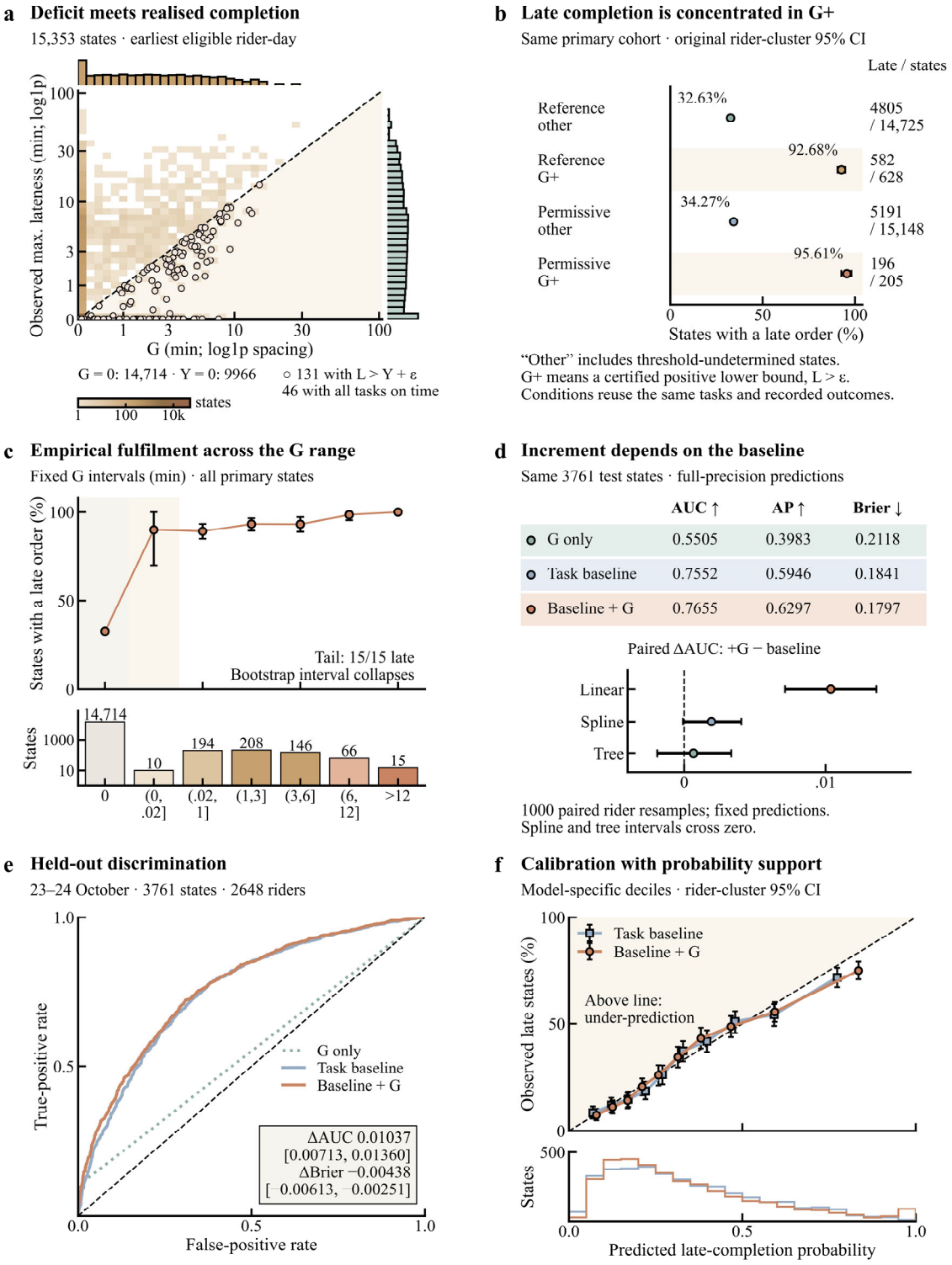


Figure 3. Conditional time deficits, subsequent fulfilment and predictive increments. (**a**) Joint deficit and maximum recorded lateness for 15,353 rider-days. Circles mark 131 discrepancies. (**b**,**c**) Observed

late-state proportions by deficit class and magnitude. (**d**) AUC, average precision and Brier score for the linear comparison, with paired AUC increments for the linear, spline and tree baselines. (**e**) Test-set receiver operating characteristic curves with the chance-discrimination diagonal. (**f**) Calibration by model-specific probability decile, with prediction counts below and the diagonal indicating perfect calibration. Whiskers on proportions are rider-bootstrap intervals.

In Figure 3b, 582 of 628 reference-positive states subsequently contain a late order (92.68%), compared with 4805 of 14,725 other states (32.63%). Under relaxed conditions, the corresponding counts are 196/205 and 5191/15,148 (95.61% and 34.27%). Rider-cluster 95% intervals and group counts show the support for these comparisons. Certified-positive G identifies a group with frequent subsequent lateness. Late states elsewhere in the sample support combining the joint requirement with order count, remaining time, and waiting information for screening.

Figure 3c separates zero, 0–0.02 min, and successive positive intervals, containing 14,714, 10, 194, 208, 146, 66, and 15 states. Upper points show subsequent late-state shares; lower counts show their support. All 15 states above 12 min contain a late order, and their empirical resampling interval collapses to 100%. These large-requirement task sets warrant priority review, although the extreme observed share is based on 15 states.

On the two historical test dates, adding G to the linear model raised AUC from 0.7552 to 0.7655 and average precision from 0.5946 to 0.6297. The paired AUC increment was 0.01037 (new rider-bootstrap interval 0.00703 to 0.01358). The spline and boosted-tree increments were 0.00192 and 0.00066, with intervals spanning zero (Supplementary Table S4). Figure 3d–f shows the fixed predictions and historical intervals. These comparisons address within-release screening, while the allocation experiment uses G in minutes to check the shared route [72].

Supplementary Figures S1–S3 compare the numerical bounds, input assumptions, and recorded execution. Supplementary Figures S4 and S5 show the internal prediction increments for the fixed test dates.

Of 57,440 pending-pickup references, 23,922 had a positive preparation timestamp at or before the decision, and 813 had the provider's zero code for an absent timestamp. Both yield a zero wait from the snapshot. A further 13,177 forecasts fell after the subsequent recorded pickup. Of 131 model–record discrepancies, 127 also occurred in the policy sample. They account for 0.85% of 15,353 fulfilment states and 20.86% of the 628 certified-positive states. The model comparisons retain these cohorts (Supplementary Figure S3).

4.3. Targeted Allocation Reduces Task Conflicts Under the Same Budget

The main comparison isolates the allocation of minutes by holding resources and operating conditions fixed. Both strategies retain 13,772 cohorts, 38,626 orders, and 115,878 order-minutes, with 3 min of initial rest and a 6-min cap. In Figure 4a, uniform allocation assigns 3 min to each of six orders. The capped rule assigns 2.4 min to five orders and 6 min to the highest-priority order; both use 18 order-minutes. The circuit links completion requirements, priorities, cohort scaling, and cap enforcement. Allowance moves from less time-constrained orders to a tighter order within the same queue, after which the joint requirement is checked again.

Figure 4b relates priority q to the assigned extension, with reference lines at 3 min and 6 min. A total of 792 orders reach the cap. Equal priorities can receive different extensions because companion orders and cohort budgets determine the scaling factor. Allocation must therefore use the complete current cohort rather than an order's priority alone. Recomputing residual G then identifies cohorts that still require additional time after capped allocation.

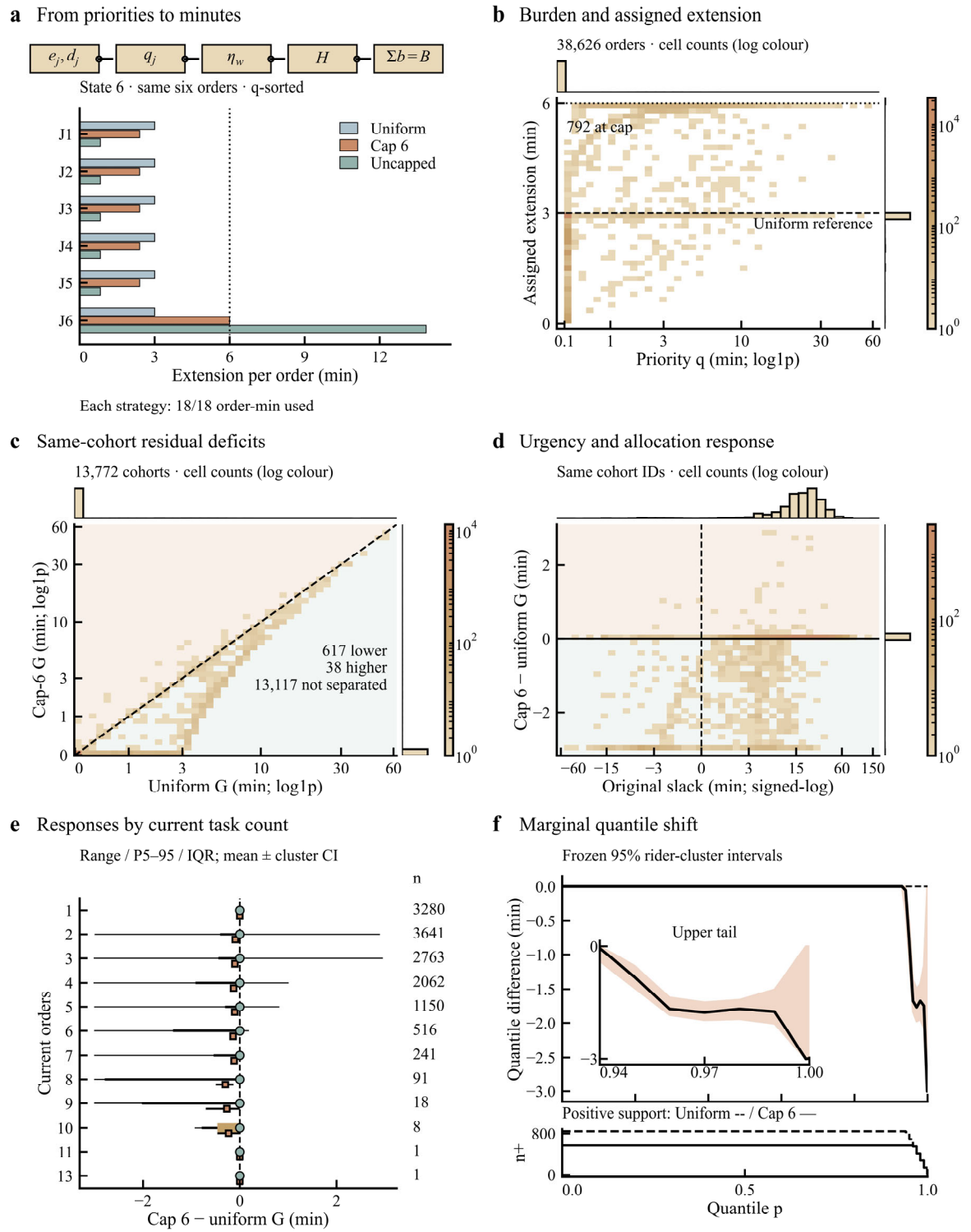


Figure 4. Order-level allocation and cohort responses under an equal budget. (**a**) Six-order example at a budget of 18 order-minutes. (**b**) Priorities and extensions for 38,626 orders. Lines mark 3 and 6 min. (**c**) Paired residual deficits under uniform and capped allocation for 13,772 cohorts. (**d**,**e**) Paired changes by tightest deadline and order count. Strips mark the interquartile and 5th–95th percentile ranges. Groups of 11 and 13 orders each contain one cohort. (**f**) Differences at matched quantiles. Logarithmic or signed-log axes retain zeros and show values in minutes.

At the same budget, 835 cohorts had a certified residual conflict under uniform allocation and 565 under capped individual targeting (6.06% and 4.10%). The paired difference was −1.96 percentage points (95% rider-cluster interval −2.19 to −1.74), or 270 fewer

positive cohorts. This is a 32.3% reduction relative to the 835 uniform-positive cohorts. Comparing the bounds for each cohort, 617 deficits decreased, and 38 increased. Of the 13,117 remaining pairs, 12,557 had identical allocations, 552 more were within threshold under both rules, and eight had overlapping or insufficiently separated bounds.

Figure 4d relates each cohort's tightest recorded promise to its change in G; Figure 4e groups the same paired changes by order count. The central 50% strip, P5–P95 span, and full range retain common and extreme responses. The mean paired change is −0.0830 min (95% interval: [−0.0913, −0.0753]). The 11-order and 13-order groups each contain one cohort and support case inspection. The distributions locate task combinations requiring allocation review and retain the adverse responses obscured by the average.

Figure 4f subtracts the complete G distributions at matching quantiles. Most quantiles remain in the zero mass; larger differences appear in the upper tail. The distributional reduction is concentrated in larger residual requirements. Matched cohort IDs in Figure 4c–e complement this distributional view by identifying the actual tasks that improve or worsen.

The priorities differed within 1215 multi-order cohorts. These cohorts accounted for all 270 net fewer conflicts and all 38 increases. Among 9277 multi-order cohorts with equal priorities, 211 had an initial joint conflict, and 31 remained positive after equal extensions. A further 3280 cohorts contained one order. In either group, the individual rule reproduces uniform allocation. A dispatcher can identify these cases before computing the individual allocation and use a joint route to look for a better distribution.

With the 6-min cap, slack-ranked targeting left 565 certified-positive cohorts and one undetermined. Joint-route targeting left 512 positive and none undetermined, compared with 565 and none for individual targeting. The 95th percentile of per-order extension was 6 min for slack ranking and 3 min for either weighted rule. A dispatcher with a baseline route can use its lateness to direct the same budget toward orders that compete along the route (Supplementary Table S6).

4.4. Per-Order Caps Control Long Extensions and Quantify the Retained Improvement

The cap determines how much targeting is possible at a fixed mean budget. In Figure 5a, a 3-min budget and 3-min cap force every order to receive 3 min, so uniform and burden-based allocation coincide. Uniform, 6-min-capped, and uncapped allocation leave 835, 565, and 470 positive cohorts, respectively, with shares of 6.06%, 4.10%, and 3.41%. Relative to uniform allocation, targeting removes a net 270 cohorts with the cap and 365 without it. The cap retains 74.0% of the uncapped net reduction while leaving 95 additional positive cohorts, or about 0.69 percentage points. The comparison quantifies the residual-conflict cost of limiting individual promise changes.

Figure 5b orders all 38,626 extensions and exposes the long-extension tail. Without a cap, P99 is 9.04 min, and the maximum reaches 24.51 min. A fixed mean allowance can therefore conceal large changes for individual customers. The 6-min cap lowers the maximum by 18.51 min while retaining 270/365 of the net reduction in positive cohorts. Combining the cap with the mean budget controls this concentration and makes the retained feasibility benefit explicit.

Figure 5c traces where the capped minutes go. Relative to uncapped allocation, 3209.43 order-minutes are removed from 764 orders and reassigned to 2366 other orders within their original cohorts. Another 35,496 orders remain unchanged within 10^{-8} min. The cap preserves the full allowance through within-cohort redistribution. In Figure 5d, independently sorted curves measure each strategy's concentration of minutes, while the lower trace uses a common q ranking to compare allocation to the same priority positions. Each strategy uses 115,878 order-minutes. The two views show both conserved resource use and the orders receiving the promise changes.

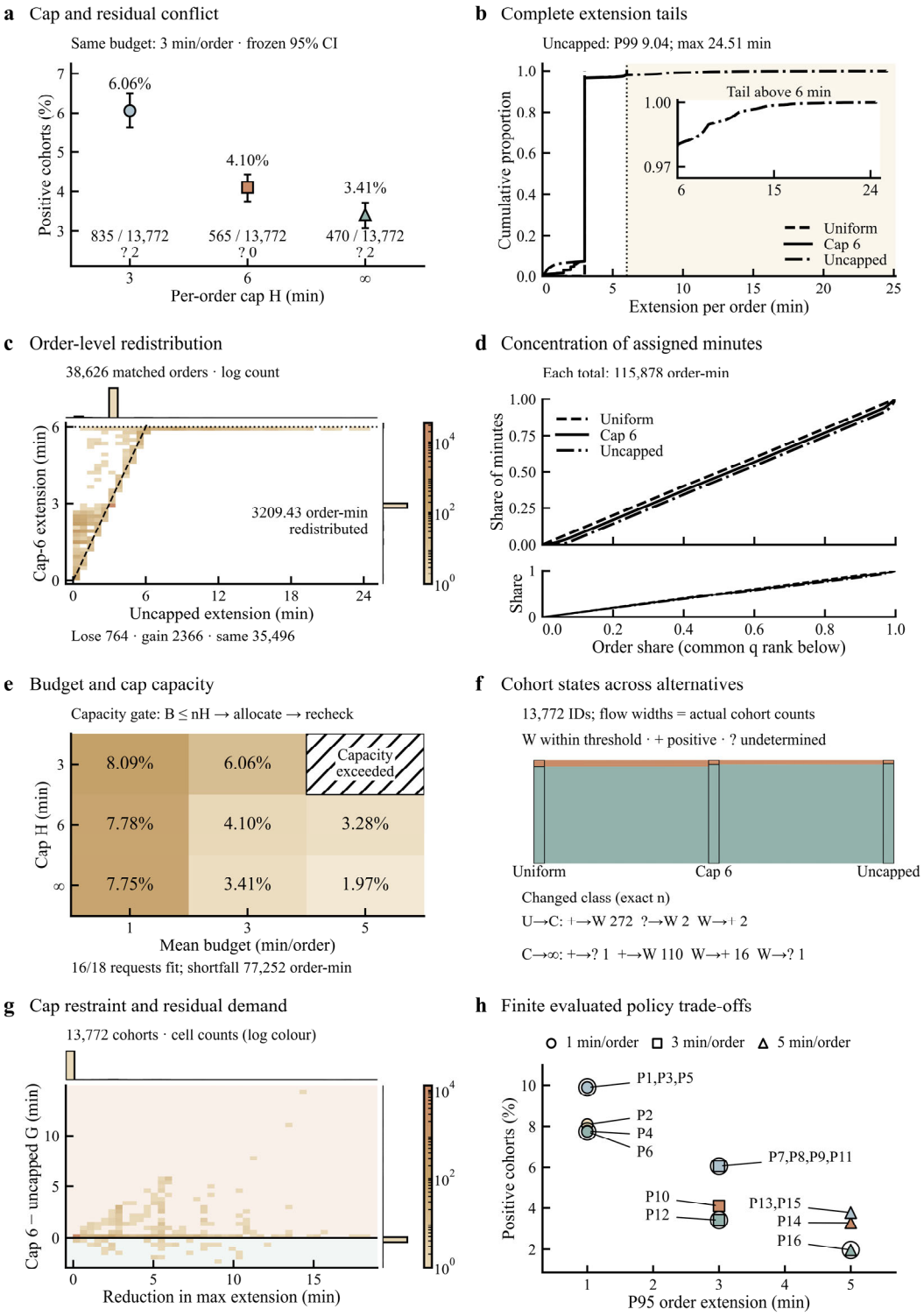


Figure 5. Feasibility gains and promise-extension costs of per-order caps. (**a**,**b**) Residual positive-cohort shares and empirical extension distributions under three caps. (**c**,**d**) Transfer of minutes between orders and cumulative concentration. (**e**) Mean budgets crossed with caps. (**f**) Paired classifications of 13,772 cohorts. (**g**) Reduction in maximum extension versus change in residual deficit. (**h**) Positive share versus the 95th percentile of extensions. The 38,626 orders are the same across settings. Counts use $L > 0.02$ min. In (**a**,**f**), a question mark denotes an undetermined classification: the lower bound is at most 0.02 min and the upper bound exceeds 0.02 min. The over-capacity setting has no allocation.

Figure 5e checks mean budgets of 1, 3, and 5 min against caps of 3 min, 6 min, and no cap. A 5-min budget with a 3-min cap exceeds total order capacity by 77,252 order-minutes and must be rejected as capacity-incompatible before allocation. The cohort transitions in Figure 5f then check the outcomes of feasible requests. Under the 6-min-cap rule, 272 uniform-positive cohorts become feasible within tolerance, two previously within-tolerance cohorts become positive, and two undetermined cohorts become within-tolerance. Shared-ID flows distinguish resolved conflicts from adverse transitions, yielding the net reduction of 270. Capacity and paired outcome checks thus address separate requirements for applying a budget–cap rule.

Figure 5g pairs the reduction in each cohort's maximum extension with its residual-G change. Upper-right cohorts obtain a shorter maximum extension at the cost of a larger residual requirement; lower-right cohorts improve on both. Figure 5h compares the 16 implementable alternatives using P95 extension and positive-cohort share. Rings mark alternatives that no other evaluated rule at the same budget improves on both axes. Rule selection can start with an acceptable extension distribution and absolute cap, then compare residual conflicts and adverse cohort changes among admissible alternatives.

Supplementary Figure S6 shows how the cohort budget is conserved. Supplementary Figures S7 and S8 show the effect of the route-selection criterion and variation across the 23 zones.

If the budget is an upper limit, the platform can keep minutes in reserve. Following the selected route and removing extensions beyond those needed for its residual bound used 6227.01 of 115,878 available order-minutes. All 12,090 initially feasible cohorts received zero. The calculation retained a route certificate for every cohort (Supplementary Figure S11).

4.5. Exact Resource Benchmarks Identify Allocation Opportunities and Service Outcomes

Exact resource benchmarks identify why a task set remains infeasible. In Figure 6a, B^*/n is the minimum order-minute requirement per order. Among 48 small instances, 38, 42 and 48 have legal routes under caps of 3 min, 6 min and no cap; the 3-min-per-order budget supports 38, 42 and 44 instances. These counts separate restrictions imposed by the cap from shortages of total allowance. Figure 6b ranks B^*/n and shows the budget margin $3 - B^*/n$: negative values quantify the additional average allowance required. The aligned U, B and E rows compare uniform allocation, burden-based allocation and exact resource feasibility for the same tasks.

At the 0.02-min criterion and a 6-min cap, uniform allocation made 38 of the 48 selected instances feasible. Individual targeting and exact enumeration each made 42 feasible. For these 42, we compared the extensions needed by the returned route with the exact minimum at the same tolerance. The mean excess was 0.204 order-min, the median was zero, and the maximum was 2.167 order-min. Eight returned routes required more than the minimum. With the cap removed, the counts were 38, 43 and 44. One feasible allocation was therefore missed even though sufficient minutes were available. Supplementary Table S8 compares the individual extensions needed on these routes.

Simulated late-order shares against recorded promises were 7.62%, 7.27% and 7.08% under uniform, capped and uncapped individual allocation, respectively. Against revised promises, the shares were 3.00%, 2.11% and 1.81% (Figure 6d). The first comparison holds the customer's reference time fixed and isolates the effect of the selected route on simulated completion. The second also credits the revised promise.

The corresponding mean simulated route durations were 16.218, 16.148, and 16.129 min. Capped targeting advanced the final completion by 4.22 s on average. The larger operational difference concerns which promises the route can meet within the same al-

lowance. Figure 6e,f relate the simulated route-end changes to compliance with the recorded promises.

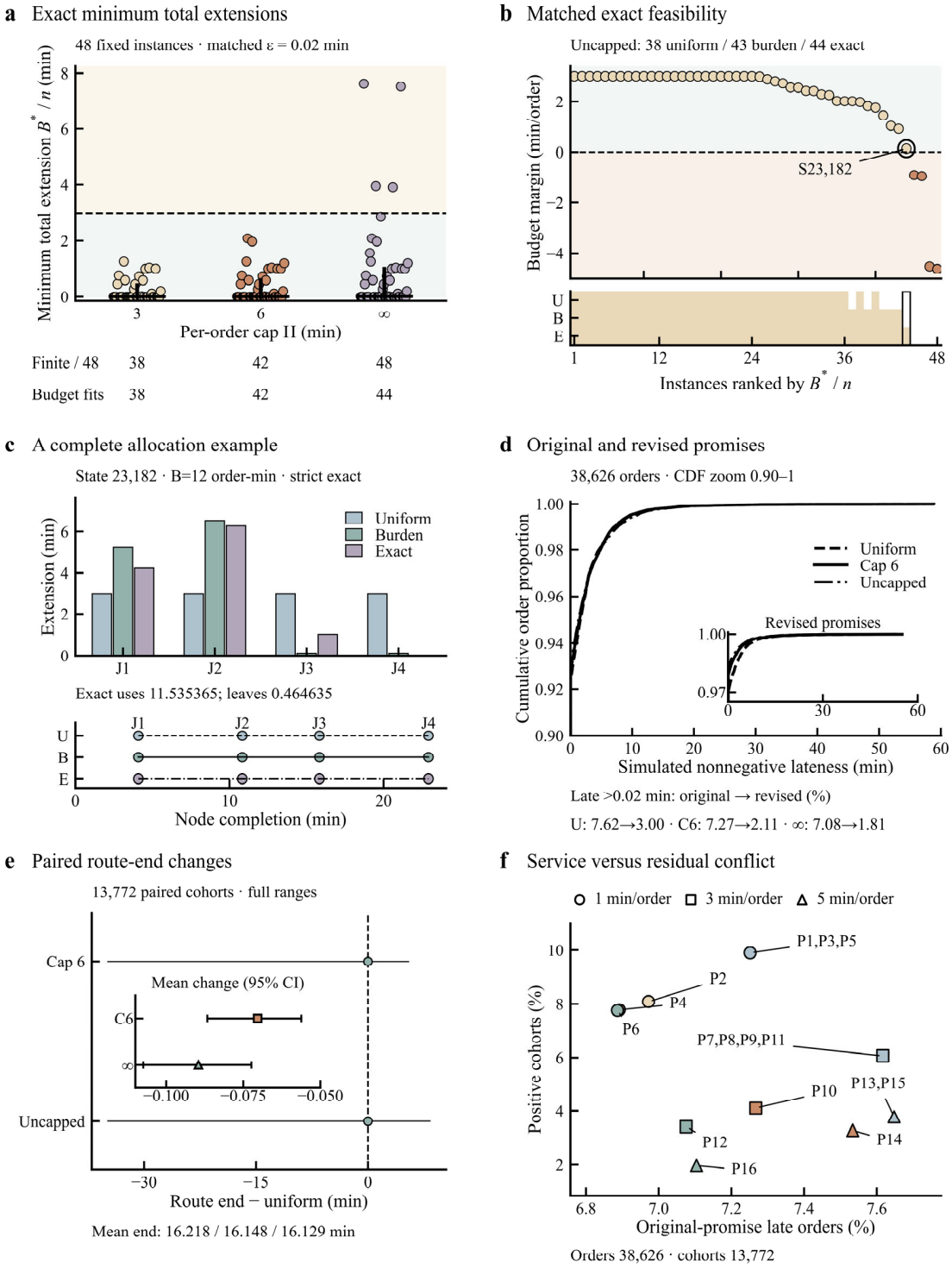


Figure 6. Minimum-total-extension benchmarks, task plans and service outcomes. (**a**,**b**) Exact minimum order-minutes per order and budget margins for 48 small instances. In (**b**), the asterisk in B^* denotes the minimum total extension across eligible routes under the specified cap and tolerance. (**c**) Extensions by order for a four-order instance. (**d**) Simulated lateness against recorded and revised promises for 38,626 orders. (**e**) Paired changes in route end time. (**f**) Order-level lateness versus cohort-level residual conflicts over 16 implementable settings. The 0.02-min criterion applies to the feasibility counts. End time is the last completion on the selected route.

4.6. Sample Lineage and Task Coverage Define the Operating Population

The release contains 654,343 waybill rows and 568,546 unique accepted orders. Of 62,044 rider snapshots, 18,667 were empty. Of the 43,377 nonempty states, 590 failed rider matching, 743 more contained an order accepted after the snapshot, and 26 more contained a completed order. This leaves 42,018 consistent states. The first daily snapshot contained 13,819 states, of which 47 exceeded 14 nodes. The other 28,199 consistent states occurred at the two later snapshots. Figure 7 shows the overlapping policy and recorded-fulfilment samples.

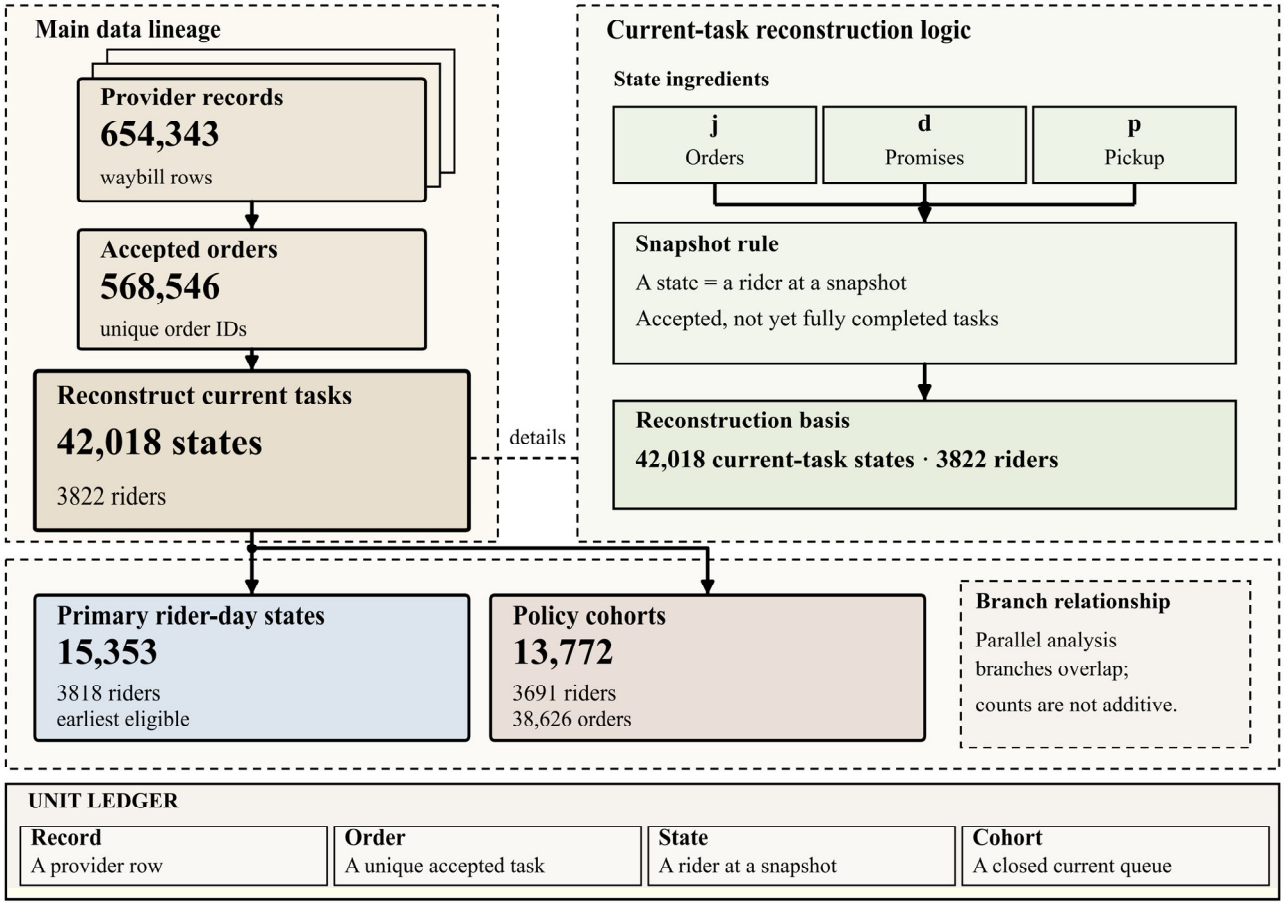


Figure 7. Public records, current-task reconstruction and analysis populations. Provider rows are linked to unique accepted orders, then to unfinished tasks for each rider at a snapshot. The two branches select 15,353 rider-days for subsequent fulfilment and 13,772 cohorts for allocation. They share underlying records. Counts along a branch use the unit printed beside each box.

Figure 8a shows 24 cells for eight dates and three snapshot ranks, with 1659–1831 states per cell and 5084–5405 per day. The three ranks contain 13,819, 13,989, and 14,210 states. These counts define each period's contribution to the comparison. Figure 8b distinguishes order count from computational workload: a collected order retains one delivery node, whereas an uncollected order also requires a pickup node. Consequently, equally sized order queues can require different amounts of route computation. Remaining-node count can guide solver allocation while order count describes the workload faced by the rider.

Figure 8c compares minimum remaining promise time for one order, two to three orders, and at least four orders, with 9567, 19,173, and 13,278 states. Negative values identify expired promises; the median, interquartile range, P5–P95, and full range retain common and extreme slack. These distributions locate near-deadline tasks within each workload

group. Combined with node counts, they guide review priority and solver effort within the observed lunch-peak population.

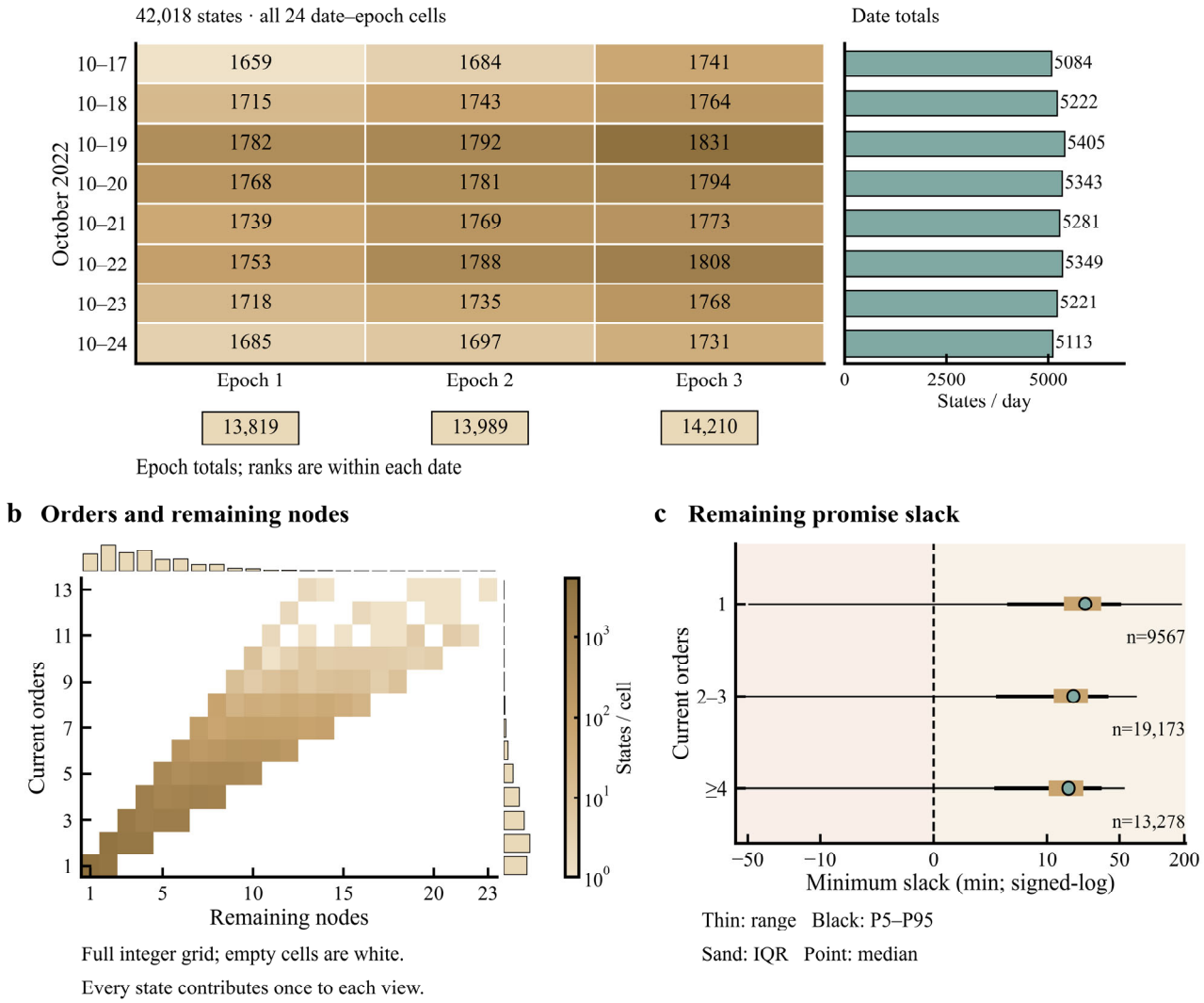


Figure 8. Eight-day snapshot coverage, current orders and remaining nodes. (**a**) Consistent states by date and snapshot rank over eight days. (**b**) Active orders and remaining pickup or delivery nodes. Color gives cell frequency. (**c**) Shortest time to a recorded promise by order-count group. Time is measured in minutes from the snapshot. The thin line spans the full range, the thick black line spans the 5th–95th percentiles, the sand bar spans the interquartile range, and the point marks the median.

All 47 first-snapshot states above 14 nodes entered an additional bounded comparison. These states averaged 8.98 orders versus 2.80 in the policy sample. Uniform, capped, and uncapped rules gave 3, 2, and 3 certified-positive cases, respectively, with one undetermined under each. These counts describe the 47 heavier queues in this release (Supplementary Table S5).

4.7. Budget and Cap Choices

Figure 9 quantifies the return from increasing the mean allowance at a fixed cap. Rows show budgets of 1, 3 and 5 min; columns give residual-G distributions for 13,772 cohorts and extension distributions for 38,626 orders. With $H = 6$ min, positive-cohort shares are 7.78%, 4.10% and 3.28%. Increasing the budget from 1 to 3 min adds 2 min per order and lowers the positive share by about 3.68 percentage points. The next 2-min increase, from 3

to 5 min, lowers it by about 0.82 points. The smaller second gain quantifies diminishing returns within the evaluated grid, allowing further promise relaxation to be weighed against residual-conflict reduction.

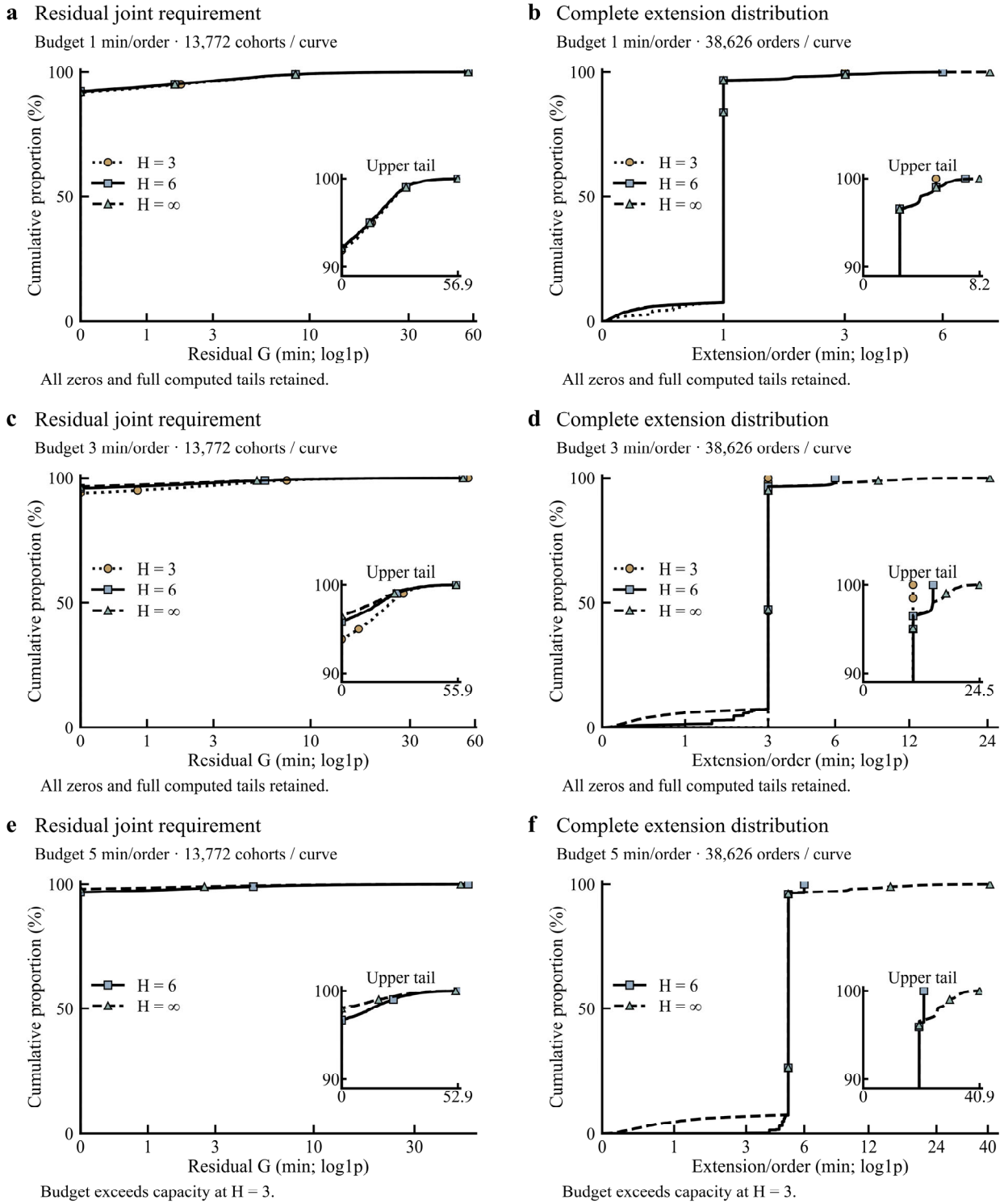


Figure 9. Residual requirements and extension distributions across evaluated budgets and caps. Panels (**a**,**b**), (**c**,**d**), and (**e**,**f**) show mean allowances of 1, 3, and 5 min per order, respectively. Panels (**a**,**c**,**e**) show cumulative distributions of residual deficit for 13,772 cohorts per setting. Panels (**b**,**d**,**f**) show cumulative distributions of promise extensions for 38,626 orders. Curves distinguish the evaluated policies and caps and include the point mass at zero. The 5-min allowance exceeds capacity under the 3-min cap.

The paired distributions show what each budget leaves unresolved. Residual-G tails locate substantial remaining requirements; P95, P99, and the maximum describe the spread and concentration of customer promise changes. Zero values retain tasks meeting the feasibility criterion. The 5-min budget with a 3-min cap is marked as capacity-incompatible. The discrete grid supports a choice among evaluated budget–cap combinations based on both residual conflicts and order extensions. Customer acceptance and the financial value of these changes are separate inputs to a business decision.

4.8. Decision Thresholds, Operating Inputs and Runtime

Raising the decision threshold from 0.02 min to 1, 2 and 5 min changed the uniform-minus-capped difference from 1.96 to 1.53, 1.30 and 0.58 percentage points. At 5 min, capped targeting left 167 positive cohorts versus 247 for uniform allocation. The customer-facing threshold therefore determines how many queues merit review, while numerical accuracy remains fixed (Supplementary Tables S2 and S3).

Across ten changes to operating inputs, the same-budget advantage of individual targeting ranged from 1.04 percentage points with no initial rest to 3.19 points with five minutes of rest. Counting service once per consecutive merchant stop gave a 1.96-point difference. With complete readiness forecasts, the difference was 1.97 points. The 32.3% relative reduction and 74.0% retained fraction describe the reference scenario (Supplementary Table S5).

The complete capped decision took a median of 0.028 ms and a 99th percentile of 0.281 ms over 13,771 warm calls. At 14 nodes, the 95th percentile was 2.71 ms. Every main-sample interval reached the numerical target. These local computation times identify the heavier queues for which a dispatcher may need to retain a bound (Supplementary Table S7).

5. Discussion

5.1. Contribution to Platform Promise Management

Promise adjustment is an upstream management decision through which platforms can accommodate specified operating time in already-assigned deliveries. Established routing models supply completion constraints, and bounded resource allocation supplies budget and cap enforcement. The framework links total allowance, order-level distribution, and individual limits to a common feasibility check. The analytical quantities have distinct uses: G measures the common extension needed, b_j specifies a customer's adjustment, and B^* tests the sufficiency of the total allowance on enumerated tasks. A residual conflict can then be examined as a routing, allocation, or resource problem before additional promise time is granted.

The equal-budget comparison demonstrates why the placement of promise minutes matters. Reallocating 115,878 order-minutes lowers residual-positive cohorts from 835 to 565 while retaining the same tasks, rider assignments, and 3-min initial rest. Separately, 952 joint-excess states identify requirements missed by individual bounds, and 474 heuristic-positive states become feasible through route search. The evidence supports a sequence of management checks: examine the whole queue, test the visit plan, then allocate the available promise allowance. This decision complements capacity planning, batching and workforce scheduling [16,20,22] by addressing customer commitments after the workload has been assigned.

5.2. Selecting the Allowance and Customer-Facing Cap

The mean allowance and individual cap control different aspects of the service decision. When both equal 3 min, targeting is impossible because every order must receive 3 min. A 6-min cap permits redistribution and retains 74.0% of the uncapped net reduction while

lowering the longest extension from 24.51 to 6 min. Relative to uncapped allocation, the cap exchanges about 0.69 percentage points of additional residual conflict for an 18.51-min reduction in the largest promise change. Managers can set an individual service limit separately from the total allowance and quantify the associated loss in modelled feasibility.

Allocation objectives and individual constraints shape resource outcomes [35,36]. The mean budget controls the aggregate change in promises; extension percentiles and maxima show its distribution across customers. Concentration curves locate the assigned minutes, and paired transitions identify task sets that improve or worsen. Table 4 links these measures to rule selection: set the allowance and cap, compare the conflict–extension trade-off, and review adverse cohort outcomes. Customer preference research indicates that delivery attributes and delay-compensation thresholds carry different values across users [11,13]. The cap is therefore an operational control; its customer acceptance and commercial terms require local assessment.

Table 4. Key result measures and their uses in promise management.

Measure	Result in This Study	Use in an Engineering Decision
Positive residual deficit	835 → 565 cohorts; 6.06% → 4.10%	Assess residual task conflicts under the same budget and operating conditions.
P95 and P99	Uncapped P99: 9.04 min	Set extension thresholds for most customers and review the longer tail.
Maximum order extension	Uncapped: 24.51 min; capped: 6 min	Set an absolute per-order promise-change limit and quantify its feasibility cost.
Net reduction retained by the cap	270/365 ≈ 74.0%	Compare the conflict reduction retained when an individual cap is imposed.
Cohort budget and B^*	$B^* \le B_w \le n_w H$	Identify whether to revise the budget, cap, or order-level allocation.
Order-level redistribution	3209.43 order-min; 764 donor and 2366 recipient orders	Verify redistribution within the original rider task set when the cap binds.
Lateness under two promise definitions	Original: 7.62% → 7.27%; updated: 3.00% → 2.11%	Assess simulated service against the original and revised commitments separately.
Route end time	16.218 → 16.148 min; about 4.22 s earlier on average	Check whether promise adjustments also change task completion time.

Service measures then check the operating consequences against unchanged and revised commitments. In the main simulation, recorded-promise late-order shares fall from 7.62% to 7.27% and revised-promise shares from 3.00% to 2.11%, while mean route completion is about 4.22 s earlier. The dominant gain is a better match between promise minutes and task requirements. Retaining the recorded-promise measure keeps the initial service obligation visible after commitments are revised. For small cohorts with remaining conflicts, B^* distinguishes inadequate allowance from improvable allocation. The combined checks indicate whether to reconsider the budget or cap, or redistribute existing minutes.

The relative reduction of 32.3% equals 270/835, and the retained uncapped reduction of 74.0% equals $270/(835 - 470)$. Both use net positive-cohort changes; extension percentiles and confidence intervals separately describe individual promise changes and estimation uncertainty.

5.3. Limitations and Field Validation

The model is evaluated on eight lunch periods from one platform and 23 anonymized zones. The primary policy comparison covers the first daily snapshot and up to 14 remaining nodes. The three released decision times cluster near 11:25, 11:27, and 11:30 local time. Evening demand, new assignments during a route, and customer responses to changing promises require separate observations.

Preparation forecasts, local travel times, and service at a shared merchant determine whether a conflict is actionable. The order records supply the recorded customer promise but no revision history. The initial rest is a scenario input. Before changing a promise, a dispatcher would need current estimates, a record of the communicated deadline, and a rule for notifying the customer.

The recorded-arrival comparison tests whether historical execution is compatible with the model [48,49]. Independent survey respondents describe the working environment (Supplementary Figure S9). Previous studies have examined occupational health and safety among gig drivers [73] and road-safety compliance, riding behaviour, and work-related factors among food-delivery riders [74–76]. Establishing effects on actual breaks, fatigue, traffic incidents, and customer acceptance would require outcomes linked to an implemented policy.

6. Conclusions

For a fixed set of orders, the common-extension interval identifies whether a route change can meet the recorded promises. Uniform extensions translate this requirement by the same amount, down to zero. In the reconstructed states, route search found 474 cases where a positive heuristic deficit could be removed within tolerance.

Across 13,772 cohorts, reallocating the same allowance by individual order need reduced certified residual conflicts from 835 to 565 under a 6-min per-order cap. Joint-route priorities reduced the count to 512. A dispatcher can use the individual rule as a first calculation and spend the additional route computation where orders compete for the same minutes.

The cap retained 74.0% of the net reduction achieved without a cap and limited the largest extension to 6 min. On the 48 enumerated small instances, the exact resource requirement distinguished a shortage of minutes from an allocation that could be improved. An operational trial could test how these choices affect actual deliveries, breaks, and customer acceptance.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/su182312173/s1. Figure S1: Numerical checks by independent enumeration and uniform promise translation. Figure S2: Relative geometry, preparation releases and node service inputs. Figure S3: Complete fingerprints of 131 model–record compatibility differences. Figure S4: Paired predictive increments from structural time quantities across task baselines. Figure S5: Predictive increments for unseen riders and two historical test dates. Figure S6: Cohort budget conservation and capped order-level extensions. Figure S7: Stored computation times and route end times at a common upper bound. Figure S8: Allocation responses and task composition across 23 anonymous zones. Figure S9: Perceived algorithmic management and self-reported behaviour related to timely delivery. Figure S10: Sample selection and coverage. Figure S11: Priority groups and optional budget use. Figure S12: Decision thresholds and operating inputs. Figure S13: Allocation alternatives, exact resource use and computation. Table S1: The first common snapshot occurs near 11:25 local time on each date. Table S2: Positive/undetermined counts above each operational threshold. Table S3: Capped-minus-uniform differences in the positive-cohort percentage. Table S4: Fixed-prediction rider resampling on the two historical test dates. Table S5: Recomputed positive counts and differences under each set of inputs. Table S6: Allocation alternatives on the 13,772 policy

cohorts. Table S7: End-to-end calculation from one in-memory state through distances, individual completion, allocation, revised promises, joint bounds and earliest ending route. Table S8: Per-order extensions (minutes) for the 125 orders in the 42 feasible exact instances at the six-minute cap and 0.02-min tolerance. Source_Data.

Author Contributions: Conceptualization, Z.X.; methodology, Z.X. and M.S.; software, Z.X. and P.H.; validation, M.S., Y.Z. and P.H.; formal analysis, Z.X.; investigation, Y.Z.; data curation, Z.X.; writing—original draft preparation, Z.X.; writing—review and editing, Z.X., M.S., Y.Z. and P.W.; visualization, Z.X.; supervision, Y.Z.; project administration, P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This study used publicly available, anonymized secondary operational and survey datasets. No participants were recruited or contacted, and no intervention or collection of identifiable personal information was undertaken. For the original rider survey, Zhang et al. [6] reported ethical approval from the academic committee of the first author's home university. The present study analyzed the released anonymous survey data separately from the operational records.

Informed Consent Statement: No new participant recruitment or consent collection was undertaken for this secondary analysis. Zhang et al. [6] reported obtaining online informed consent from participants in the original survey. The present study used the publicly released anonymized data.

Data Availability Statement: Operational records and their terms of use are available through the Meituan–INFORMS TSL Research Challenge [51]. The independent survey is available with Zhang et al. [6]. The executable reconstruction, allocation, exact enumeration, statistical analysis, and revision-figure scripts are available at https://github.com/Xzr-CUMTB/food-delivery-promise-allocation, accessed on 23 November 2026 (revision e244b3d, committed on 11 November 2026). The Supplementary Tables report the aggregate comparisons.

Acknowledgments: This research was supported by data provided by Meituan. The researchers who conducted the independent survey made the anonymous data publicly available [6].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Meemken, E.M.; Bellemare, M.F.; Reardon, T.; Vargas, C.M. Research and policy for the food-delivery revolution. *Science* **2022**, *377*, 810–813. [CrossRef] [PubMed]
2. Alvarez-Palau, E.J.; Calvet-Liñán, L.; Viu-Roig, M.; Gandouz, M.; Juan, A.A. Economic profitability of last-mile food delivery services: Lessons from Barcelona. *Res. Transp. Bus. Manag.* **2022**, *45*, 100659. [CrossRef]
3. Christie, N.; Ward, H. Delivering hot food on motorcycles: A mixed method study of the impact of business model on rider behaviour and safety. *Saf. Sci.* **2023**, *158*, 105991. [CrossRef]
4. Nguyen-Phuoc, D.Q.; Nguyen, L.N.T.; Su, D.N.; Nguyen, M.H.; Oviedo-Trespalacios, O. Deadly meals: The influence of personal and job factors on burnout and risky riding behaviours of food delivery motorcyclists. *Saf. Sci.* **2023**, *159*, 106007. [CrossRef]
5. Kavta, K.; Sharif Azadeh, S.; Maknoon, Y.; Wang, Y.; Homem de Almeida Correia, G. Estimating the value of safety against road crashes: A stated preference experiment on route choice of food delivery riders. *Transp. Res. Part C Emerg. Technol.* **2025**, *179*, 105272. [CrossRef]
6. Zhang, L.; Yang, J.; Zhang, Y.; Xu, G. Gig worker's perceived algorithmic management, stress appraisal, and destructive deviant behavior. *PLoS ONE* **2023**, *18*, e0294074. [CrossRef] [PubMed]
7. Zhang, Y.; Wang, D.; Liu, Y.; Du, K.; Lu, P.; He, P.; Li, Y. Urban food delivery services as extreme heat adaptation. *Nat. Cities* **2025**, *2*, 170–179. [CrossRef]
8. Kellogg, K.C.; Valentine, M.A.; Christin, A. Algorithms at work: The new contested terrain of control. *Acad. Manag. Ann.* **2020**, *14*, 366–410. [CrossRef]
9. Yu, D.; Zhang, J.; Yun, G. Delivery riders' safety and delivery efficiency in on-demand food delivery industry: The moderating role of monitoring algorithms. *Res. Transp. Bus. Manag.* **2024**, *55*, 101143. [CrossRef]
10. Chiang, C.-H.; Chung, K.-C. Understanding the risks: Health and safety challenges for food delivery couriers. *Res. Transp. Bus. Manag.* **2025**, *62*, 101441. [CrossRef]

11. Chen, J.; Sharif Azadeh, S.; Maknoon, Y.; Jiang, H. The value of delay compensation in online food delivery: Analysis of willingness to pay and preference heterogeneity. *Res. Transp. Bus. Manag.* **2026**, *67*, 101710. [CrossRef]
12. Kiba-Janiak, M.; Marcinkowski, J.; Jagoda, A.; Skowrońska, A. Sustainable last mile delivery on e-commerce market in cities from the perspective of various stakeholders. Literature review. *Sustain. Cities Soc.* **2021**, *71*, 102984. [CrossRef]
13. Kiba-Janiak, M.; Cheba, K.; Mucowska, M.; de Oliveira, L.K.; Piecyk, M.; Evangelista, P.; Prockl, G.; Rześny-Cieplińska, J. How to design a sustainable last-mile delivery and returns business model from E-Customers' expectations perspective? *Res. Transp. Bus. Manag.* **2024**, *56*, 101194. [CrossRef]
14. Leveson, N. A new accident model for engineering safer systems. *Saf. Sci.* **2004**, *42*, 237–270. [CrossRef]
15. Zio, E. The future of risk assessment. *Reliab. Eng. Syst. Saf.* **2018**, *177*, 176–190. [CrossRef]
16. Yildiz, B.; Savelsbergh, M. Service and capacity planning in crowd-sourced delivery. *Transp. Res. Part C Emerg. Technol.* **2019**, *100*, 177–199. [CrossRef]
17. Steever, Z.; Karwan, M.; Murray, C. Dynamic courier routing for a food delivery service. *Comput. Oper. Res.* **2019**, *107*, 173–188. [CrossRef]
18. Gu, R.; Liu, Y.; Poon, M. Dynamic truck–drone routing problem for scheduled deliveries and on-demand pickups with time-related constraints. *Transp. Res. Part C Emerg. Technol.* **2023**, *151*, 104139. [CrossRef]
19. Ulmer, M.W.; Thomas, B.W.; Campbell, A.M.; Woyak, N. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transp. Sci.* **2021**, *55*, 75–100. [CrossRef]
20. Simoni, M.D.; Winkenbach, M. Crowdsourced on-demand food delivery: An order batching and assignment algorithm. *Transp. Res. Part C Emerg. Technol.* **2023**, *149*, 104055. [CrossRef]
21. Kim, S.H.; Chung, B.D. Integrated food delivery problem considering both single-order and multiple order deliveries. *Comput. Ind. Eng.* **2024**, *196*, 110458. [CrossRef]
22. Ulmer, M.W.; Savelsbergh, M. Workforce scheduling in the era of crowdsourced delivery. *Transp. Sci.* **2020**, *54*, 1113–1133. [CrossRef]
23. Ciacco, A.; Luzzi, M.; Scalzo, E. Meal delivery with unmanned aerial vehicle: A case study from the University of Calabria. In *Beyond Frontiers of Operations Research: Emerging Technologies and Innovative Optimization Paradigms*; Springer: Cham, Switzerland, 2025; Volume 14, pp. 87–98. [CrossRef]
24. Garg, V.; Niranjan, S.; Prybutok, V.; Pohlen, T.; Gligor, D. Drones in last-mile delivery: A systematic review on efficiency, accessibility, and sustainability. *Transp. Res. Part D* **2023**, *123*, 103831. [CrossRef]
25. Jazairy, A.; Persson, E.; Brho, M.; von Haartman, R.; Hilletofth, P. Drones in last-mile delivery: A systematic literature review from a logistics management perspective. *Int. J. Logist. Manag.* **2025**, *36*, 1–62. [CrossRef]
26. Liu, W.; Liu, L.; Qi, X. On-time meal delivery assisted by drone resupply. *Nav. Res. Logist.* **2026**, *73*, 1047–1066. [CrossRef]
27. Solomon, M.M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **1987**, *35*, 254–265. [CrossRef]
28. Taillard, É.; Badeau, P.; Gendreau, M.; Guertin, F.; Potvin, J.Y. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transp. Sci.* **1997**, *31*, 170–186. [CrossRef]
29. Vidal, T.; Crainic, T.G.; Gendreau, M.; Prins, C. Time-window relaxations in vehicle routing heuristics. *J. Heuristics* **2015**, *21*, 329–358. [CrossRef]
30. Pillac, V.; Gendreau, M.; Guéret, C.; Medaglia, A.L. A review of dynamic vehicle routing problems. *Eur. J. Oper. Res.* **2013**, *225*, 1–11. [CrossRef]
31. Psaraftis, H.N.; Wen, M.; Kontovas, C.A. Dynamic vehicle routing problems: Three decades and counting. *Networks* **2016**, *67*, 3–31. [CrossRef]
32. Berbeglia, G.; Cordeau, J.F.; Laporte, G. Dynamic pickup and delivery problems. *Eur. J. Oper. Res.* **2010**, *202*, 8–15. [CrossRef]
33. Savelsbergh, M.; Van Woensel, T. 50th anniversary invited article—City logistics: Challenges and opportunities. *Transp. Sci.* **2016**, *50*, 579–590. [CrossRef]
34. Lu, M.; Liu, R.; Jin, Z.; Yuan, Q. A causal discovery and inference framework for on-demand food delivery delays. *npj Sustain. Mobil. Transp.* **2026**, *3*, 22. [CrossRef]
35. Matl, P.; Hartl, R.F.; Vidal, T. Workload equity in vehicle routing problems: A survey and analysis. *Transp. Sci.* **2018**, *52*, 239–260. [CrossRef]
36. Karsu, Ö.; Morton, A. Inequity averse optimization in operational research. *Eur. J. Oper. Res.* **2015**, *245*, 343–359. [CrossRef]
37. Bertsimas, D.; Farias, V.F.; Trichakis, N. The price of fairness. *Oper. Res.* **2011**, *59*, 17–31. [CrossRef]
38. Savelsbergh, M.W.P.; Sol, M. The general pickup and delivery problem. *Transp. Sci.* **1995**, *29*, 17–29. [CrossRef]
39. Campbell, A.M.; Savelsbergh, M. Efficient insertion heuristics for vehicle routing and scheduling problems. *Transp. Sci.* **2004**, *38*, 369–378. [CrossRef]
40. Ropke, S.; Pisinger, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **2006**, *40*, 455–472. [CrossRef]

41. Bellman, R. Dynamic programming treatment of the travelling salesman problem. *J. ACM* **1962**, *9*, 61–63. [CrossRef]
42. Held, M.; Karp, R.M. A dynamic programming approach to sequencing problems. *J. Soc. Ind. Appl. Math.* **1962**, *10*, 196–210. [CrossRef]
43. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
44. Laporte, G. Fifty years of vehicle routing. *Transp. Sci.* **2009**, *43*, 408–416. [CrossRef]
45. Patriksson, M. A survey on the continuous nonlinear resource allocation problem. *Eur. J. Oper. Res.* **2008**, *185*, 1–46. [CrossRef]
46. Patriksson, M.; Strömberg, C. Algorithms for the continuous nonlinear resource allocation problem—New implementations and numerical studies. *Eur. J. Oper. Res.* **2015**, *243*, 703–722. [CrossRef]
47. Savelsbergh, M.W.P. The vehicle routing problem with time windows: Minimizing route duration. *ORSA J. Comput.* **1992**, *4*, 146–154. [CrossRef]
48. Sargent, R.G. Verification and validation of simulation models. *J. Simul.* **2013**, *7*, 12–24. [CrossRef]
49. Oberkampf, W.L.; Trucano, T.G.; Hirsch, C. Verification, validation, and predictive capability in computational engineering and physics. *Appl. Mech. Rev.* **2004**, *57*, 345–384. [CrossRef]
50. Hund, L.; Schroeder, B. A causal perspective on reliability assessment. *Reliab. Eng. Syst. Saf.* **2020**, *195*, 106678. [CrossRef]
51. Meituan. The First INFORMS TSL Data-Driven Research Challenge: Background and Data Description, 21 March 2024, Table 2, p. 6. Available online: https://github.com/meituan/Meituan-INFORMS-TSL-Research-Challenge (accessed on 26 September 2026).
52. Voccia, S.A.; Campbell, A.M.; Thomas, B.W. The same-day delivery problem for online purchases. *Transp. Sci.* **2019**, *53*, 167–184. [CrossRef]
53. Agatz, N.; Campbell, A.; Fleischmann, M.; Savelsbergh, M. Time slot management in attended home delivery. *Transp. Sci.* **2011**, *45*, 435–449. [CrossRef]
54. Behrendt, A.; Savelsbergh, M.; Wang, H. A prescriptive machine learning method for courier scheduling on crowdsourced delivery platforms. *Transp. Sci.* **2023**, *57*, 889–907. [CrossRef]
55. Neria, G.; Hildebrandt, F.D.; Tzur, M.; Ulmer, M.W. The restaurant meal delivery problem with ghost kitchens. *Transp. Sci.* **2025**, *59*, 433–450. [CrossRef]
56. Roberts, D.R.; Bahn, V.; Ciuti, S.; Boyce, M.S.; Elith, J.; Guillera-Arroita, G.; Hauenstein, S.; Lahoz-Monfort, J.J.; Schröder, B.; Thuiller, W.; et al. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography* **2017**, *40*, 913–929. [CrossRef]
57. Cawley, G.C.; Talbot, N.L.C. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107. Available online: https://www.jmlr.org/papers/v11/cawley10a.html (accessed on 23 November 2026).
58. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
59. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. Available online: https://jmlr.org/papers/v12/pedregosa11a.html (accessed on 23 November 2026).
60. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
61. Davis, J.; Goadrich, M. The relationship between precision-recall and ROC curves. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 233–240. [CrossRef]
62. Saito, T.; Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* **2015**, *10*, e0118432. [CrossRef] [PubMed]
63. Brier, G.W. Verification of forecasts expressed in terms of probability. *Mon. Weather Rev.* **1950**, *78*, 1–3. [CrossRef]
64. Gneiting, T.; Raftery, A.E. Strictly proper scoring rules, prediction, and estimation. *J. Am. Stat. Assoc.* **2007**, *102*, 359–378. [CrossRef]
65. Van Calster, B.; McLernon, D.J.; van Smeden, M.; Wynants, L.; Steyerberg, E.W. Calibration: The Achilles heel of predictive analytics. *BMC Med.* **2019**, *17*, 230. [CrossRef] [PubMed]
66. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. *Proc. Mach. Learn. Res.* **2017**, *70*, 1321–1330. Available online: https://proceedings.mlr.press/v70/guo17a.html (accessed on 23 November 2026).
67. Efron, B. Bootstrap methods: Another look at the jackknife. *Ann. Stat.* **1979**, *7*, 1–26. [CrossRef]
68. Field, C.A.; Welsh, A.H. Bootstrapping clustered data. *J. R. Stat. Soc. B* **2007**, *69*, 369–390. [CrossRef]
69. Peng, R.D. Reproducible research in computational science. *Science* **2011**, *334*, 1226–1227. [CrossRef] [PubMed]
70. Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.W.; da Silva Santos, L.B.; Bourne, P.E.; et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **2016**, *3*, 160018. [CrossRef] [PubMed]
71. Sandve, G.K.; Nekrutenko, A.; Taylor, J.; Hovig, E. Ten simple rules for reproducible computational research. *PLoS Comput. Biol.* **2013**, *9*, e1003285. [CrossRef] [PubMed]

72. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef] [PubMed]
73. Christie, N.; Ward, H. The health and safety risks for people who drive for work in the gig economy. *J. Transp. Health* **2019**, *13*, 115–127. [CrossRef]
74. Nguyen-Phuoc, D.Q.; Nguyen, N.A.N.; Nguyen, M.H.; Nguyen, L.N.T.; Oviedo-Trespalacios, O. Factors influencing road safety compliance among food delivery riders: An extension of the job demands-resources (JD-R) model. *Transp. Res. Part A Policy Pract.* **2022**, *166*, 541–556. [CrossRef]
75. Oviedo-Trespalacios, O.; Rubie, E.; Haworth, N. Risky business: Comparing the riding behaviours of food delivery and private bicycle riders. *Accid. Anal. Prev.* **2022**, *177*, 106820. [CrossRef] [PubMed]
76. Papakostopoulos, V.; Nathanael, D. The complex interrelationship of work-related factors underlying risky driving behavior of food delivery riders in Athens, Greece. *Saf. Health Work* **2021**, *12*, 147–153. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.