Hybrid Constraint Satisfaction and Markov Decision Process Framework for Delivery Scheduling under Uncertainty

Your Name

September 9, 2025

Preface and Acknowledgements

This thesis is submitted in partial fulfillment of the requirements for the degree of . . .
I would like to thank . . .

Abstract

This thesis investigates the optimization of delivery scheduling in uncertain environments by integrating Constraint Satisfaction Problems (CSPs) and Markov Decision Processes (MDPs). The hybrid approach aims to minimize lateness, reduce resource usage, and maintain feasibility with respect to hard operational constraints. CSPs are used to enforce strict scheduling constraints such as capacity limits and delivery time windows, while MDPs capture stochastic elements such as uncertain travel times and dynamic order arrivals. The proposed hybrid framework applies exact Value Iteration on small-scale toy scenarios and employs Monte Carlo rollout methods for large-scale delivery simulations. Experimental evaluation compares baseline CSP-only methods with the hybrid approach, demonstrating improved robustness and adaptability under uncertainty.

Contents

Chapter 1

Introduction

Efficient delivery scheduling is a central challenge in logistics and warehouse operations. Traditional optimization approaches often assume deterministic travel times and static order sets, yet real-world systems are inherently uncertain. Factors such as traffic conditions, vehicle breakdowns, and last-minute orders make robust scheduling difficult.

This thesis addresses the problem by combining two complementary approaches:

- Constraint Satisfaction Problems (CSPs) to enforce hard scheduling constraints.
- Markov Decision Processes (MDPs) to model stochastic decision-making under uncertainty.

The key research question is: *How can a hybrid CSP–MDP framework improve delivery scheduling performance under uncertain conditions?*

Chapter 2

Literature Review

Relevant work spans three domains:

1. CSP-based scheduling and routing methods (e.g., vehicle routing with time windows).
2. Markov Decision Process methods for uncertain planning.
3. Hybrid AI methods that combine deterministic optimization with probabilistic models.

This thesis builds on these ideas by applying them specifically to delivery systems.

Chapter 3

Methodology

3.1 Constraint Satisfaction Model

We define binary assignment variables:

$$x_{v,i} = \begin{cases} 1 & \text{if vehicle } v \text{ serves order } i, \\ 0 & \text{otherwise.} \end{cases}$$

Constraints:

$$\sum_v x_{v,i} = 1 \quad \forall i \in \mathcal{C} \quad \text{(each order assigned)} \tag{3.1}$$

$$\sum_{i \in \sigma_v} q_i \le Q_v \quad \forall v \quad \text{(capacity)} \tag{3.2}$$

$$a_i \le t_i \le b_i - s_i \quad \forall i \quad \text{(time windows)} \tag{3.3}$$

Objective function:

$$\min \sum_i w^{(L)} \max(0, t_i + s_i - d_i) + w^{(D)} \sum_v \text{dist}(\sigma_v) + w^{(R)} \cdot \#\text{vehicles}$$

3.2 Markov Decision Process Model

An MDP is defined by the tuple (S, A, P, R, γ):

- States $s \in S$: vehicle positions, pending orders, current time.
- Actions $a \in A(s)$: continue, reroute, reassign, wait, drop order.
- Transitions:
$$P(s'|s,a) = \Pr(\text{next state } s' \mid s, a)$$
- Reward function:
$$R(s,a) = -\left(\alpha \cdot \text{lateness} + \beta \cdot \text{travel cost} + \gamma \cdot \mathbf{1}_{\text{dropped}}\right)$$

3.3 Hybridization Approaches

Three approaches are considered:

1. **Sequential Hybrid**: CSP generates a baseline plan, MDP handles local disruptions.
2. **Coupled Iterative Hybrid**: MDP simulates futures, CSP reoptimizes with sampled scenarios.
3. **Constraint-Guided Policy**: MDP proposes actions, CSP filters infeasible ones.

3.4 Value Iteration for Small Instances

The Bellman optimality equation:

$$V^*(s) = \max_{a \in A(s)} \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s') \right]$$

Algorithm 1 Value Iteration

```
Initialize V(s) = 0 for all states s
repeat
    Δ ← 0
    for each state s do
        v ← V(s)
        V(s) ← max_{a∈A(s)} ( R(s,a) + γ Σ_{s'} P(s'|s,a) V(s') )
        Δ ← max(Δ, |v − V(s)|)
    end for
until Δ < θ
Derive policy: π(s) = arg max_{a∈A(s)} [· · ·]
```

3.5 Monte Carlo Rollout for Large Instances

For large-scale delivery, exact value iteration is infeasible. Monte Carlo rollout approximates the Bellman update by sampling:

$$Q(s,a) \approx \frac{1}{N} \sum_{i=1}^{N} \left(R(s,a) + \gamma V(s'_i) \right)$$

where s'_i are sampled next states under (s,a).

—

Chapter 4

Results

- **Toy Case Study**: Exact Value Iteration for 1 vehicle, 3 deliveries.
- **Large Simulation**: Monte Carlo rollout with 20–50 deliveries.
- Compare baseline CSP-only vs hybrid CSP+MDP.

Chapter 5

Discussion

The sequential hybrid was computationally efficient but limited in adaptability. Monte Carlo rollout enabled scalable decision-making with reduced lateness, though at the cost of increased runtime.

Chapter 6

Conclusion

This thesis demonstrated a hybrid CSP–MDP approach to delivery scheduling under uncertainty. Exact value iteration provided a benchmark on small cases, while Monte Carlo rollout enabled scalable solutions. Future work includes integrating reinforcement learning and more sophisticated stochastic models.

References