Electronic Companion

This electronic companion provides implementation details and supplementary analyses for the paper *"Dispatch or Hold? An Inverse Optimization and Reinforcement Learning Approach for Multi-Objective On-Demand Delivery"*.

EC.1. Cutting-Plane Algorithm

Section 4.1 of the main paper formulates the data-driven inverse optimization problem and describes its master problem and cut-generating forward subproblems. Algorithm EC.1 provides the complete implementation-level procedure. Starting from a feasible coefficient vector, each iteration solves the forward multi-objective weighted set partitioning (MOWSP) problem for every observed instance and evaluates the resulting absolute suboptimality loss. Instances that violate the current inverse optimality condition generate cuts for the master problem, which is then resolved to update the coefficient vector. Because this estimation is conducted offline, the repeated forward solves do not affect the speed of online dispatch decisions.

Algorithm EC.1: Cutting-Plane Algorithm for Data-Driven Inverse Weighted Set Partitioning

---

Input: Instances $\{(O_i^{\text{disp}}, C^i, \hat{x}^i)\}_{i=1}^N$

Output: Estimated weights $\theta^{(k)} = (\theta_1^{(k)}, \theta_2^{(k)}, \theta_3^{(k)})$ minimizing the total suboptimality of $\hat{x}^i$.

Initialize: $\theta^{(0)} \leftarrow$ initial guess; construct master problem $\mathcal{M}$: $\min_{\theta, \{\delta_i \geq 0\}} \sum_i \delta_i$

for $k = 1, 2, \ldots$ *until convergence* **do**

  for $i = 1$ **to** N **do**

    Solve forward problem (6) with $\theta^{(k-1)}$ to obtain $x^{i*}(\theta^{(k-1)})$

    Compute suboptimality gap $l_{\text{ASO}}(\hat{x}^i, x^{i*}(\theta^{(k-1)}))$ as in Equation (11)

    if $l_{\text{ASO}}(\hat{x}^i, x^{i*}(\theta^{(k-1)})) > 0$ **then**

      Add the constraint below to $\mathcal{M}$: `// Add a new cut`

      $l_{\text{ASO}}(\hat{x}^i, x^{i*}(\theta^{(k-1)})) \leq \delta_i$, **where** θ are the *decision variables* in $\mathcal{M}$, while $x^{i*}(\theta^{(k-1)})$ and $\hat{x}^i$ are *fixed*.

  Solve master problem $\mathcal{M}$ to update $\theta^{(k)}$ and $\{\delta_i^{(k)}\}$

  if *no cuts added* **then**

    break

return $\theta^{(k)}$

EC.2. PPO Hyperparameters

Table EC.1 reports the network architecture and optimization settings used in all PPO experiments. The settings follow commonly used PPO defaults and were refined through preliminary experiments to obtain stable convergence in the dispatch environment.

	Parameter	**Value**
Network Architecture	Hidden Layer Dimension (d_{ffn})	128
	Number of Attention Layers (L)	4
	Number of Attention Heads (H)	4
	Dropout Probability	0.1
PPO Configuration	Learning Rate (base, min)	$1 \times 10^{-4}, 1 \times 10^{-5}$
	Discount Factor (γ)	0.99
	GAE Parameter (λ)	0.95
	Clipping Ratio (ϵ)	0.2
	Number of Epochs per Rollout	10
	Value Loss Coefficient (ζ_v)	0.5
	Entropy Loss Coefficient (ζ_e)	0.001
	Cardinalty Loss Coefficient (ζ_c)	0.001
	Batch Size	64
	Rollout Size	12000
	Maximum Gradient Norm	0.5
	Total Steps	1×10^6

Table EC.1 Hyperparameters for the Proximal Policy Optimization (PPO) Algorithm

EC.3. Courier Usage of PPO Under Cardinality Regularization

Because detailed real-time courier locations, workloads, and availability are not observed, the PPO loss includes a cardinality regularization term that penalizes deviations between the expected number of dispatched orders and the dispatch volume observed under the practice policy. In order to evaluate whether this term provides an operationally reasonable proxy for courier capacity, Figure EC.1 compares the average number of couriers used per dispatch period by the PPO and practice policies. The PPO policy closely tracks the practice-policy benchmark and remains at or slightly below it over the evaluated hours, indicating that the regularization term prevents systematic over-usage of couriers.

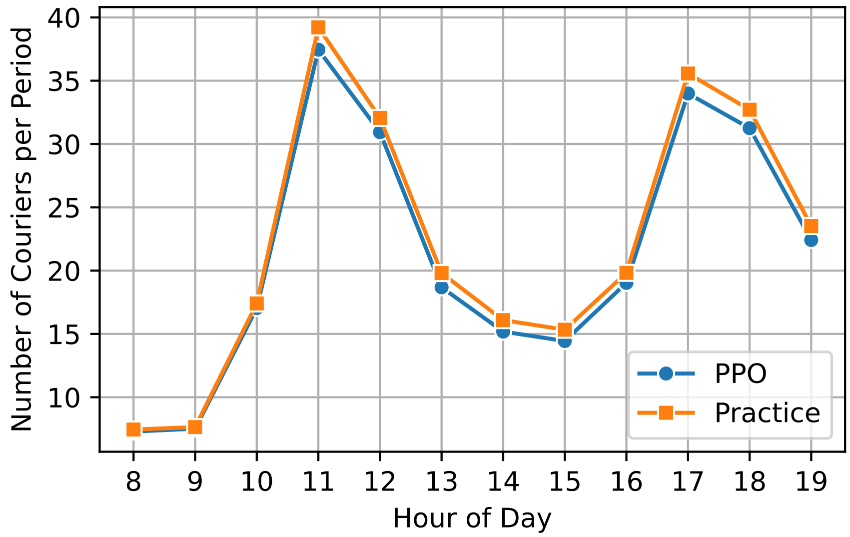

Figure EC.1 **Average number of couriers used per dispatch period, aggregated by hour over 8–19 h. The PPO policy closely tracks and does not exceed the courier usage observed under the practice policy.**

EC.4. Performance Benefits of Data-Driven Adaptive Weights

The data-driven inverse optimization (DDIO) module estimates time-varying objective weights that reflect the platform's implicit trade-offs among travel distance, delivery timeliness, and courier usage. To assess whether these adaptive weights reproduce observed grouping decisions more accurately than a fixed specification, Table EC.2 compares the test-set percentage error obtained with DDIO weights against that obtained with equal weights. DDIO yields a lower mean error in every time window and substantially reduces the mean error during the midday and evening peak periods. Its lower standard deviations also indicate more consistent performance across test instances.

Table EC.2 **Evaluation Percentage Error (%) under data-driven inverse optimization (DDIO) weights compared with equal weights.**

Time	**DDIO**				**Equal Weights**			
Time window	Mean	Std	Min	Max	Mean	Std	Min	Max
8–10	1.16%	3.66%	0.00%	22.22%	1.93%	5.19%	0.00%	34.62%
11–13	3.23%	3.36%	0.00%	16.67%	7.96%	6.88%	0.00%	33.33%
14–16	3.21%	2.95%	0.00%	19.44%	7.44%	5.09%	0.00%	27.50%
17–19	2.74%	3.44%	0.00%	17.65%	7.59%	6.42%	0.00%	34.29%

The downstream value of the adaptive weights is evaluated by training PPO policies under alternative objective specifications. Table EC.3 compares PPO+DDIO with an equal-weight PPO policy and three single-objective PPO policies. PPO+DDIO achieves the lowest weighted cost and provides the most balanced performance across distance, courier usage, and service-timeliness measures. The single-objective policies can improve the metric they prioritize, but those gains are offset by deterioration in the remaining operational dimensions and overall costs.

Table EC.3 Performance comparison of PPO variants under different objective specifications. PPO+DDIO denotes the PPO model trained with DDIO-derived objective weights, while the other variants are trained using equal-weight or single-objective formulations. Each entry reports the mean value, with the deviation relative to PPO+DDIO shown in brackets.

	Distance (km/order)	Courier (#/order)	Delay Rate (%)	Delay (min /late order)	Cost per order
PPO+DDIO	1.43	0.84	4.52	4.88	1.78
PPO (Equal θ)	1.44 (+0.82%)	0.85 (+0.50%)	4.67 (+3.46%)	4.94 (+1.07%)	1.81 (+0.85%)
PPO ($\theta_1 = 1$)	1.39 (−2.92%)	0.91 (+7.29%)	6.01 (+33.08%)	5.19 (+6.19%)	1.86 (+3.93%)
PPO ($\theta_2 = 1$)	1.46 (+1.95%)	0.99 (+17.17%)	3.60 (−20.26%)	4.90 (+0.25%)	1.87 (+4.49%)
PPO ($\theta_3 = 1$)	1.61 (+12.93%)	0.74 (−12.92%)	19.36 (+328.66%)	6.15 (+25.90%)	2.03 (+13.67%)