Optimized Routing and Scheduling in Home Health Care Using Deep Reinforcement Learning

by Eliud Omollo

B.S. in Electrical Engineering, May 2003, Pennsylvania State University
M.S. in Electrical and Computer Engineering, May 2006, Purdue University
M.A. in Data Science, May 2020, Harvard University

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

January 9, 2026

Praxis directed by

Myron Boyd
Professorial Lecturer in Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Eliud Omollo has passed the Final Examination for the degree of Doctor of Engineering as of November 3, 2025. This is the final and approved form of the Praxis.

Optimized Routing and Scheduling in Home Health Care Using Deep Reinforcement Learning

Eliud Omollo

Praxis Research Committee:

Amir Etemadi, Adjunct Professor of Engineering and Applied Science, Praxis Chair

Maron Boyd, Professorial Lecturer in Engineering and Applied Science, Praxis Advisor

Hamza Alsarhan, Professorial Lecturer in Engineering and Applied Sciences, Committee Member

© 2025 by Eliud Omollo
All rights reserved

Dedication

In loving memory of my sister, Rebecca Atieno, who always believed I could achieve more than I imagined. Her love and faith continue to give me strength and purpose.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Myron Boyd, for his guidance and support throughout this project.

I am also thankful to my team at CarePine Home Health, whose dedication and professionalism allowed me the time and focus to pursue this research.

The motivation for this study originated from my experience managing CarePine Home Health, where I observed firsthand the complexities of scheduling in home health care. These experiences inspired the formulation of this problem.

Finally, I thank my family and close friends for their encouragement and understanding during this process

Abstract of Praxis

Optimized Routing and Scheduling in Home Health Care Using Deep Reinforcement Learning

The growing elderly population in the U.S. has increased demand for home care services. Even though the primary objective of home health agencies is to provide patient care, home health care regulations, labor laws, contracts, and clinical standards require them to follow additional guidelines that may compromise their ability to do so. These hard constraints include matching clinician skills against patient care needs, scheduling clinicians during designated time windows, observing work time limits, and so on. Additionally, home health agencies must balance other secondary objectives, such as travel efficiency, equitable work distribution, patient preferences, and continuity of care.

Classical scheduling methods typically handle these soft objectives sequentially. They first maximize the primary objective, such as visit fulfillment, before addressing secondary objectives. Similarly, metaheuristic scheduling algorithms often optimize a single primary objective rather than balancing conflicting objectives at the same time.

To address the limitations of single objective optimization techniques shown in classical, heuristics and metaheuristic methods, we use deep reinforcement learning(DRL) to train an agent to account for competing secondary objectives. We start by formulating the problem as a Markov decision process. We define a reward function as an aggregator function composed of weighted signals of each secondary objective we wish to prioritize. We then train the agent using Proximal Policy Optimization algorithm to maximize the reward. The agent interacts with a simulation environment using samples of varying instance sizes (small: 40 patients, 120 visits, 5 clinicians; medium: 80 patients, 225 visits, 10 clinicians; large: 150 patients, 430 visits, 20 clini-

cians). Action masking enforces hard constraints so that clinicians are never assigned outside of their work hours, visits occur within specified time windows, and skill requirements are satisfied. We configure the weights of the reward function to reflect priorities typically seen in home health. Visit fulfillment receives a higher weight than secondary objectives to reflect its position as the primary objective in home health care. To account for different home health operational priorities, we also implement three aggregator variants, *Weighted Sum*, *Chebyshev*, and *Policy-Aware*.

We evaluated the DRL model against exact optimization method, metaheuristic method, and greedy approaches to address three questions. Does the DRL model achieve competitive *Visit Fulfillment* rates on small and medium problem instances? Does the DRL model explore the *Pareto frontiers* more effectively along visit fulfillment, idle rate, and utilization objectives? Does the DRL model perform comparably in the *General Composite Score* composed of ten secondary metrics ?

The results reveal that performance varies by instance size. On small size instances, the DRL model scored 96% on *Visit Fulfillment* compared to 81% for exact optimization methods. On medium size instances, the DRL model achieved a *Visit Fulfillment* rate of of 79-80% , showing no significant difference from the benchmark methods. Across small and medium scale, the DRL model achieved 6-140 times greater *Pareto hypervolume* than all benchmark approaches. These results show that DRL model has greater ability to explore trade-offs in small and medium instances. The *General Composite Score* shows 23% improvement over exact optimization method at small scale and comparative at medium scale. At large scale, the DRL model scored 53% on *Visit Fulfillment*, compared to 90% with exact optimization method. This indicates that either substantially more training, or architectural changes would be necessary for the DRL model to be comparable at larger instances.

This praxis demonstrates that deep reinforcement learning is a viable routing and scheduling solution for small and medium size home health care agencies. The novelty

of this work lies in dynamic action masking to enforce constraints, a decomposed reward function that uses weights to assign priority to secondary objectives and a regularization strategy to provide generalization over unseen instances.

Table of Contents

Dedication iv

Acknowledgments v

Abstract of Praxis vi

Table of Contents ix

List of Figures xiv

List of Tables xvi

Nomenclature xviii

Glossary xxiv

1 Introduction 1

1.1 Background 1

1.2 Research Motivation 2

1.3 Problem Statement 2

1.4 Thesis Statement 3

1.5 Research Objectives 3

1.6 Research Questions and Hypotheses 4

1.6.1 Research Questions 4

1.6.2 Research Hypotheses 4

1.7 Scope of Research 5

1.8 Research Limitations 6

1.9 Organization of Praxis 6

2 Literature Review **8**
2.1 Classical Optimization Approaches to HHCRSP **8**
2.2 Evolution Toward Heuristics and Metaheuristics **8**
2.2.1 Multiobjective with fairness. **9**
2.2.2 Dynamic insertion and real time adaptation. **9**
2.2.3 Multiobjective with Continuity. **9**
2.3 Introduction of Learning-Based Methods **10**
2.3.1 Foundations of Reinforcement Learning for Routing and Healthcare **10**
2.3.2 Handling Stochasticity and Uncertainty in Dynamic Environments **11**
2.3.3 Zero-Shot Generalization in Deep Reinforcement Learning **11**
2.3.4 Reward Shaping, Efficiency, and Sample Complexity **11**
2.3.5 Fairness-Aware Reinforcement Learning in Scheduling **12**
2.3.6 Addressing Overestimation and Stability Challenges **12**
2.3.7 Robustness under Real-World Environmental Disruptions **13**
2.4 Unaddressed Gaps in the Literature **13**

3 Methodology **16**
3.1 Dataset **16**
3.2 Constraint Derivation and Feasibility Mapping **21**
3.3 Feature Encoding for Policy Learning **23**
3.4 Reward Function Design **24**
3.4.1 Weighted Sum Aggregator. **25**
3.4.2 Chebyshev Aggregator. **25**
3.4.3 Policy-Aware Aggregator. **26**

3.5 Reward Components 26
3.5.1 Fulfillment Reward. 27
3.5.2 Travel Efficiency Penalty. 27
3.5.3 Workload Fairness Reward. 28
3.5.4 Continuity Reward. 28
3.5.5 Preference Satisfaction Reward. 29
3.6 Addressing Generalization and Credit Assignment 30
3.6.1 Exploration and Exploitation. 30
3.6.2 Credit Assignment in Long Horizons. 30
3.6.3 Function Approximation Stability. 31
3.6.4 Generalization Across Instances. 31
3.7 Ethics-by-Design 31
3.7.1 Workload Equity. 31
3.7.2 Relational Continuity. 32
3.7.3 Preference Satisfaction. 32
3.8 PPO Protocol 32
3.9 Environment Simulation 33
3.10 Training 35
3.11 Baselines and Solvers 35
3.11.1 Nearest Neighbor (Greedy Construction). 35
3.11.2 CP-SAT (Exact Optimization). 36
3.11.3 Adaptive Large Neighborhood Search (Iterative Improvement). 37

4 Results 39
4.1 Hypothesis 1 39
4.1.1 Visit Fulfillment - Small Instances 40
4.1.2 Visit Fulfillment - Medium Instances 42

4.1.3 Visit Fulfillment - Large Instances 42
4.2 Hypothesis 2A 44
4.2.1 Hypervolume - Small Instances 47
4.2.2 Hypervolume - Medium Instances 50
4.2.3 Hypervolume - Large Instances 53
4.3 Hypothesis 2B 56
4.3.1 Set Coverage - Small Instances 60
4.3.2 Set Coverage - Medium Instances 60
4.3.3 Set Coverage - Large Instances 61
4.4 Hypothesis 3 62
4.4.1 General Composite Score - Small Instances 62
4.4.2 General Composite Score - Medium Instances 63
4.4.3 General Composite Score - Large Instances 64

5 Conclusions 79
5.1 Summary of Findings 79
5.2 Practical Applications 81
5.3 Limitations 81
5.4 Future Research 82
5.4.1 Hybrid optimization. 83
5.4.2 Adaptive weighting. 83
5.4.3 Curriculum learning. 83
5.4.4 Transfer learning. 83
5.4.5 Explainability. 84
5.4.6 Field validation. 84
5.5 Concluding Remarks 84

Bibliography 85

A Appendix A **89**

List of Figures

3.1 Partial Patient Data **18**

3.2 Partial Clinician Data **18**

3.3 Partial Continuity Data **19**

3.4 Partial Speed Data **20**

3.5 Partial Traffic Data **21**

4.1 3D Pareto Frontiers - Weighted Sum (Small) **48**

4.2 3D Pareto Frontiers - Policy Aware (Small) **49**

4.3 3D Pareto Frontiers - Chebyshev (Small) **50**

4.4 3D Pareto Frontiers - Weighted Sum (Medium) **51**

4.5 3D Pareto Frontiers - Policy Aware (Medium) **52**

4.6 3D Pareto Frontiers - Chebyshev (Medium) **53**

4.7 3D Pareto Frontiers - Weighted Sum (Large) **54**

4.8 3D Pareto Frontiers - Policy Aware (Large) **55**

4.9 3D Pareto Frontiers - Chebyshev (Large) **56**

4.10 Operational Balance - Metrics Components - Weighted Sum (Small) **66**

4.11 Operational Balance - Metrics Components - Policy Aware (Small) **67**

4.12 Operational Balance - Metrics Components - Chebyshev (Small) **68**

4.13 Operational Balance - Metrics Components - Weighted Sum (Medium) **69**

4.14 Operational Balance - Metrics Components - Policy Aware (Medium) **70**

4.15 Operational Balance - Metrics Components - Chebyshev (Medium) **71**

4.16 Operational Balance - Metrics Components - Weighted Sum (Large) **72**

4.17 Operational Balance - Metrics Components - Policy Aware (Large) **73**

4.18 Operational Balance - Metrics Components - Chebyshev (Large) **74**

4.19 Operational Balance — GCS_{10} - Weighted Sum (Small) **74**

4.20 Operational Balance — GCS_{10} - Policy Aware (Small) **75**

4.21 Operational Balance — GCS_{10} - Chebyshev (Small) **75**

4.22 Operational Balance — GCS_{10} - Weighted Sum (Medium) **76**

4.23 Operational Balance — GCS_{10} - Policy Aware (Medium) **76**

4.24 Operational Balance — GCS_{10} - Chebyshev (Medium) **77**

4.25 Operational Balance — GCS_{10} - Weighted Sum (Large) **77**

4.26 Operational Balance — GCS_{10} - Policy Aware (Large) **78**

4.27 Operational Balance — GCS_{10} - Chebyshev (Large) **78**

A.1 Sample Caregiver Schedule **90**

List of Tables

4.1 Visit Fulfillment: DRL vs Benchmarks — Performance Metrics (Small) 41

4.2 Visit Fulfillment: DRL vs Benchmarks — Performance Metrics (Medium) 43

4.3 Visit Fulfillment: DRL vs Benchmarks — Performance Metrics (Large) 45

4.4 Visit Fulfillment: DRL vs Benchmarks — Statistical Comparison 46

4.5 Comprehensive Performance Metrics: DRL vs Benchmarks 57

4.6 Detailed Statistical Tests: DRL vs Benchmarks (Mann-Whitney U) 58

4.7 Pareto Dominance Summary: DRL vs Benchmarks Across Problem Sizes and Aggregators 58

4.8 Statistical Significance of DRL vs Benchmarks (Mann-Whitney U Test, p=0.05) 59

4.9 Comprehensive GCS_{10} Summary: Mean ± SD Across Problem Sizes and Aggregators 65

4.10 Statistical Significance for Comprehensive GCS_{10}: DRL vs Baselines (p-value / Cliff's δ) 65

4.11 Win-Tie-Loss Summary for Comprehensive GCS_{10}: DRL vs Baselines (Instance-Level Comparisons) 65

4.12 Comprehensive GCS_{10} Metric Definition: Component Metrics, Types, Optimization Direction, and Weights 66

Nomenclature

Indices and Sets

i	Index for caregivers, $i \in \mathcal{G}$
j	Index for visits, $j \in \mathcal{V}$
d	Index for days in the planning horizon, $d \in \mathcal{D}$
t	Time index in minutes, $t \in \{0, \ldots, 10080\}$
$\mathcal{G}$	Set of caregivers
$\mathcal{P}$	Set of patients
$\mathcal{V}$	Set of patient visits (each corresponding to a scheduling request)
$\mathcal{D}$	Set of scheduling days
$\mathcal{E}$	Set of expertise categories
$\mathcal{A}_t$	Set of feasible actions at time t
$\mathcal{V}_d$	Set of visits scheduled on day d
$\mathcal{Z}$	Set of spatial zones for travel modeling
$\mathcal{A}$	Set of executed caregiver–visit assignments

Parameters

a_j, b_j	Earliest and latest allowable service time for patient j
d_j	Service duration (in minutes) for patient j
s_{ij}	Continuity-of-care score between caregiver i and patient j
θ_{ij}	Binary: 1 if caregiver i satisfies mandatory skills for j

ϕ_{ij} Binary: 1 if caregiver i satisfies optional preferences for j

$w_i^{\min}, w_i^{\max}$ Minimum and maximum weekly work time for caregiver i

$w_{id}^{\min}, w_{id}^{\max}$ Minimum and maximum daily work time for caregiver i on day d

τ_{ij}^t Travel time (minutes) from caregiver i to patient j at time t

λ_z^t Traffic multiplier for zone z at time t

T_{day} Length of a scheduling day in minutes (typically 1440)

T_{week} Total number of minutes in the scheduling week

$T_{\max}$ Maximum allowable work minutes across the week

R_j Number of required visits for patient j

λ_{idle} Weight coefficient for idle time penalty in reward function

$\lambda_{\text{fairness}}$ Weight coefficient for fairness penalty in reward function

$\lambda_{\text{continuity}}$ Weight coefficient for continuity reward in the reward function

λ_{pref} Weight coefficient for preference satisfaction reward

Decision and Policy Variables

y_{ijd} Binary variable: 1 if caregiver i visits patient j on day d

x_t Action taken at timestep t (caregiver–visit pair)

$\pi(a_t \mid s_t)$ Policy: probability of action a_t given state s_t

$\pi'(a_t \mid s_t)$ Masked policy: $\text{Softmax}(Z_t + \log \mathbf{F}_t)$

Z_t Raw policy logits at time t

$\mathbf{F}_t \in \{0,1\}^{C \times V_t}$ Feasibility mask at time t for all caregiver–visit pairs

C Number of available caregivers

V_t Number of assignable visits at time t

State and Features

$\mathbf{s}_t$ Environment state at time t

μ_t Urgency scalar $\in [0, 1]$ at time t

u_i Utilization ratio of caregiver i

u_i^{index} Relative utilization index for caregiver i

w_i^{used} Normalized workload actually used for caregiver i

$I_{i,t}$ Cumulative idle time (in minutes) for caregiver i up to time t

I_i Total idle time for caregiver i

F_j Fulfillment status vector for patient j

S_j Set of fulfilled visits for patient j

P_j^{opt} Flag indicating optional preferences met for patient j

τ_{ij}^{avail} Indicator: 1 if caregiver i is temporally compatible with visit j, else 0

σ_j Time slack (in minutes) between caregiver availability end and patient latest time window

C_{ij} Normalized continuity score between caregiver i and patient j

R_{ij} Recency score (in $[0, 1]$) based on time since caregiver i last visited patient j

S_{ij} Ratio of past visits where caregiver i satisfied optional skill preferences for patient j

τ_i^{tight} Average daily time window tightness for caregiver i

σ_i^{dayvar} Standard deviation of daily max work times for caregiver i

$\delta_i^{\text{compact}}$ Average number of days between scheduled workdays for caregiver i

δ^{workload} Standard deviation of workload across all caregivers

d_1 Dimensionality of caregiver feature vectors in the observation encoder

d_2 Dimensionality of visit feature vectors in the observation encoder

$\bar{u}$ Mean utilization ratio across all caregivers

Skill Representation

S_i Binary skill vector of caregiver i (length $|\mathcal{E}|$)

R_j^m Binary vector of mandatory skills required for visit j

R_j^o Binary vector of optional skills preferred for visit j

ρ_i Skill redundancy: number of skills caregiver i possesses, $\sum_{e \in \mathcal{E}} S_i[e]$

κ_j Skill complexity: total number of required and optional skills for visit j, $\sum_{e \in \mathcal{E}} (R_j^m[e] + R_j^o[e])$

PPO and Learning Parameters

$V(s_t)$ State-value function

$Q(s_t, a_t)$ Action-value function

$A(s_t, a_t)$ Advantage function

$\hat{A}_t$ Estimated advantage at time t

δ_t Temporal difference (TD) error

$\hat{r}_t$ Observed reward at time t

γ Discount factor for future reward

λ GAE (Generalized Advantage Estimation) smoothing parameter

ϵ PPO clipping threshold

η Entropy regularization coefficient

α Learning rate

Reward Components

r_t Total reward at time t

r^{fulfill} Reward for fulfilling a required patient visit

r^{cont} Reward for continuity of care

r^{travel} Penalty for long or inefficient travel

r^{idle} Penalty for idle time between visits

r^{fair} Penalty for workload imbalance

r^{pref} Reward for satisfying patient optional skill preferences

R Cumulative reward: $R = \sum_{t=0}^{T} \gamma^t r_t$

Routing and Spatial Features

ℓ_i Location associated with caregiver i at time t

ℓ_j Location associated with patient visit j

$\mathcal{Z}_\ell$ Zone assignment of location ℓ

$\mathcal{D}_{\ell_i \ell_j}$ Base travel duration (minutes) from caregiver i's location ℓ_i to patient j's location ℓ_j

$\mathcal{C}_{z_i z_j b}$ Traffic congestion multiplier between zones z_i and z_j at time block b

τ_{ij}^{t} Congestion-adjusted travel time (minutes) from caregiver i to patient j at time t, computed as $\tau_{ij}^{t} = \mathcal{A}_{\ell_i \ell_j}^{b_t}$

$\mathcal{A}_{\ell_i \ell_j}^{b}$ Precomputed travel time (minutes) from caregiver i's location ℓ_i to patient j's location ℓ_j during time block b, adjusted for traffic congestion; used to derive τ_{ij}^{t}

B Number of discrete time blocks in the planning week (e.g., 336, each representing a 30-minute interval)

$x_{\ell}^{\text{base}}, y_{\ell}^{\text{base}}$ Coordinates of location ℓ in $\mathbb{R}^2$ from MDS applied to base travel durations

$x_{\ell b}^{\text{traf}}, y_{\ell b}^{\text{traf}}$ Coordinates of location ℓ at time block b from MDS applied to traffic-adjusted durations

$L_{\ell b}$ Composite spatial embedding of location ℓ at time block b; includes both base and traffic-aware components

$\mathcal{G}_v$ Set of feasible caregivers for visit v, based on skill compatibility and routing feasibility

$\sigma_v^{\text{scarcity}}$ Skill scarcity score for visit v; defined as the inverse of $|\mathcal{G}_v|$

Glossary

Masking-Aware Decoding A technique in DRL where invalid actions are excluded during sampling or inference by modifying the logits before applying softmax. xxiv

MDP Markov Decision Process. A mathematical framework for modeling decision-making where outcomes are partly random and partly under the control of a decision maker. xxiv

Policy Gradient A family of reinforcement learning methods that directly optimize the policy by computing gradients of expected return with respect to policy parameters. xxiv

Preference Alignment The extent to which caregiver-patient assignments satisfy optional skill preferences beyond mandatory qualifications. xxiv

Reward Decomposition A strategy in which the overall reward is split into interpretable components to align training incentives with multi-objective goals. xxiv

Simulation Fidelity The degree to which the simulated environment realistically reflects operational constraints, temporal structure, and behavioral variability of real-world home care scheduling. xxiv

Time Window The allowable range of start times within which a patient visit must occur, typically defined by earliest and latest thresholds. xxiv

Chapter 1 – Introduction

1.1 Background

By the year 2050, the number of individuals aged 65 and up in the United States is projected to reach 83.9 million, nearly double from the count of 43.1 million in 2012 (U.S. Census Bureau, 2022). As this group lives longer, the demand for home health care is projected to rise (Miller et al., 2025)

Studies show that poor assignments of clinicians in home health can contribute to clinician burnout, workforce turnover, and compromised patient care (Fikar and Hirsch, 2017; Fathollahi-Fard et al., 2018). Other studies show that inefficient routing can increase travel times by up to 95% (Fikar and Hirsch, 2018). Travel time and poor scheduling can lead to delayed care, wasted time, and increased costs.

Constraint-based methods and mixed-integer programming models can minimize travel time or maximize coverage in isolation. These methods are costly and do not scale well to larger problem instances (Cappanera and Scutellà, 2015). Metaheuristic algorithms such as adaptive large neighborhood search offer faster approximations. However, they depend on hand-tuned parameters that are optimized for single objectives rather than for balancing competing goals (Fathollahi-Fard et al., 2018).

Home health care scheduling aims to maximize visit fulfillment while promoting clinician workload equity, maintaining continuity of care, and minimizing travel burden. Classical methods prioritize the primary objective of visit fulfillment and only then do they address secondary objectives. This approach is problematic due to conflicting objectives. For example, assigning a familiar clinician to maintain continuity may increase travel time; an unbalanced workload could strain patient-clinician relationships, and maximizing coverage without regard for equity can increase turnover.

Deep reinforcement learning offers an alternative. Rather than optimizing one goal at the expense of others or handling objectives sequentially, a DRL model can balance competing concerns simultaneously through a decomposed reward structure.

The DRL model learns which trade-offs matter. It examines when to prioritize coverage, when continuity justifies added travel, and how to distribute work without compromising patient care.

1.2 Research Motivation

As the demand for home health increases, so does the complexity of providing patient care. Patients require different types of care with varying skill requirements, treatments can only occur at certain times and patients are spread out in different areas. Similarly, clinicians possess different skills and qualifications, work preferences and they too are spread out in different areas. Time windows determine when visits can occur, while travel times depend on traffic patterns that shift throughout the day. The combinations of valid assignments grow exponentially with problem size.

Current approaches separate routing from assignment, treat objectives in isolation, or rely on static plans that fail to respond to scheduling dynamics seen in home health care. Studies show progress on constrained vehicle routing problems and scheduling under time windows. However, there is limited progress on methods that address multiple objectives in a balanced way.

We ask the following questions. Can a DRL model outperform classical optimization techniques across small, medium, and large instances? Can a DRL model achieve competitive *Visit Fulfillment* rates on small and medium problem instances while balancing secondary objectives? And can it do so while exploring richer *Pareto frontiers*?

1.3 Problem Statement

Home health care scheduling systems must assign clinicians to patients in a way that guarantees that clinicians have the skills required by the patients, that treatment times are observed, that clinicians can reach the patients on time, and so on. These

are hard constraints that must be legally or ethically enforced. At the same time, they must optimize multiple secondary objectives such as equitable work distribution, minimize travel time, reduce idle time, etc. These objectives often conflict with one another.

Inefficient scheduling systems disrupt patient care (Zhao et al., 2024), increase operational costs through excessive travel Fikar and Hirsch (2018), and contribute to workforce turnover through inequitable workload distribution (Fikar and Hirsch, 2017).

Existing methods either optimize a single primary objective or handle secondary objectives sequentially, rather than optimizing all objectives simultaneously while managing their conflicts.

1.4 Thesis Statement

A deep reinforcement learning model is required to optimize the primary objective of visit fulfillment while simultaneously balancing secondary home health care objectives.

1.5 Research Objectives

This research aims to accomplish the following:

1. Design a DRL model capable of addressing the primary objective of visit fulfillment while balancing secondary objectives.
2. Train the model on three different problem sizes.
3. Validate the model on unseen data.
4. Evaluate the DRL model's performance against exact optimization, heuristic, and metaheuristic methods.

1.6 Research Questions and Hypotheses

Deep reinforcement learning may outperform traditional heuristic methods in managing the demands of home health care delivery. The following research questions and hypotheses test this capability.

1.6.1 Research Questions

- **RQ1:** Does a multiobjective deep reinforcement learning model achieve competitive visit-fulfillment rates compared to single-objective optimization methods (exact optimization, metaheuristic, constructive heuristic) on small and medium instances under identical hard constraints?

- **RQ2:** Does a deep reinforcement learning model yield a Pareto frontier over fulfillment, average idle rate, and average utilization rate with greater hypervolume and set coverage (C-metric) than exact optimization, metaheuristic, and constructive heuristic methods under identical hard constraints?

- **RQ3:** Across small and medium problem instances, does a deep reinforcement learning model achieve a higher Comprehensive General Composite Score (GCS_{10}, comprising 10 metrics across efficiency, equity, quality, and operational dimensions) than exact optimization and constructive heuristic methods under identical hard constraints?

1.6.2 Research Hypotheses

- **H1:** Under identical hard constraints, a multiobjective deep reinforcement learning model attains visit-fulfillment rates that are competitive with or superior to single-objective benchmark methods (exact optimization, metaheuristic, constructive heuristic) on small and medium HHCRSP instances.

- **H2a:** Under identical hard constraints, a deep reinforcement learning model achieves a statistically significantly larger Pareto hypervolume than exact optimization, metaheuristic, and constructive heuristic methods across all instance sizes.

- **H2b:** Under identical hard constraints, a deep reinforcement learning model demonstrates superior set coverage (C-metric) compared to metaheuristic and constructive heuristic methods across all scales. It exhibits full dominance at small scales, partial dominance at medium scales, and complementary Pareto contributions at large scales when compared with the exact optimization method.

- **H3:** The deep reinforcement learning policy's mean Comprehensive GCS_{10} will exceed both exact optimization and constructive heuristic methods on small and medium problem sizes.

1.7 Scope of Research

This Praxis focuses on designing and evaluating a DRL model to address the Home Health Care Routing and Scheduling Problem (HHCRSP). We assess DRL performance across three problem scales. Small and medium instances represent scenarios where clinician capacity approximately matches or slightly exceeds patient demand. Large instances represent scenarios where demand exceeds capacity. This study also presents three reward aggregation techniques that reflect operational modes typical in home health care.

The primary objective is visit fulfillment, defined as the proportion of requested visits that are scheduled. Secondary objectives are travel efficiency, workload equity, continuity of care, idle time, schedule density and preference satisfaction. We compare the DRL model with exact optimization, heuristic, and metaheuristic benchmarks to examine how it balances these competing priorities.

1.8 Research Limitations

The results should be read with the following limitations in mind. The study does not include dynamic or stochastic factors such as real-time rescheduling, varying travel times, or unexpected patient cancellations. We use static data to isolate and test the DRL model's effectiveness and scalability. To that end, the findings may not reflect the uncertainty present in day-to-day home health operations.

The problem formulation assumes a single centralized scheduler. However, in practice, some agencies delegate scheduling to clinicians, share scheduling responsibility between schedulers and clinicians, or divide scheduling across teams.

While the data are synthetic and represent a typical home health agency, important nuances are missing. These include clinician preferences, changes in patient condition, emergency visits, inter-visit dependencies, and home health agency's policies. Field testing is therefore needed to confirm performance in practice.

Finally, results depend on the choice of reward weights. This Praxis uses visit fulfillment as the primary objective. Organizations with different priorities (e.g., workload balance or retention) would likely require different weights. The model also assumes a fixed set of hard constraints, so performance may differ when assumptions differ.

1.9 Organization of Praxis

Chapter 2 examines the literature on home health care scheduling, classical optimization approaches and metaheuristic methods. We identify limitations in existing approaches then discuss reinforcement learning methods that motivate our praxis.

In chapter 3, we introduce reinforcement learning, define the dataset, discuss DRL model, PPO and the challenges in training reinforcement learning models.

Chapter 4 presents results across three research questions. We report visit fulfill-

ment rates comparing the DRL model against exact optimization, metaheuristic, and constructive benchmarks. We analyze Pareto frontier quality through hypervolume and set coverage metrics. Lastly, we evaluate the DRL model against a comprehensive metric.

In Chapter 5, we summarize findings, practical applications, acknowledge the imitations of our work and suggest future research.

Chapter 2 – Literature Review

In this chapter, we look at the limitations of classical and metaheuristic approaches to the HHCRSP and how deep reinforcement learning could improve on these methods.

2.1 Classical Optimization Approaches to HHCRSP

Work by Rasmussen et al. (2012) modeled HHCRSP as a set partitioning problem with side constraints. They solved the problem using branch-and-price with time and window based branching by controlling soft clinician-patient preferences and synchronizing constraints.

Similarly, Assaf et al. (2024) applied Choco Solver to solve the medical appointment problem. They were able to integrate workload fairness as an optimization objective. Their work demonstrated that fairness constraints can be encoded within a framework. However, the model remained static and was unable to adapt or learn from feedback.

Both methods produced strong results, though the results were limited because optimization was performed sequentially. Coverage was first maximized, then secondary goals were then handled later.

2.2 Evolution Toward Heuristics and Metaheuristics

Metaheuristics methods emerged as a way to overcome the computational cost of running exact optimization methods. These methods offered faster solutions than exact methods. Heuristics methods build schedules sequentially using rules like nearest neighbor insertion. On the other hand, metaheuristics can improve solutions through neighborhood search, evolutionary operators, or swarm intelligence.

2.2.1 Multiobjective with fairness.

Fu et al. (2024) developed a multi-objective artificial bee colony algorithm that incorporated domain knowledge from the home health care industry. Ala et al. (2022) introduced fuzzy logic with metaheuristic search to model workload fairness. They used membership functions to describe clinician satisfaction and to balance assignments. Both techniques showed that fairness can be encoded within metaheuristic frameworks but it requires careful parameter tuning.

2.2.2 Dynamic insertion and real time adaptation.

Bennett and Erera (2011) proposed insertion heuristics to provide dynamic routing. This approach addressed real time patient requests through myopic strategies. Their work revealed that greedy insertion minimizes immediate travel cost but often produces fragmented schedules that accumulate inefficiency over time. This myopic behavior is a motivation for reinforcement learning based methods that balance short term feasibility with long term optimization.

2.2.3 Multiobjective with Continuity.

Clapper et al. (2024) introduced a technique that balanced continuity of care with route efficiency. Clapper et al. encoded continuity using a Continuity of Care Index. They then integrated visit frequency, relational exclusivity, and scheduling consistency. This work demonstrated that continuity can be used as a valid metric that can be optimized.

Metaheuristics are limited in the following ways. First, they only optimize a single objective. Second, they require manual parameter tuning which limits generalization across different problem sizes And Lastly, they produce static policies that cannot adapt to feedback or learn from experience.

2.3 Introduction of Learning-Based Methods

Reinforcement learning techniques address these limitations through three capabilities. Learning through interaction enables adaptive policies that improve from experience. Decomposed reward structures allow multiple objectives to be optimized simultaneously so that an agent can discover which trade-offs matter rather than optimizing a single objective at a time. And function approximation through neural networks supports generalization across problem instances.

2.3.1 Foundations of Reinforcement Learning for Routing and Healthcare

Nazari et al. (2018) proposed a DRL framework for the Capacitated Vehicle Routing Problem using an attention based decoder and policy gradient training. Their architecture removed the encoder RNN to reduce computation costs. The model generalized and consistently outperformed classical heuristics.

Schulman et al. (2017) introduced Proximal Policy Optimization (PPO), which has become foundational for policy learning in complex domains. PPO stabilizes policy updates through a clipped surrogate objective that prevents excessively large policy changes, enabling robust learning under partial observability and sparse rewards. Its compatibility with continuous and discrete action spaces, entropy regularization, and general advantage estimation makes it well-suited for multiobjective scheduling environments.

Ta-Dinh et al. (2024) developed a reinforcement learning approach for online dynamic home health care scheduling that modeled real-time patient acceptance and routing decisions as a Markov Decision Process. They trained a Double Deep Q-Network (DDQN) agent for sub-millisecond decision-making, demonstrating feasibility for real-world dispatch integration. Their work validates that learning-based methods can achieve the inference speeds required for operational deployment. How-

ever, their focus on acceptance decisions differs from this Praxis's emphasis on comprehensive multiobjective scheduling.

2.3.2 Handling Stochasticity and Uncertainty in Dynamic Environments

In home health care, conditions can change quickly. For example, patients can cancel services, clinicians can call off or weather conditions can lead to delayed services. Reinforcement learning offers adaptive policies that respond to these uncertainties by learning from feedback.

Zhang et al. (2023) developed a home health care routing and scheduling model that incorporates this stochasticity. Their hybrid system integrated Q-learning with Ant Colony Optimization to demonstrate that stochasticity can be integrated within reinforcement learning.

Salbanya et al. (2023) applied reinforcement learning to daily caregiver assignment in a home health setting in Barcelona. They formulated the task as Markov decision process and demonstrated that reinforcement learning can cope with uncertainty.

2.3.3 Zero-Shot Generalization in Deep Reinforcement Learning

Kirk et al. (2023) introduce zero shot generalization. The idea is that a policy trained on instances sampled from an environment family should perform well on unseen instances from the same family without retraining. In this Praxis, we test whether policies trained on one set of instances generalize to held out instances from the same environment family.

2.3.4 Reward Shaping, Efficiency, and Sample Complexity

In their article, Gupta et al. (2022) demonstrate that reward shaping can simplify model complexity. Their finding shows that appropriately designed shaping functions can speed up convergence without inducing bias in the optimal policy. We use this

finding to decompose our reward function..

Devidze et al. (2022) developed a reward shaping approach that adjusts reward signals in the environment based on the agent's exploration history in relation to sample efficiency in sparse-reward environments. Sun et al. (2022) proposed a reward-shifting approach that stabilizes learning in environments with delayed reward feedback. These techniques indicate that modular, decomposed reward functions can guide effective learning while reducing sample complexity. We use this approach in this Praxis to separate fulfillment, travel performance, workload fairness, continuity, and preference satisfaction (Chapter 3.4).

2.3.5 Fairness-Aware Reinforcement Learning in Scheduling

Hermelin et al. (2025) provide a foundation for fairness. This framework measures fairness using max-min fairness and proportional fairness. We adopt this approach in our workload equity reward. (Section 3.7 and Section 4.4).

In their paper, Yu et al. (2023) introduce fairness in reinforcement learning to build inclusion within e-learning agents. They illustrated that one can learn fairness objectives during training, rather than enforcing fairness through post-processing. The design in this praxis treats workload equity as a relevant objective. (Chapter 3.7).

2.3.6 Addressing Overestimation and Stability Challenges

Ma et al. (2021) propose the Feasible Actor-Critic (FAC) algorithm. Rather than aiming to maximize average constraint satisfaction, FAC guarantees that the safety criterion is met at every state. In the home health care setting, this means, for example, that an infeasible schedule for just one patient can result in moderate to severe harm. Although we do not use FAC in this Praxis, its underlying principles, such as constraint satisfaction, drive action-masking that we implement in Section (Chapter 3.2). Here, action masking guarantees that hard constraints are satisfied at

all times.

Han et al. (2020) introduced an actor-critic framework that utilizes control theory ideas to provide stability of the system during learning. These systems encourage equitable work distribution and promote balanced service delivery across the service area.

2.3.7 Robustness under Real-World Environmental Disruptions

Dessevre et al. (2022) examined sensitivity of home health care schedules to changes in the environmental. They modeled disruptions in clinicians' arrival time, travel durations and service time. Small deviations in initial parameters led to substantial increases in both early and late arrivals to the patients' homes. These finding suggest that rigid schedules can deteriorate rapidly in practice and reinforce the need for adaptable learning systems.

According to Wu et al. (2025), policy gradient methods, such as PPO, are increasingly used in health care because they are robust, can handle multiobjective reward functions, and can apply domain constraints with flexibility. This further validates our use of PPO in this Praxis.

2.4 Unaddressed Gaps in the Literature

Despite some recent momentum to apply reinforcement learning to the HHCRSP, four gaps exist in the literature.

Gap 1: Reward functions do not encode ethical and relational factors. Studies by Gupta et al. (2022), Devidze et al. (2022), and Sun et al. (2022) demonstrate the benefits of reward shaping and the acceleration of learning through exploration. However, few of the proposed models encode continuity, fairness, or preference as prioritized objectives. Early works also treat ethical objectives as soft constraints and not part of the reward objective.

Gap 2: Constraint satisfaction mechanisms remain underdeveloped in learning-based scheduling. Most reinforcement learning applications to routing rely on penalty-based rewards to discourage infeasible actions, contaminating the learning signal and requiring careful penalty tuning (Nazari et al., 2018; Pan and Liu, 2023). Action masking, which guarantees constraint satisfaction by restricting the agent's action space to feasible assignments, has been explored in robotics and game playing but remains underutilized in healthcare scheduling. Penalty based methods force the agent to learn what not to do through trial and error, while action masking ensures that the agent never considers infeasible actions, accelerates learning and guarantees valid results.

Gap 3: Generalization to unseen problem instances remains limited. Most deep reinforcement learning models are trained and evaluated within fixed topologies or task structures. While Kirk et al. (2023) and Korkmaz (2024) provide formal frameworks for analyzing generalization, few healthcare scheduling models test instance level generalization. They don't test whether policies trained on one set of patient demographics, geographic distributions, and constraint patterns can perform competitively on held out instances with different characteristics. This praxis addresses instance level generalization.(Chapter 3.6). However, zero-shot transfer remains an open challenge for future work.

Gap 4: Continuity of care is rarely treated as a primary design variable. Despite empirical evidence linking continuity to reduced mortality (Pereira Gray et al., 2018), many reviewed studies fail to encode continuity as a structured reward dimension or long-term constraint. While some frameworks acknowledge preference alignment or historical pairings (Salbanya et al., 2023; Zhang et al., 2023), few models optimize continuity simultaneously with coverage and efficiency. Clapper et al. (2024) demonstrated that explicit continuity metrics enable 26% improvement with only 2% cost increase, showing that continuity can be formalized and optimized.

These four gaps motivate the architectural decisions presented in Chapter 3. Specifically, decomposed reward functions that encode ethical objectives as design priorities, action masking that guarantees constraint satisfaction without reliance on penalty based learning, and modular aggregation that enables systematic Pareto frontier exploration. The methodology addresses the research questions posed in Chapter 1.6.1, with results tested empirically in Chapter 4.

Chapter 3 – Methodology

At its core, reinforcement learning (RL) consists of an agent, actions, states, a policy, a reward function, and the environment. The agent observes the state of the environment and selects an action based on its policy. The agent receives a reward indicating the value of the decision made. The environment then transitions to a new state and the process repeats.

Through many such interactions, the agent learns a policy that maps states to actions in order to maximize the cumulative reward. This learning process is accomplished through trial and error, making RL ideal for problems where the optimal behavior is unknown or difficult to specify in advance.

Home health care scheduling is a natural fit for RL. To explore this, we model the HHCRSP as a Markov Decision Process (MDP) implemented through Proximal Policy Optimization (PPO).

At each step, the agent observes the current scheduling state, including pending visits, clinician availability, clinician qualification, workload hours, traffic and other information, and then assigns a clinician to a patient. The environment returns a reward signal that reflects the quality of the decision before transitioning to the next state. Through repeated interactions, the agent learns which assignment patterns maximize the reward objective.

3.1 Dataset

The evaluation uses the Mendeley benchmark dataset developed at Polytechnique Montréal. It includes 60 synthetic cases that model home healthcare operations. These cases are grouped into three categories that reflect how real agencies operate. Each one covers a full week of scheduling with decisions made in 30-minute steps.

- **Small**: 40 patients, 120 visits, 5 clinicians, 25 workdays

- **Medium**: 80 patients, 225 visits, 10 clinicians, 45 workdays
- **Large**: 150 patients, 430 visits, 20 clinicians, 90 workdays

Small cases represent situations where there are enough clinicians to explore trade-offs between the primary objective and secondary objectives. These scenarios test whether the DRL policy can balance multiple goals when resources are plentiful.

Medium cases introduce scenarios where there is a fragile line between number of requested visits and the number of qualified and available clinicians.

Large cases simulate scenarios where the number of requested visits exceeds the number of qualified and available clinicians. These instances test how the model behaves under clinician scarcity.

Each case describes patient and clinician details in JSON files. Patient records include the number of required visits, service duration, time windows, required and optional skills, and location IDs. Clinician records specify skill sets, daily and weekly work limits, availability windows, and home locations. Continuity scores from 2 to 4 represent the strength of past clinician-patient relationships.

The dataset also provides additional routing data. The first dataset is a distance-speed-duration file that defines travel costs between all locations. Second is a location-to-zone map that supports zone-based travel modeling. Finally, time-indexed congestion multipliers that adjust travel times at 30-minute intervals throughout the week.

Figure 3.1: Partial Patient Data


```
{
    "0":{
        "location":38,
        "number_visits":2,
        "service_duration":54,
        "available_days": [1,5],
        "time_window_per_available_day": [602,764,847,1009],
        "optional_expertises":["Expertise1"],
        "mandatory_expertises":["Expertise2"]
    },
    "1":{
        "location":8,
        "number_visits":2,
        "service_duration":54,
        "available_days": [1,2,5,6],
        "time_window_per_available_day": [886,994,822,984,575,737,673,889],
        "optional_expertises":[],
        "mandatory_expertises":["Expertise4"]
    },
    "2":{
        "location":139,
        "number_visits":5,
        "service_duration":44,
        "available_days": [1,2,4,5,6],
        "time_window_per_available_day": [720,896,643,819,804,892,863,995,630,762],
        "optional_expertises":["Expertise4","Expertise5"],
        "mandatory_expertises":["Expertise2","Expertise1"]
    }
```


Figure 3.2: Partial Clinician Data


```
{
    "0":{
        "location":74,
        "expertises":["Expertise2","Expertise1","Expertise3"],
        "min_work_time_week":441,
        "max_work_time_week":2103,
        "work_days": [1,2,3,4],
        "time_window_per_work_day": [739,1324,712,1172,712,1339,664,1174],
        "min_max_work_time_per_work_day": [11,479,101,446,12,507,86,428]
    },
    "1":{
        "location":54,
        "expertises":["Expertise1","Expertise5","Expertise2"],
        "min_work_time_week":220,
        "max_work_time_week":1621,
        "work_days": [1,2,3,5,6],
        "time_window_per_work_day": [520,1149,491,1151,592,1057,643,1340,654,1188],
        "min_max_work_time_per_work_day": [88,559,138,633,4,465,118,613,48,534]
    },
    "2":{
        "location":98,
        "expertises":["Expertise1","Expertise4"],
        "min_work_time_week":129,
        "max_work_time_week":1774,
        "work_days": [0,2,3,5,6],
        "time_window_per_work_day": [679,1386,486,1067,673,1156,574,1170,601,1244],
        "min_max_work_time_per_work_day": [169,615,104,511,91,386,154,512,6,520]
    },
```

Figure 3.3: Partial Continuity Data

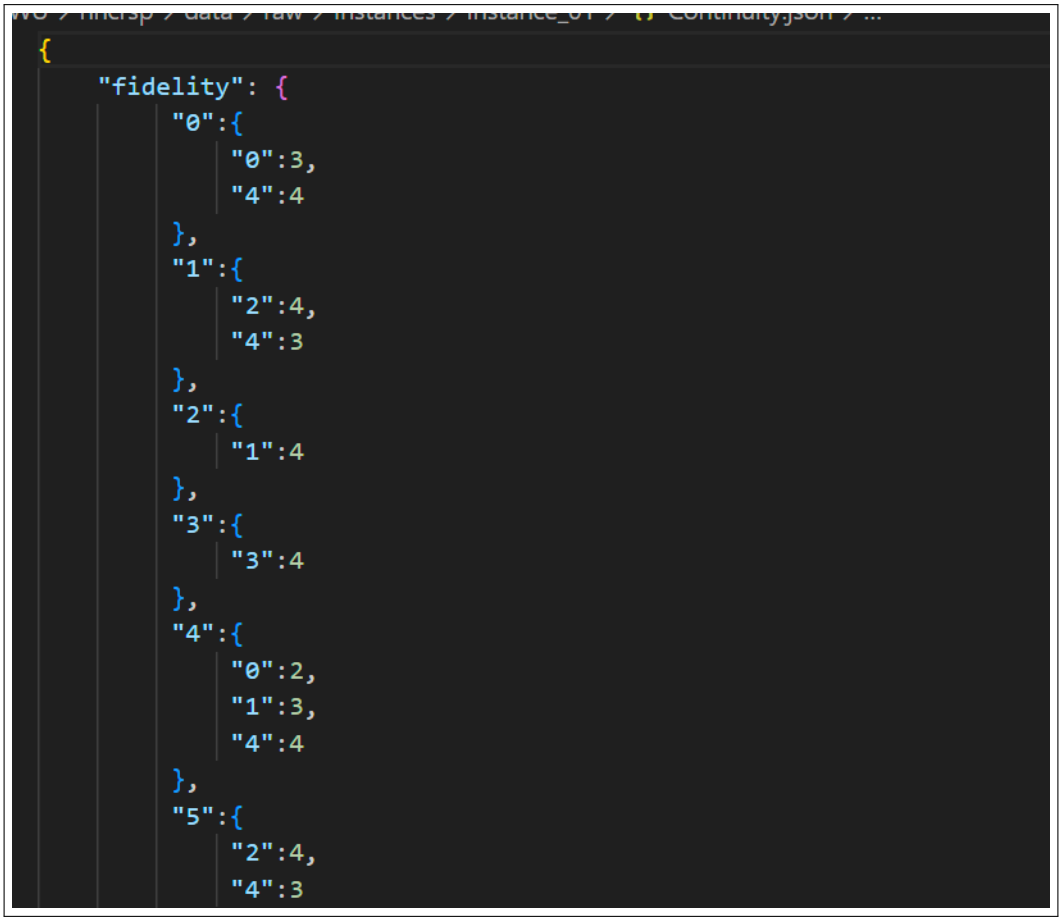


Figure 3.4: Partial Speed Data

```
Nb_Locations : 141

from,to,distance,speed,duration
0,0,0,0,0,
0,1,7.595,0.614151,12.3667,
0,2,7.181,0.579113,12.4,
0,3,9.161,0.777454,11.7833,
0,4,26.546,1.11226,23.8667,
0,5,11.603,0.755896,15.35,
0,6,10.11,0.789844,12.8,
0,7,11.876,0.750063,15.8333,
0,8,12.648,0.750623,16.85,
0,9,9.638,0.682739,14.1167,
0,10,28.047,1.19603,23.45,
0,11,15.478,0.8786,17.6167,
0,12,6.64,0.592857,11.2,
0,13,11.486,0.639295,17.9667,
0,14,10.891,0.644438,16.9,
0,15,12.969,0.847647,15.3,
0,16,50.746,1.28579,39.4667,
0,17,8.198,0.778291,10.5333,
0,18,10.382,0.6727,15.4333,
0,19,11.067,0.627618,17.6333,
0,20,11.925,0.750787,15.8833,
0,21,10.151,0.666368,15.2333,
0,22,43.475,1.26258,34.4333,
0,23,7.027,0.685561,10.25,
0,24,11.029,0.647495,17.0333,
0,25,13.742,0.821235,16.7333,
```

Figure 3.5: Partial Traffic Data

```
Nb_Zones : 14
Nb_section : 336

from_zone,to_zone,id_section,traffic_index
0,0,0,1
0,0,1,1
0,0,2,1
0,0,3,1
0,0,4,1
0,0,5,1
0,0,6,1
0,0,7,1
0,0,8,1
0,0,9,1
0,0,10,1
0,0,11,1
0,0,12,0.95
0,0,13,0.9
0,0,14,0.85
0,0,15,0.8
0,0,16,0.75
0,0,17,0.75
0,0,18,0.8
0,0,19,0.85
0,0,20,0.95
```

3.2 Constraint Derivation and Feasibility Mapping

The environment enforces six hard constraints to keep the schedules feasible. Skill matching ensures that each clinician has the mandatory qualifications required by a patient. Availability matching checks that the clinician working hours overlap with the patient's requested time window. Work budget limits how many minutes a clinician can be scheduled per day and per week. Time window prevent assignments from overlapping for the same clinician. Travel feasibility checks that clinicians can make it

to the patient on time given current traffic conditions and time of day. Finally, unique assignment guarantees that each visit is completed once with no double-booking.

At every decision step t, the environment builds a binary feasibility matrix $\mathbf{F}_t \in \{0,1\}^{C \times V_t}$. Here, C is the number of clinicians and V_t is the number of visits that can be assigned at timestep t. Each entry $\mathbf{F}_t[i,j]$ stores the outcome of all hard constraint checks for assigning clinician i to visit j.

In other words:

1. The visit will occur within the window requested by the patient.
2. The clinician has the skills required to treat the patient.
3. Assigning the visit will not exceed the clinician's daily or weekly limits.
4. The visit has not been assigned to another clinician.
5. The clinician can get to the patient on time from their current (or prior visit) location.
6. Traffic conditions at that time of day will still allow the clinician to arrive on time.

A value of 1 means the assignment is feasible; a value of 0 means at least one constraint is violated.

The PPO policy produces raw action logits $\mathbf{Z}_t \in \mathbb{R}^{C \times V_t}$. To make sure the PPO policy never selects an action that is not feasible, we mask these logits before turning them into probabilities. We do this by performing an elementwise addition as shown below before applying a softmax.

$$\pi'(a_t \mid s_t) \;=\; \mathrm{Softmax}\Big(\mathbf{Z}_t + \log \mathbf{F}_t\Big). \tag{3.1}$$

where $\pi'(a_t \mid s_t)$ is the masked policy at time t, a_t is the action taken at time t, and s_t is the state at time t.

The mask sets the probability of visits that are not feasible to zero probability, and softmax turns the remaining scores into a probability distribution. This has the effect of trimming the action space to only the visits that are feasible. The agent can only take valid actions, which guarantees that treatment, legal and ethical requirements are met and not left to chance. It also improves sample efficiency because exploration now stays within the feasible region. In addition, it stabilizes training because PPO's updates are not wasted on impossible moves. Lastly, by shrinking the action space at each step, the design helps credit assignment and speeds up convergence.

3.3 Feature Encoding for Policy Learning

In Section 3.2 we described how constraints define which actions are allowed. In this section we focus on how the scheduling state is represented in DRL network. Each feature gives the agent information needed to make reasonable decisions within the feasible actions.

Skill features describe which qualifications each clinician has and which each patient needs. By storing them as binary vectors, matching skills with patient requirements becomes transparent and exact. Additional values capture how many skills a clinician can cover or how closely their skills match the patient's requirements.

Temporal features represent how urgent each visit is. They include the remaining time in a patient's window, the time of day, the day of the week, and how far along the schedule is in the current horizon. These features help the agent recognize when it should prioritize visits that are close to expiring or delay those with more flexibility.

Workload features track how busy each clinician is. They record used and remaining daily or weekly work minutes and the overall progress toward fulfilling all visits. This information supports fair scheduling, preventing both overload and underuse.

Spatial features describe where clinicians and patients are located. Locations are embedded in a low-dimensional space that preserves real travel distances. The

environment updates clinician positions as they are assigned. These signals help the policy learn how to make travel efficient scheduling decisions.

Relational features capture the human elements of scheduling. They include continuity scores that reflect past clinician-patient relationships and patient preferences. These features help the policy maintain consistent care when possible and respect assignment preferences.

All continuous features are normalized to stable ranges so that training remains consistent. Time-related values use the minutes in a day as a base, travel times are clipped by global percentiles, and utilization measures stay within the range of zero to one. This scaling prevents unstable gradients and helps the model converge smoothly.

3.4 Reward Function Design

As the agent interacts with the environment, it learns to take actions that maximize the reward. However, in a multiobjective model, the objectives can conflict with each other such that taking one action that improves the reward can lead to another action that reduces it. For example, a clinician may be located closer to several patients and therefore can be assigned to the patients. This approach would maximize visit fulfillment but it would lead to over exposure of one clinician. It could lead to resignation of the overworked clinician, resignation of the under-worked clinician(due to no work), patient dissatisfaction, violations of labor laws and so on.

The reward function must therefore balance these secondary objectives and not just focus on a single objective, however important.

We define the multiobjective reward vector as

$$\mathbf{c}_t = \Big[c_1(t),\, c_2(t),\, \ldots,\, c_K(t)\Big]^\top, \tag{3.2}$$

where each component $c_k(t)$ corresponds to objectives under consideration e.g. visit

fulfillment, travel and so on.

An aggregation function then maps this vector to a single scalar reward

$$r_t \;=\; A\Big(\mathbf{c}_t;\, \theta_A\Big), \tag{3.3}$$

where $A(\cdot)$ is the chosen aggregator (e.g., Weighted Sum, Chebyshev, Policy-Aware), and θ_A denotes its parameters (e.g., objective weights $\{w_k\}$).

To account for different scheduling modes in home health care, we implement three aggregator types:

3.4.1 Weighted Sum Aggregator.

This method takes a simple linear combination of components using weights:

$$r_t^{ws} = \sum_{k=1}^{K} \alpha_k c_k(t) \tag{3.4}$$

Each $\alpha_k > 0$ defines the importance of objective k. Larger weights suggest the relative importance of that objective. This aggregator is applicable in cases where a home health agency has clear priorities. For example, a home health agency that prioritizes visit fulfillment with slight consideration on idle time might set $r_{\text{fulfillment}}$ to be way larger than r_{idle}

The weights can also be tuned during training depending on operational needs of the home health agency.

3.4.2 Chebyshev Aggregator.

This aggregator improves whichever objective is currently worse, prevents the model from neglecting objectives and yields a balanced outcome.

$$r_t^{cheb} = -\max_k\{w_k|c_k^* - c_k(t)|\} - \rho\sum_{k=1}^{K} w_k|c_k^* - c_k(t)| \tag{3.5}$$

Here, c_k^* is the target value for objective k, w_k are relative weights of the objectives, and ρ is a small smoothing factor.

An example use case for this aggregator is a scenario in which a home health agency wants to maintain high visit fulfillment without overloading a single clinician. In this scenario they might set a minimum target on r_{fair}.

3.4.3 Policy-Aware Aggregator.

This method enforces performance thresholds by applying penalties when required conditions are not met:

$$r_t^{pa} = \begin{cases} -\kappa, & \text{if } G(\mathbf{c}_{\leq t}; \tau) \text{ violated} \\ \sum_{k=1}^{K} \alpha_k(t) c_k(t), & \text{otherwise} \end{cases} \tag{3.6}$$

If a constraint $G(\mathbf{c}_{\leq t}; \tau)$ is violated, a large penalty $\kappa \gg 0$ is applied. Otherwise, rewards are aggregated normally using time-varying weights $\alpha_k(t)$.

An example use case of this aggregator would be to implement a hard penalty if the daily fulfillment falls below a required threshold, otherwise optimize travel and workload equity. A home health agency could also increase travel time during a storm or otherwise enforce minium visit fulfillment.

In this Praxis, we designed the environment such that model could accept any of the above aggregators without altering the mechanics of the environment. This modular approach allowed us to isolate the effects of aggregation on DRL performance.

3.5 Reward Components

Reward components are the individual objectives that the DRL aims to maximize or balance.

3.5.1 Fulfillment Reward.

This component provides positive feedback when a clinician is assigned to a required visit:

$$r_{fulfill}(t) = 1 + f_j(t)(\mu_{j,t} + \sigma_j) \tag{3.7}$$

The fulfillment reward provides a base value of 1 whenever a valid visit is scheduled. The reward also includes an additional term that increases with the visit's current urgency $\mu_{j,t}$ and the patient's priority σ_j. This means that urgent visits or high priority patients are favored. This bonus is also scaled by the fraction of the patient's visit that have already been completed $f_j(t)$. As patient approaches full coverage each subsequent visit contributes a larger bonus.

3.5.2 Travel Efficiency Penalty.

This component discourages long or inefficient travel routes:

$$r_{travel}(t) = -\alpha_{travel} \cdot \left(\frac{\tau_{ij}^{t} - \tau_{min}}{\tau_{max} - \tau_{min}} \right)^2 \tag{3.8}$$

The travel term subtracts reward in proportion to how long the trip is. Short trips incur only a small penalty. On the other hand, very long trips are punished more because the term is *squared*, which makes the penalty grow faster as travel time increases. The factor α_{travel} sets how much the policy should care about travel relative to other goals. The agent is therefore encouraged toward nearby assignments and compact routes, while still allowing occasional longer trips when they are justified by other objectives.

3.5.3 Workload Fairness Reward.

The fairness term looks at how evenly work is shared across clinicians and rewards changes that make it more balanced.

$$r_{fair}(t) = -\alpha_{fair} \cdot (\text{Gini}(\mathbf{u}_t) - \text{Gini}(\mathbf{u}_{t-1})) \tag{3.9}$$

where $\mathbf{u}_t$ is the number of assignments already scheduled to each clinician and Gini($\mathbf{u}$) is the equality factor such that 0 means perfectly even while larger values indicate more unevenness. This signal is zero-centered which means that if inequality decreases after the assignment, the difference is negative and $r_{\text{fair}}(t)$ is positive, resulting in a bonus for improving balance. On the other hand, if inequality increases, the difference is positive and the term becomes negative, resulting in a penalty for making workloads more uneven. Lastly, if inequality is unchanged, the term is zero.

A zero-centered design is helpful for several reasons. It avoids bias when the system is already balanced, since a steady state contributes roughly zero and does not nudge the policy to make unnecessary changes. It also gives local informative credit by rewarding small improvements when they occur. This helps the learner to connect each action to its effect. Keeping the signal centered around zero helps with PPO stability by moderating reward scale and variance. Because there is no constant payoff for being fair, the policy cannot gain by shuffling work without real improvement. Finally, the signal is not affected by scale, and plays well with other objectives. It does not require a fixed global target for inequality because it treats any improvement as positive and any decline as negative, and its overall influence can be tuned through α_{fair} without overwhelming other objectives.

3.5.4 Continuity Reward.

This term encourages consistent clinician-patient relationships:

The continuity reward favors assigning a patient to a clinician they already know, with extra emphasis when that relationship is recent.

$$r_{cont}(t) = \alpha_{cont} \cdot C_{ij} \cdot R_{ij} \tag{3.10}$$

where C_{ij} a number between 0 and 1 that indicates how strong the working relationship is between clinician i and patient j. The value of C_{ij} is higher when that clinician has seen the patient more often or more consistently in the past, and lower when the pairing is new or infrequent. R_{ij} is a number between 0 and 1 that reflects how recently clinician i was assigned to patient j. It's close to 1 if they saw each other very recently, and it fades toward 0 as more time passes without a visit. α_{cont} is a positive scaling constant that sets how much the continuity reward should matter compared with other objectives. Larger values reward consistent and recent clinician-patient pairings. This product means the signal is strongest when the pairing has both a solid history i.e. higher interactions (C_{ij} is high) and and recent visits (R_{ij} is high). If either is weak, the reward shrinks. However, as new assignments are made to a different clinician, if consistent, the new clinician starts to build Continuity rewards while the former clinician's rewards start to decay.

3.5.5 Preference Satisfaction Reward.

The preference reward gives credit when the assigned clinician matches a patient's optional skill preferences e.g. language or specialty. It measures the overlap between the clinician's skills and the patient's preferred skills:

$$r_{\text{pref}}(t) \;=\; \alpha_{\text{pref}} \cdot \frac{|\text{Skills}_i \cap \text{Preferred}_j|}{|\text{Preferred}_j|}, \tag{3.11}$$

where Skills_i is the set of optional skills offered by clinician i and Preferred_j is the set of optional skill preferences for patient j. If a patient has no optional preferences, i.e.

denominator $= 0$, $r_{\text{pref}}(t) = 0$.

The fraction represents the portion of the patient's preferences that are satisfied. A value of 1 means all preferences are met, values between 0 and 1 reflect partial matches, and 0 means none are met. The coefficient α_{pref} sets how much this consideration matters relative to the other objectives. Because the score is proportional, partial matches still earn meaningful credit, which is useful when a perfect match is not feasible due to other constrains. For example, if a patient prefers $\{\text{Spanish}, \text{Wound Care}, \text{IV Therapy}\}$ and a clinician offers $\{\text{Spanish}, \text{IV Therapy}, \text{Pediatrics}\}$, the overlap covers two of three preferences, leading to a reward of $\alpha_{\text{pref}} \times 2/3$.

3.6 Addressing Generalization and Credit Assignment

3.6.1 Exploration and Exploitation.

Scheduling tasks often create an imbalance between short-term rewards and long-term benefits. If the agent converges too early, it may settle for suboptimal local patterns instead of discovering better global strategies. To avoid this, entropy regularization maintains randomness in the agent's decisions during training. High entropy at the start encourages broad exploration of possible assignments. As training progresses, entropy weights decay gradually, allowing the policy to focus on exploiting the most rewarding strategies it has learned.

3.6.2 Credit Assignment in Long Horizons.

Many decisions in home health care scheduling show their impact only much later in the planning horizon. For example, keeping a flexible clinician available early in the week may enable successful completion of a difficult visit hundreds of timesteps later. These delayed effects make it difficult to assign credit to earlier actions. Generalized Advantage Estimation (GAE) helps address this by averaging temporal differences across time using exponentially weighted smoothing. An appropriate choice of λ

balances bias and variance. Larger λ propagates rewards farther back in time while smaller λ increases bootstrapping but shortens the credit horizon.

3.6.3 Function Approximation Stability.

Deep neural networks can become unstable in environments with large and dynamically constrained action spaces. When targets shift or features vary widely, gradients may explode or updates may overshoot. To prevent this, we use gradient clipping to limit the size of updates, we normalize observations to keep inputs within consistent ranges, and implement dropout to prevent over-reliance on specific feature patterns. Logit masking guarantees that hard feasibility constraints are differentiable.

3.6.4 Generalization Across Instances.

The model must perform well on new, unseen combinations of patients, clinicians, and locations, rather than memorizing those seen in training. To achieve this, first, during training we sample randomized scheduling instances with different distributions of patients, workloads, and geography. This forces the policy to learn transferable principles instead of fixed patterns. Second, dropout regularization improves robustness by randomly masking parts of the network during training thus preventing overfitting to instance-specific details.

3.7 Ethics-by-Design

The reward design integrates workload equity, relational continuity, and preference satisfaction as part of the DRL optimization.

3.7.1 Workload Equity.

The fairness component (r_{fair}) promotes even distribution of assignments among clinicians. It rewards actions that reduce imbalance and penalizes those that increase it.

This reward signal helps the policy avoid exploitative scheduling patterns where some clinicians face excessive workloads while others remain underused. By encouraging balanced utilization, the policy supports staff well-being and helps reduce burnout and turnover.

3.7.2 Relational Continuity.

The continuity component (r_{cont}) preserves long-term clinician-patient relationships by rewarding assignments that maintain consistency. Continuity is not only operationally desirable but also clinically and ethically important, as it improves patient outcomes and enhances job satisfaction for clinicians.

3.7.3 Preference Satisfaction.

The preference component (r_{pref}) honors patient preferences leading to better outcomes and patient satisfaction.

The modular reward structure allows organizations to adjust ethical priorities based on their needs. For example, an agency facing severe staffing shortages might reduce fulfillment weighting to favor workload balance to improve clinician retention. Others serving vulnerable populations may increase continuity weighting to guarantee consistent care relationships.

3.8 PPO Protocol

PPO is a policy gradient method that updates how an agent makes decisions through controlled optimization steps. One of the main issues in policy gradient methods is that large updates can destabilize learning, causing the agent to suddenly perform worse than before and undoing prior progress. PPO alleviates this problem through a clipping mechanism, which limits how much the action policy can change in one update. The algorithm interacts with an environment to collect batches of experience

and performs optimization over multiple epochs. During each update, the new policy is compared to the old policy, and once the policy ratio is above a defined threshold, the new and old policy are clipped. This allows updates to be meaningful without being destructive. Clipping is especially useful for maintaining stability in large action spaces.

There are multiple reasons why PPO is ideal for home health care scheduling. The first reason is that the home health scheduling problem is a sequential decision over a long horizon. Long episodes mean that assigning visits poorly early in the week may lead to infeasible visits later in the week. Because PPO is stable, it won't allow the policy to make drastic updates that break patterns it has learned. Second, the action space changes dynamically as clinicians become unavailable and visits are assigned to patients. PPO handles this through the masked policy approach described in Section 3.2. It maintains stable updates even as the feasible visits change. Third, PPO has controlled update steps which allow the agent to balance multiple objectives in this problem. Finally, PPO requires minimal hyperparameter tuning and it performs well even on large problem sizes.

3.9 Environment Simulation

The scheduling simulation has a weekly horizon with 30-minute time intervals. At each time interval, the environment tracks the information necessary to make scheduling decisions, including which visits need to be scheduled, the location of each clinician, the number of hours each clinician has worked as well as patient/clinician assignment history.

The decision process functions in a loop. The agent receives the current schedule state. This includes a list of valid assignments that represents each clinician's expertise and current location to determine which clinician can be assigned to which visit without violating any constraints.

After selecting an assignment from the list, the agent makes necessary updates in the clinician's schedule, such as the number of visits assigned, number of hours worked per day and per week, and continuity score. It also makes updates on the patient's profile including visits completed and visits remaining.

The selected visit is then recorded as complete and the clock advances to account for service duration plus travel time to reach the patient. Travel time considers traffic, location, direction of travel and time of day. Once all the updates are made, the process continues.

The simulation ends after one of three conditions occur. First, all required visits are scheduled. Second, the week is completed. Third, there are no longer any valid assignments.

Each assignment will alter the state, making it observable and predictable. If a clinician is assigned to a visit, there will be one less available work hour and their work count will have one additional visit. Their location will change to the patient location address, and the visited patient's subsequent visit will be marked fulfilled. Decisions are made using information only observable in the assignment, and the agent never sees other visits in the future nor any constraints that have not yet transpired. Therefore, decisions only occur using available information at that time.

Throughout the day, travel times will vary according to the traffic patterns. The simulation accounts for a zone-based congestion multiplier to modify travel time according to time of day, direction of travel, and location. This multiplier behaves the same way during the assignment validation check where the rewards are calculated and clinicians' location is updated.

The environment documents information about the assignment after each selection. This includes who was assigned to each patient, at what time, and a snapshot of what the overall schedule looks like. At the end of the week, performance metrics such as coverage, travel distance, workload balance, and continuity are calculated

based on the assignment records.

3.10 Training

The dataset is divided into training, validation and testing sets in a 60/20/20 split. The split is done separately for each type of instance i.e., small, medium, and large instances. A random seed is fixed and set to provide consistent and reproducible results. Before training begins, each of the dataset's splits is shuffled to prevent the model from learning patterns.

The training process uses Optuna to sweep through several different hyperparameter configurations. Optuna tests several different learning rates, neural network architectures, and other configurations. The Optuna also sweeps different parameters that determine how the agent explores random assignments versus exploitation of what was previously learned.

The best performing hyperparameter configuration is used for validation.

Validation includes a sensitivity test on the best model by using four different random seeds to determine whether the strong performance is due to a random and lucky initialization. If the model performs well across all four random seeds, then the model is determined to have learned generalizable patterns in the training phase.

Testing is performed on all methods, DRL, CP-SAT, Nearest Neighbor, and ALNS using the same unseen data and all under the same hard constraints.

All methods are evaluated under the same hard constraints.

3.11 Baselines and Solvers

3.11.1 Nearest Neighbor (Greedy Construction).

The nearest neighbor builds a schedule one visit at a time using predetermined rules. At each step, it looks at the remaining unassigned visits that meet the constraint

requirements. From this list, it picks the easiest visit to schedule at that moment and assigns that visit to a clinician who can reach the patient the fastest or one that has the most availability.

Once the visit is scheduled, the heuristic will move on to the next visit without reconsidering its previous choices.

This approach is extremely fast. Even for large instances, solutions appear in seconds because the methods does not backtrack and never reconsiders alternate assignments. But this speed comes at a cost. The method makes decisions for the upcoming schedule based solely on immediate convenience without consideration of possible future consequences. Typically, the heuristic will paint itself into a corner earlier in the week. For example, it might schedule a flexible clinician to an easy visit on Monday only to discover on Thursday that no one is available for the difficult visit the initial flexible clinician could have handled. These mistakes have a cascading effect as earlier errors accumulate and compound, leading to fragmented schedules and insufficient coverage.

The nearest neighbor is the starting point for this evaluation. It is the simplest way to schedule and is the lowest level of performance that any reasonable method should exceed. The nearest neighbor method does not balance multiple objectives nor can it recover from poor decisions made earlier on. But its speed and simplicity provides an important point of reference for understanding improvement sophisticated methods can achieve.

3.11.2 CP-SAT (Exact Optimization).

Constraint Programming-SAT (CP-SAT) is a solver that produces a close-to-optimal schedule by testing various ways of assigning clinicians to visits. It checks each potential assignment to see if it violates the constraints, which include overlapping visits, exceeding work hours, or impossible travel times. If CP-SAT discovers that an as-

signment violates a constraint, it ignores that assignment and tries a different one. It continues to test different combinations until it either finds a schedule that satisfies all constraints and completes the maximum number of visits, or runs out of time.

In this evaluation, CP-SAT establishes the maximum number of visits that can feasibly be completed. This represents the best possible coverage performance under the given constraints. It does this by viewing each assignment as a yes-or-no proposition i.e. whether clinician i should be assigned to visit j at time t. Then, CP-SAT tests for the combination of 'yes' and 'no' answers that lead to the most assignments within the constraints. CP-SAT in this setup only focuses on maximizing coverage and does not balance any secondary objectives, such as workload distribution or continuity of care. The solver produces a solution within seconds on small instances. On medium or large instances, a limit of one-hour per instance is applied.

3.11.3 Adaptive Large Neighborhood Search (Iterative Improvement).

Adaptive Large Neighborhood Search, or ALNS, is an iterative method that starts with a complete schedule and tries to improve it step by step. The approach works by deliberately breaking parts of the existing schedule and then rebuilding those parts in a better way. For example, it might unassign all visits scheduled for a particular clinician or remove all visits in a specific geographic area, then reassign those visits while looking for ways to complete more of them. The "adaptive" part means the method learns as it goes. ALNS tracks which types of changes lead to better schedules and gradually focuses on the strategies that work best. If removing visits from overloaded clinicians consistently leads to improvements, ALNS will try that approach more often. If geographically clustered removals rarely help, it will shift to other strategies. This learning happens over many iterations, with each cycle of destroying and rebuilding taking the schedule closer to a better solution.

ALNS sits between the nearest neighbor heuristic and CP-SAT in terms of both

speed and results. It is slower than nearest neighbor because it explores many different ways to rearrange the schedule, but it could be faster than CP-SAT because it does not exhaustively test every possibility. Like CP-SAT, ALNS respects all hard constraints during reconstruction and focuses on maximizing visit coverage. However, it does not balance secondary objectives such as workload equity or continuity. ALNS also operates under the same one-hour time limit as CP-SAT. The best solution found within that window is recorded as the benchmark result.

Nearest neighbor, CP-SAT and ALNS provide three references to evaluate our DRL model. CP-SAT explores every option exhaustively, nearest neighbor makes quick local decisions, and ALNS improves solutions step by step through iterative refinement. All three focus solely on maximizing patient coverage without balancing secondary objectives. This makes them ideal for comparison. If the DRL model can match or outperform these single objective benchmarks, it demonstrates that adding multiple goals does not compromise coverage performance.

Chapter 4 – Results

In this chapter, we answer the following research questions:

- **RQ1:** Does a multiobjective deep reinforcement learning model achieve competitive visit-fulfillment rates compared to single-objective optimization methods (exact optimization, metaheuristic, constructive heuristic) on small and medium instances under identical hard constraints?
- **RQ2:** Does a deep reinforcement learning model yield a Pareto frontier over fulfillment, average idle rate, and average utilization rate with greater hypervolume and set coverage (C-metric) than exact optimization, metaheuristic, and constructive heuristic methods under identical hard constraints?
- **RQ3:** Across small and medium problem instances, does a deep reinforcement learning model achieve a higher Comprehensive General Composite Score (GCS_{10}, comprising 10 metrics across efficiency, equity, quality, and operational dimensions) than exact optimization and constructive heuristic methods under identical hard constraints?

We present our findings for each question and test the corresponding hypothesis.

4.1 Hypothesis 1

H1: Under identical hard constraints, a multiobjective deep reinforcement learning model attains *Visit Fulfillment* rates that are competitive with or superior to single objective benchmark methods (exact optimization, metaheuristic, constructive heuristic) on small and medium HHCRSP instances.

The primary objective of a home health agency is to provide patient care. The primary metric for this is *Visit Fulfillment*. It represents the percentage of requested patient visits that have been successfully assigned within a scheduling horizon. A score of 100% means every requested visit is scheduled, while lower values reflect

missed visits due to limitations such as clinician capacity, skill coverage, or time and travel constraints.

4.1.1 Visit Fulfillment - Small Instances

Table 4.1 and Table 4.4 summarize the performance of the DRL models against the benchmarks on small dataset instances. The DRL model achieved fulfillment rates of 95.63-96.46% across the three aggregator types and well above CP-SAT's 81%. That 14-15 point difference translated into perfect win-tie-loss record of 4-0-0. This means for each of the instances that were tested, the DRL model received a higher *Visit Fulfillment* score than CP-SAT, ALNS and NN. The Cliff's $\delta = 1.0$ with a 95% CI [95.48, 97.02] confirms this complete separation i.e. every DRL value exceeded every baseline value. Compared with the greedy nearest-neighbor (8.12%) and ALNS (19.79%), the DRL model improved coverage by more than 70 percentage points.

These results are notable because CP-SAT, ALNS and NN all focus on a single objective of *Visit Fulfillment* but still lost at their game. Meanwhile, DRL simultaneously optimized fairness, continuity, routing, and idle time but achieved higher *Visit Fulfillment* scores . The fact that DRL model produced higher *Visit Fulfillment* scores suggests that it learned assignment patterns that the benchmarks did not, possibly by understanding relationships between secondary objectives that the other three single-objective methods ignored.

The greedy benchmark's 8% coverage highlights how short-sighted choices ruin later opportunities, while ALNS, though better, still managed only 20% and an inference time of over 2,600 seconds per instance. DRL produced much higher fulfillment in under 30 seconds per run.

Table 4.1: Visit Fulfillment: DRL vs Benchmarks — Performance Metrics (Small)

Method	**WS**	**PA**	**Cheb**
n			
DRL	4	4	4
CP-SAT	4	4	4
NNS	4	4	4
ALNS	4	4	4
Mean ± SD (%)			
DRL	96.25±0.48	95.63±1.25	96.46±1.42
CP-SAT	81.46±6.98	81.46±6.98	81.46±6.98
NNS	8.12±1.05	8.12±1.05	8.12±1.05
ALNS	19.79±2.39	19.79±2.39	19.79±2.39
Median [IQR] (%)			
DRL	96.25 [95.83, 96.67]	95.83 [94.79, 96.67]	96.25 [95.62, 97.08]
CP-SAT	79.17 [78.33, 82.29]	79.17 [78.33, 82.29]	79.17 [78.33, 82.29]
NNS	8.33 [7.92, 8.54]	8.33 [7.92, 8.54]	8.33 [7.92, 8.54]
ALNS	20.00 [19.17, 20.62]	20.00 [19.17, 20.62]	20.00 [19.17, 20.62]
95% CI (%)			
DRL	[95.48, 97.02]	[93.64, 97.61]	[94.19, 98.72]
CP-SAT	[70.34, 92.57]	[70.34, 92.57]	[70.34, 92.57]
NNS	[6.46, 9.79]	[6.46, 9.79]	[6.46, 9.79]
ALNS	[15.98, 23.60]	[15.98, 23.60]	[15.98, 23.60]

Notes: n = Number of test instances; WS = Weighted Sum, PA = Policy Aware, Cheb = Chebyshev; pp = percentage points; Win-Tie-Loss = Record of DRL vs baseline (DRL wins-ties-losses); -= Method failed or timed out (ALNS for large instances). *Large instance DRL results based on preliminary training (1 trial, 32,768 timesteps); full convergence training (5 trials × 2M timesteps) in progress at time of writing.

4.1.2 Visit Fulfillment - Medium Instances

Table 4.2 and Table 4.4 summarize the performance of the DRL model against the benchmarks on medium dataset instances. At the medium scale of 80 patients, 225 visits, 10 clinicians and 45 workdays, DRL model achieved 79.22% *Visit Fulfillment* versus 80.00% for CP-SAT. A Mann-Whitney test found no statistically significant difference with p = 0.750.88. Cliff's $\delta = -0.50$ indicates that, in our small sample, CP-SAT tended to score higher, though the absolute gap was less that 2 percentage points and operationally small for scheduling. The 95% CIs overlapped substantially e.g., DRL [75.84, 82.61] vs CP-SAT [69.72, 90.28] under *Weighted Sum*. Still, CP-SAT led on 3 of 4 instances.

Despite optimizing multiple objectives, DRL matched CP-SAT's single-objective coverage while maintaining workload balance, continuity, and routing efficiency. CP-SAT remained very fast at approximately 3 seconds per instance, while DRL model took about 5 to 70 seconds. But these times are well within allowed time for practical daily scheduling.

Against the heuristics, DRL model's s advantage continued by approximately 70% over greedy NNS (7.11%; 95% CI [3.60, 10.62]) and approximately 60% over ALNS (19.6%; 95% CI [16.50, 22.61]).

4.1.3 Visit Fulfillment - Large Instances

Table 4.3 and Table 4.4 summarize the performance of the DRL models against the benchmarks on large dataset instances consisting of 150 patients, 430 visits, 20 clinicians, 90 workdays.

Here the trend reversed. The DRL model achieved $\approx 53\%$ visit fulfillment (95% CIs $[51.82, 54.34]$, $[51.72, 54.55]$, $[51.72, 54.45]$ across WS/PA/Cheb). These scores are far below CP-SAT's $\approx 90\%$ (95% CIs $[81.74, 98.14]$, $[81.54, 98.57]$, $[81.41, 98.59]$). CP-SAT swept the instances with a win-tie-loss of 0-0-4 and a Cliff's $\delta = -1.00$ showing

Table 4.2: Visit Fulfillment: DRL vs Benchmarks — Performance Metrics (Medium)

Method	**WS**	**PA**	**Cheb**
n			
DRL	4	4	4
CP-SAT	4	4	4
NNS	4	4	4
ALNS	4	4	4
Mean ± SD (%)			
DRL	79.22±2.13	78.78±1.86	78.89±2.65
CP-SAT	80.00±6.46	80.00±6.46	80.00±6.46
NNS	7.11±2.21	7.11±2.21	7.11±2.21
ALNS	19.56±1.92	19.56±1.92	19.67±2.00
Median [IQR] (%)			
DRL	79.11 [77.89, 80.44]	78.44 [77.89, 79.33]	79.11 [77.56, 80.44]
CP-SAT	81.56 [77.44, 84.11]	81.56 [77.44, 84.11]	81.56 [77.44, 84.11]
NNS	7.11 [6.11, 8.11]	7.11 [6.11, 8.11]	7.11 [6.11, 8.11]
ALNS	20.00 [18.89, 20.67]	20.00 [18.89, 20.67]	20.22 [18.89, 21.00]
95% CI (%)			
DRL	[75.84, 82.61]	[75.81, 81.74]	[74.67, 83.11]
CP-SAT	[69.72, 90.28]	[69.72, 90.28]	[69.72, 90.28]
NNS	[3.60, 10.62]	[3.60, 10.62]	[3.60, 10.62]
ALNS	[16.50, 22.61]	[16.50, 22.61]	[16.48, 22.85]

Notes: n = Number of test instances; WS = Weighted Sum, PA = Policy Aware, Cheb = Chebyshev; pp = percentage points; Win-Tie-Loss = Record of DRL vs baseline (DRL wins-ties-losses); = Method failed or timed out (ALNS for large instances). *Large instance DRL results based on preliminary training (1 trial, 32,768 timesteps); full convergence training (5 trials × 2M timesteps) in progress at time of writing.

complete separation with Mann-Whitney $p = 0.125$ for each aggregator ($n = 4$ per group).

This drop is in part due to limited training. While small and medium runs trained for five full trials of two million steps each, large scale instances were only partially trained with one trial of $\sim$ 32,768 steps. Even so, the policy still produced a valid, constraint-compliant schedule, showing that it learned the rules. However, it hadn't yet mastered the long-horizon trade-offs present in the larger dataset.

Because CP-SAT prioritizes visit coverage, it handled the over-constrained large instances better than the partially trained multi-objective DRL model, which continued to balance objectives and left feasible visits uncovered.

ALNS did not return a feasible solution within the allowed cutoff time and greedy NNS remained around 10% (95% CI $[8.84, 11.05]$).

These results confirm Hypothesis 1. Under identical hard constraints, a multiobjective deep reinforcement learning model attains *Visit Fulfillment* rates that are competitive with or superior to single objective benchmark methods (exact optimization, metaheuristic, constructive heuristic) on small and medium HHCRSP instances.

4.2 Hypothesis 2A

H2a: Under identical hard constraints, a deep reinforcement learning model achieves a statistically significantly larger Pareto hypervolume than exact optimization, metaheuristic, and constructive heuristic methods across all instance sizes.

When a scheduling method balances multiple objectives, such as coverage, travel time, fairness, and continuity, it produces solutions with different trade-offs. Some solutions might have high patient coverage but uneven workloads, while others might have lower patient coverage but better balance across all objectives. To compare methods that produce these different trade-offs, we need a way to measure how well

Table 4.3: Visit Fulfillment: DRL vs Benchmarks — Performance Metrics (Large)

Method	**WS**	**PA**	**Cheb**
n			
DRL	4	4	4
CP-SAT	4	4	4
NNS	4	4	4
ALNS	-	-	-
Mean ± SD (%)			
DRL	53.08±0.79	53.08±0.86	53.08±0.86
CP-SAT	89.94±5.15	90.06±5.35	90.00±5.40
NNS	9.94±0.69	9.94±0.69	9.94±0.69
ALNS	-	-	-
Median [IQR] (%)			
DRL	52.91 [52.67, 53.31]	53.02 [52.79, 53.31]	53.02 [52.79, 53.31]
CP-SAT	89.19 [86.45, 92.67]	89.19 [86.45, 92.79]	89.07 [86.28, 92.79]
NNS	10.00 [9.59, 10.35]	10.00 [9.59, 10.35]	10.00 [9.59, 10.35]
ALNS	-	-	-
95% CI (%)			
DRL	[51.82, 54.34]	[51.72, 54.45]	[51.72, 54.45]
CP-SAT	[81.74, 98.14]	[81.54, 98.57]	[81.41, 98.59]
NNS	[8.84, 11.05]	[8.84, 11.05]	[8.84, 11.05]
ALNS	-	-	-

Notes: n = Number of test instances; WS = Weighted Sum, PA = Policy Aware, Cheb = Chebyshev; pp = percentage points; Win-Tie-Loss = Record of DRL vs baseline (DRL wins-ties-losses); - = Method failed or timed out (ALNS for large instances). *Large instance DRL results based on preliminary training (1 trial, 32,768 timesteps); full convergence training (5 trials × 2M timesteps) in progress at time of writing.

Table 4.4: Visit Fulfillment: DRL vs Benchmarks — Statistical Comparison

	Small			Medium			Large		
Method	**WS**	**PA**	**Cheb**	**WS**	**PA**	**Cheb**	**WS**	**PA**	**Cheb**
Mean Diff vs DRL (pp)									
CP-SAT	14.79	14.17	15.00	-0.78	-1.22	-1.11	-36.86	-36.98	-36.92
NNS	88.13	87.50	88.33	72.11	71.67	71.78	43.14	43.14	43.14
ALNS	76.46	75.83	76.67	59.67	59.22	59.22	-	-	-
p-value									
CP-SAT	0.125	0.125	0.125	0.875	0.750	0.875	0.125	0.125	0.125
NNS	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
ALNS	0.125	0.125	0.125	0.125	0.125	0.125	-	-	-
Cliff's Delta									
CP-SAT	1.00	1.00	1.00	-0.50	-0.50	-0.50	-1.00	-1.00	-1.00
NNS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ALNS	1.00	1.00	1.00	1.00	1.00	1.00	-	-	-
Win-Tie-Loss									
CP-SAT	4-0-0	4-0-0	4-0-0	1-0-3	1-0-3	1-0-3	0-0-4	0-0-4	0-0-4
NNS	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0
ALNS	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	-	-	-
Median Runtime (seconds)									
DRL	27.3	24.3	20.9	69.7	65.4	69.3	291.9	189.6	321.7
CP-SAT	2.4	2.3	2.7	3.4	3.2	3.2	618.4	248.3	229.3
NNS	4.2	4.5	4.2	29.5	30.9	29.8	347.7	333.5	346.8
ALNS	2649.3	2643.7	2620.8	18016.2	17838.7	17754.6	-	-	-
Violations (%)									
DRL	0	0	0	0	0	0	0	0	0
CP-SAT	0	0	0	0	0	0	0	0	0
NNS	0	0	0	0	0	0	0	0	0
ALNS	0	0	0	0	0	0	-	-	-

Notes: n = Number of test instances; WS = Weighted Sum, PA = Policy Aware, Cheb = Chebyshev; pp = percentage points; Win-Tie-Loss = Record of DRL vs baseline (DRL wins-ties-losses); - = Method failed or timed out (ALNS for large instances). *Large instance DRL results based on preliminary training (1 trial, 32,768 timesteps); full convergence training (5 trials × 2M timesteps) in progress at time of writing.

each method explores the full range of possibilities.

Hypervolume provides this measurement. It calculates the total volume of objective space that a method covers when considering all its solutions together. A larger Hypervolume means that the method found more solutions with better trade-offs across all objectives. If one method consistently finds solutions that score well on multiple objectives simultaneously, it will have a larger Hypervolume than a method that only optimizes one objective or fails to find good combinations.

4.2.1 Hypervolume - Small Instances

Tables 4.5 and 4.6 summarize Pareto Hypervolume on small instances. Across the three aggregators, the DRL model achieved a Hypervolume mean of of 5.21×10^7, 5.22×10^7, and 5.34×10^7, respectively, while the baselines were an order of magnitude smaller. For example, CP-SAT at $\approx 3.7 \times 10^5$, Greedy NN at $\approx 1.4 \times 10^6$, and ALNS at $\approx 8.6 \times 10^6$. Mann-Whitney tests (Table 4.6 and Table 4.8) show statistically significant improvements for the DRL model over each baseline with $p = 0.05$ and large absolute Hypervolume differences in the order of 10^7, supporting H2a.

Figure 4.1: 3D Pareto Frontiers - Weighted Sum (Small)

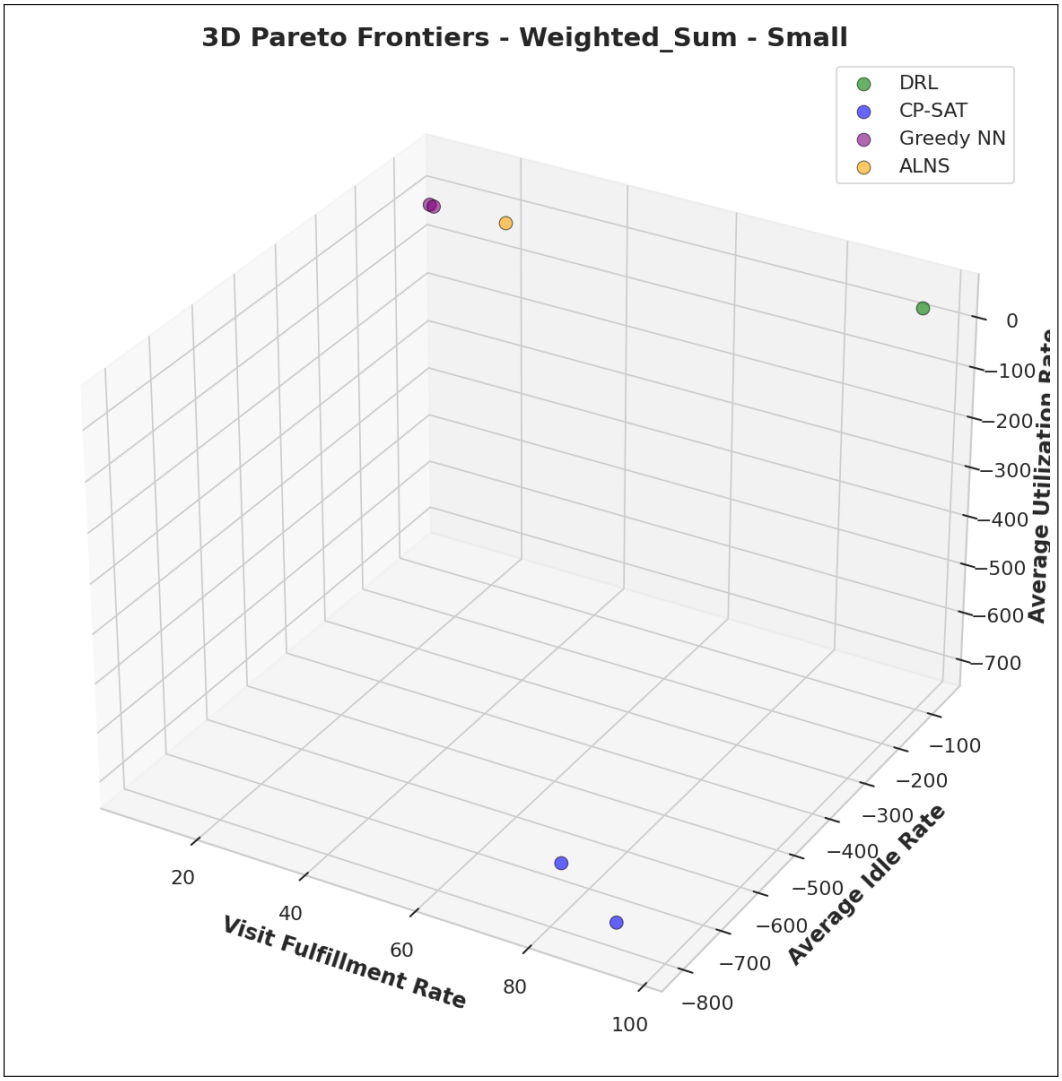


Figure 4.2: 3D Pareto Frontiers - Policy Aware (Small)

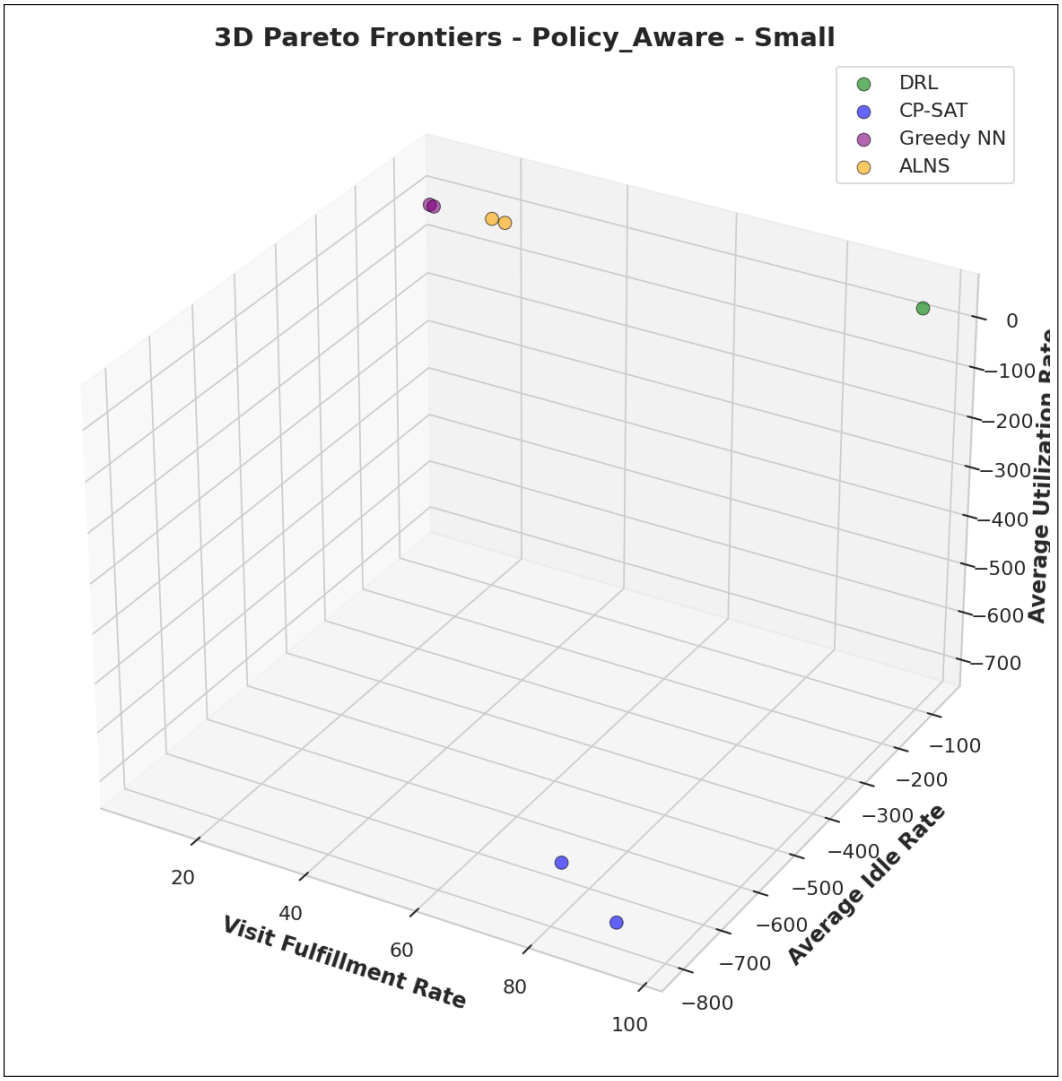
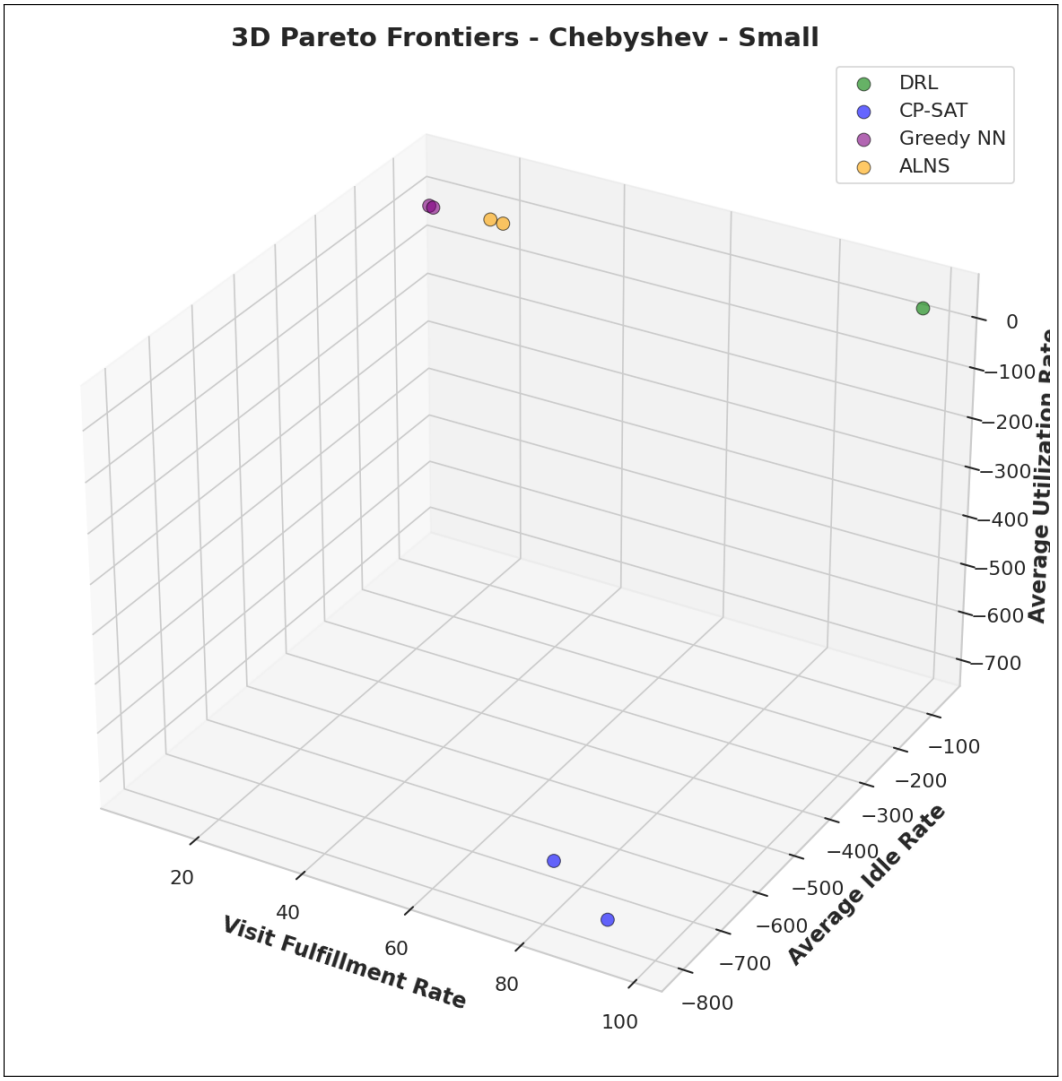


Figure 4.3: 3D Pareto Frontiers - Chebyshev (Small)



4.2.2 Hypervolume - Medium Instances

Tables 4.5 and 4.6 summarize Pareto hypervolume on the medium instances. Here, the DRL model achieved mean Hypervolume of 5.44×10^7, 5.42×10^7, and 5.46×10^7, respectively. CP-SAT received $\approx 1.46 \times 10^6$-1.48×10^6, ALNS at $\approx 1.15 \times 10^7$, and Greedy NN at $\approx 3.66 \times 10^6$. On Table 4.6, Mann-Whitney tests indicate the DRL model's Hypervolume as *statistically significantly* larger than each baseline on WS, PA, and Cheb at $p = 0.05$, with large effect sizes (absolute HV differences on the order of 4×10^7-5×10^7). With fixed *Visit Fulfillment*, these results show that DRL covers

a substantially larger region of high-quality trade-offs than exact optimization, the metaheuristic, and the constructive heuristic on the medium instances, supporting H2a.

Figure 4.4: 3D Pareto Frontiers - Weighted Sum (Medium)

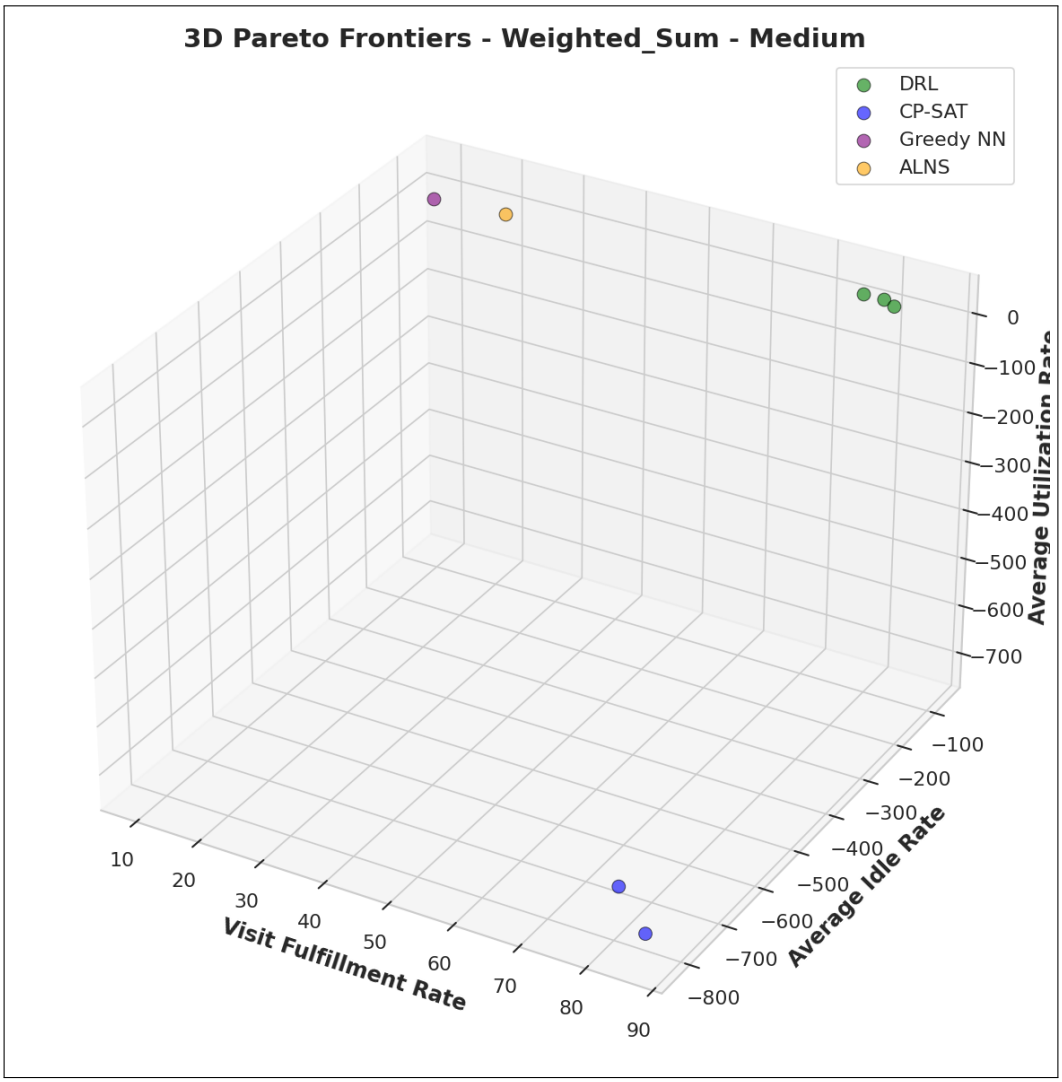


Figure 4.5: 3D Pareto Frontiers - Policy Aware (Medium)

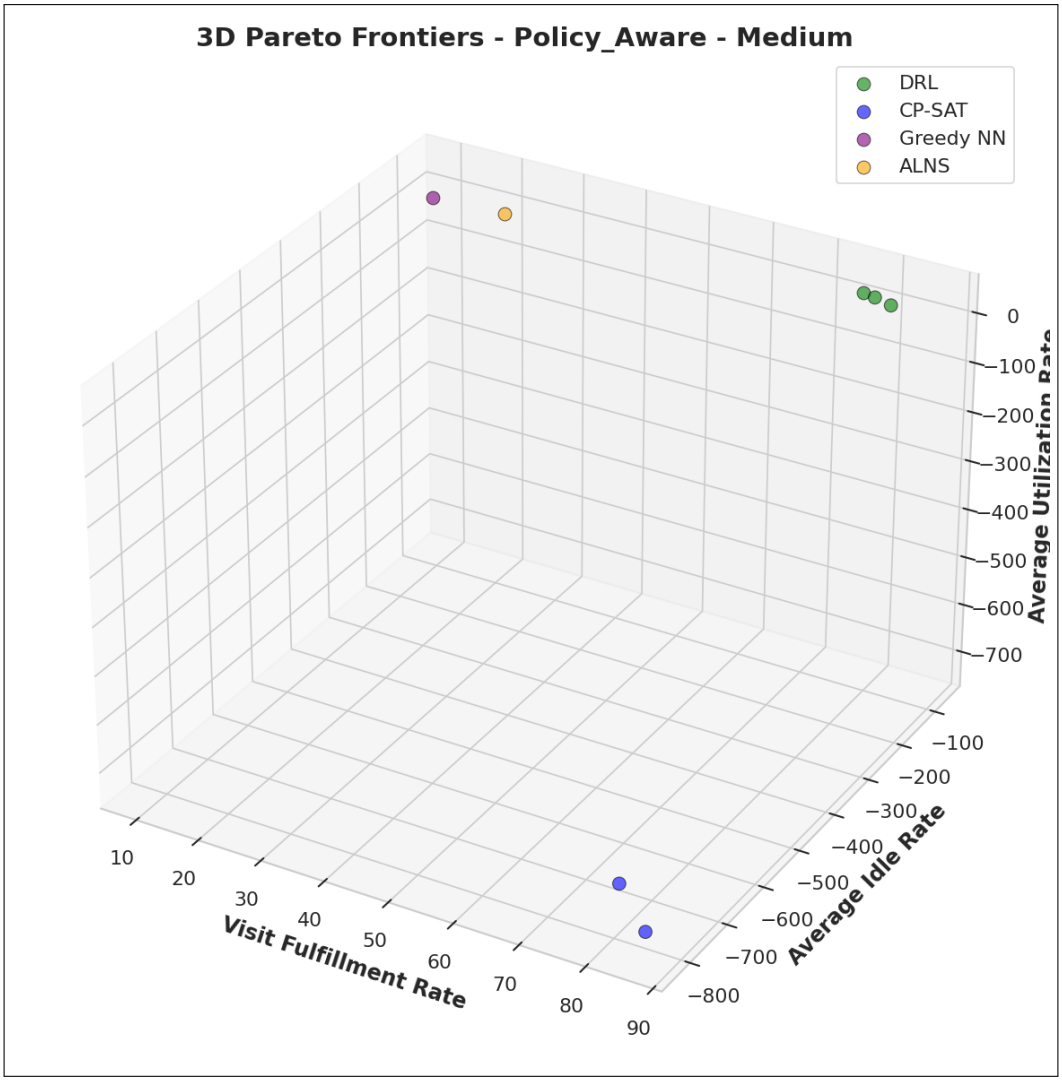
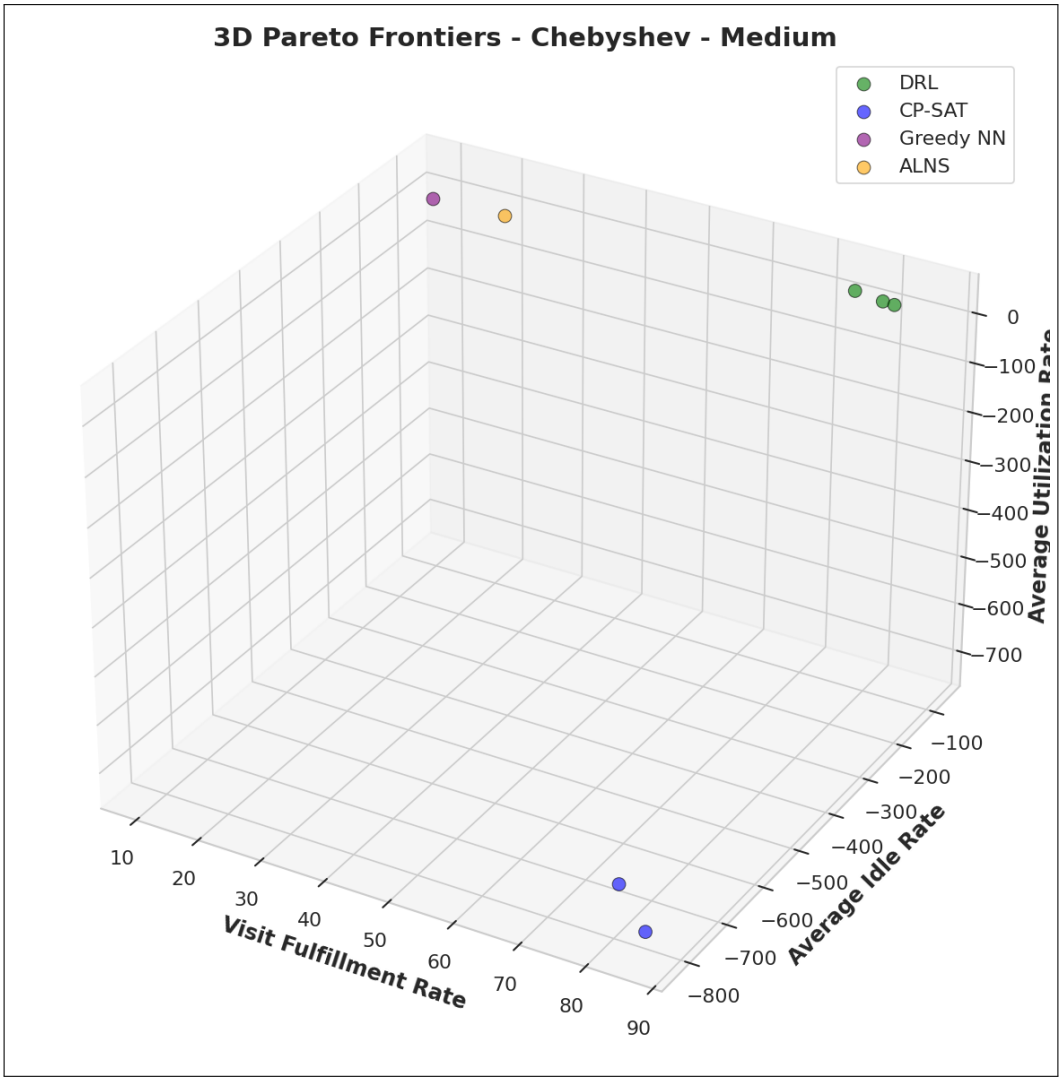


Figure 4.6: 3D Pareto Frontiers - Chebyshev (Medium)



4.2.3 Hypervolume - Large Instances

Tables 4.5 and 4.6 summarize Pareto hypervolume on the large instances. Across WS/PA/Cheb, DRL attained mean HVs of 2.50×10^7 for each aggregator, substantially exceeding CP-SAT's $\approx 1.15 \times 10^6$, 8.37×10^5, and 8.29×10^5, and Greedy NN's $\approx 9.33 \times 10^5$; ALNS is omitted at this scale due to timeout. Mann-Whitney tests (Table 4.6 and Table 4.8) indicate *statistically significant* improvements for the DRL model over CP-SAT and Greedy NN at $p = 0.05$ on all three aggregators, with large absolute Hypervolume differences in the order of 2.4-2.5 $\times 10^7$. With fixed *Visit*

Fulfillment fixed, these results show that DRL covers a markedly larger region of high-quality trade-offs than exact optimization and the constructive heuristic on the large instances.

Figure 4.7: 3D Pareto Frontiers - Weighted Sum (Large)

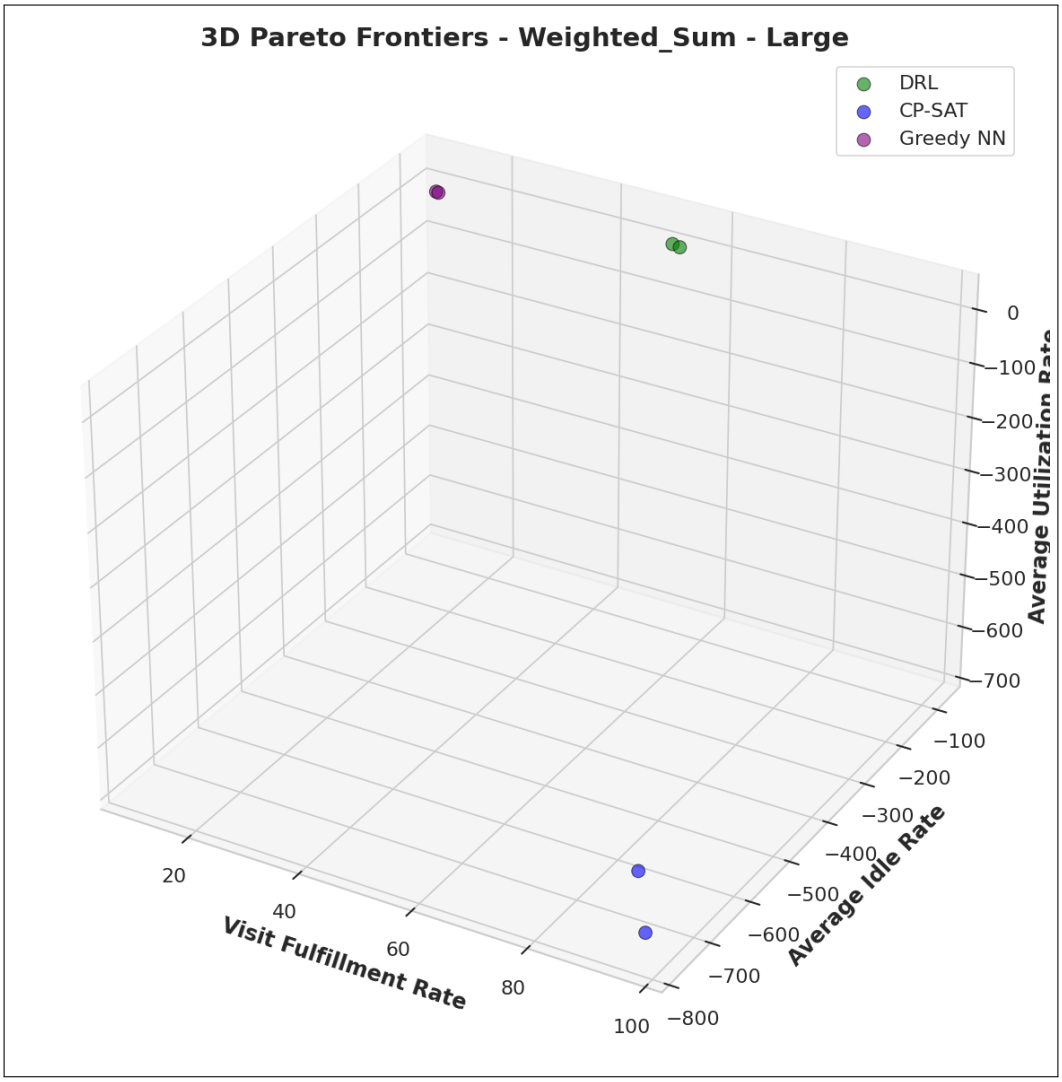


Figure 4.8: 3D Pareto Frontiers - Policy Aware (Large)

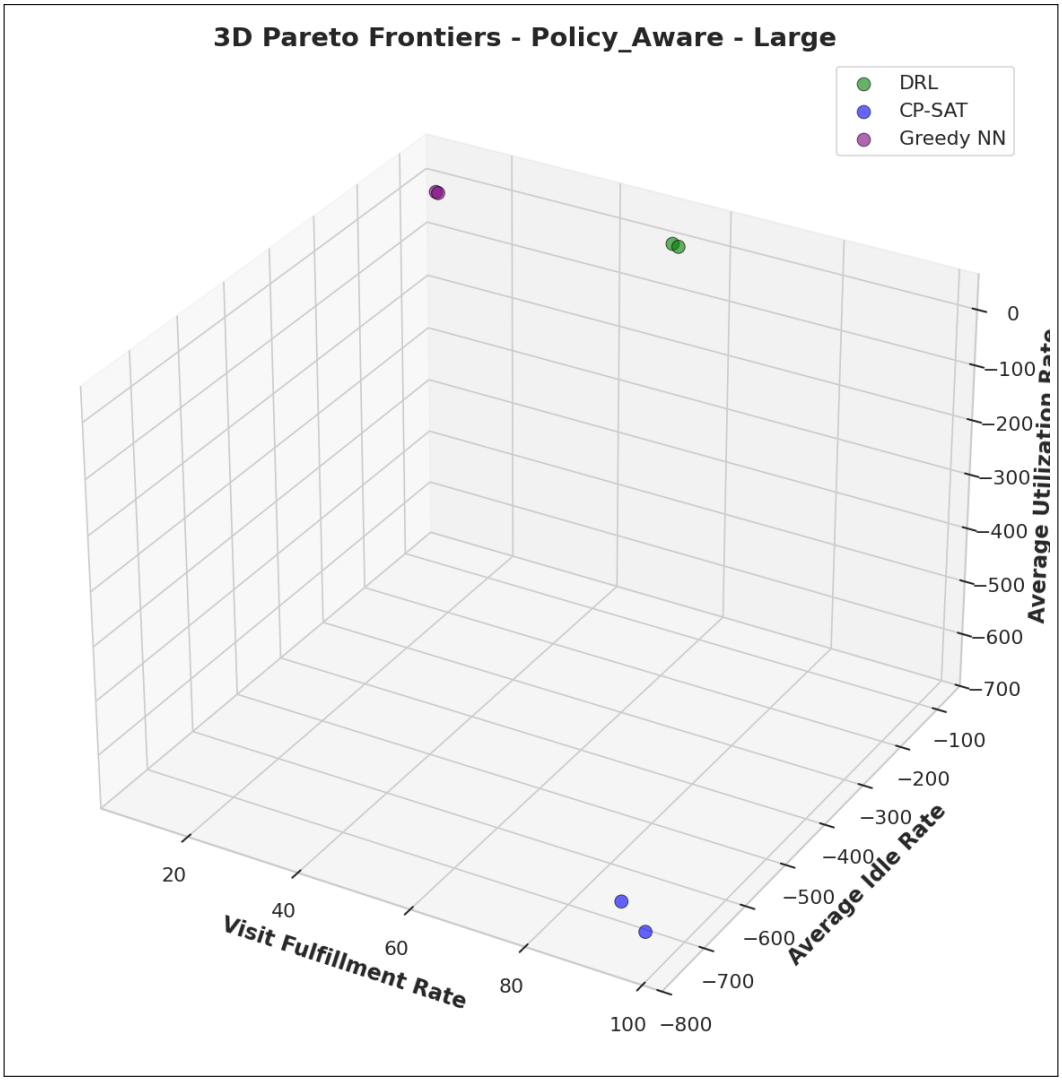
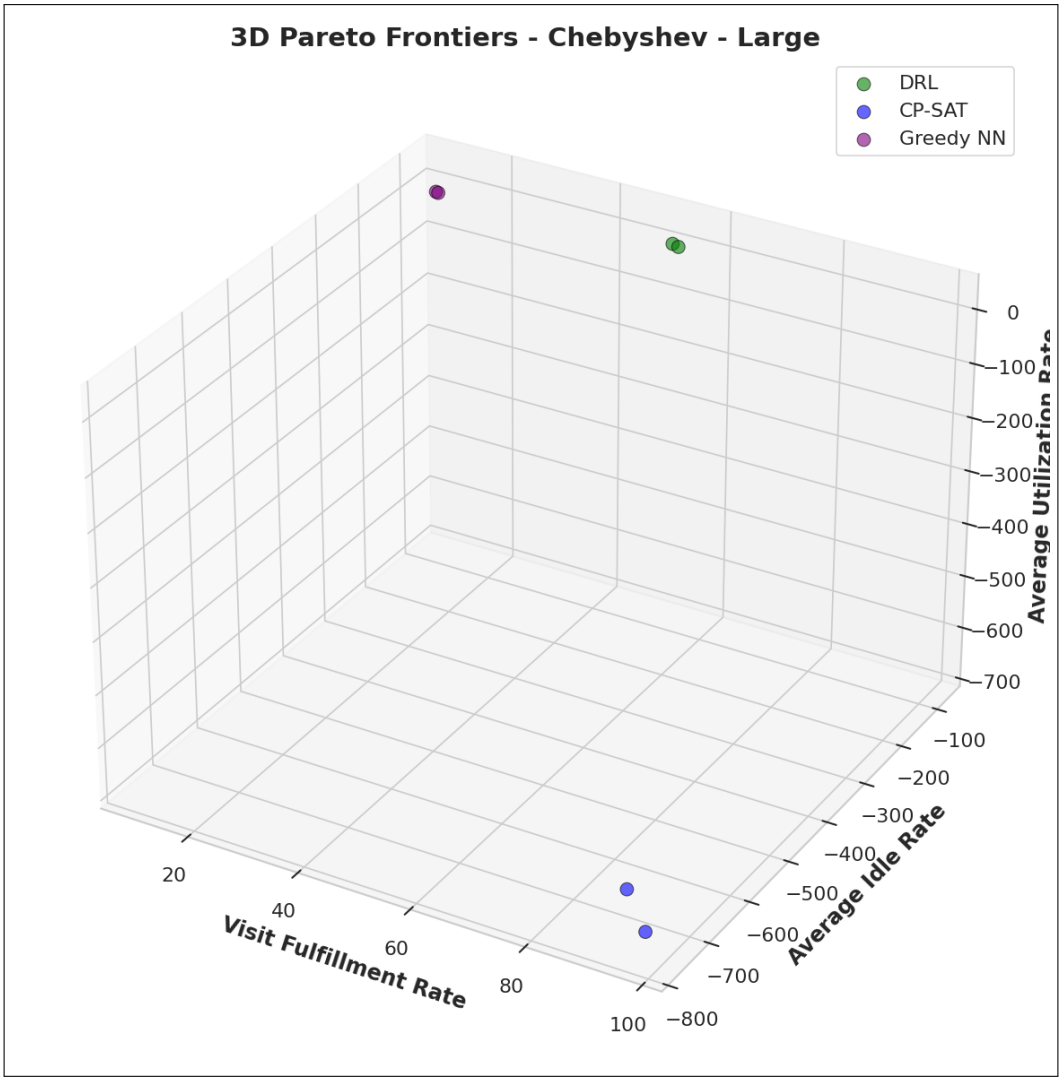


Figure 4.9: 3D Pareto Frontiers - Chebyshev (Large)



These results confirm Hypothesis 2A. Under identical hard constraints, a deep reinforcement learning model achieves a statistically significantly larger Pareto hypervolume than exact optimization, metaheuristic, and constructive heuristic methods across all instance sizes.

4.3 Hypothesis 2B

H2b: Under identical hard constraints, a deep reinforcement learning model demonstrates superior set coverage (C-metric) compared to metaheuristic and constructive

Table 4.5: Comprehensive Performance Metrics: DRL vs Benchmarks

	Small			**Medium**			**Large**		
Method	WS	PA	Cheb	WS	PA	Cheb	WS	PA	Cheb
Number of Solutions									
DRL	4	4	4	4	4	4	4	4	4
CP-SAT	4	4	4	4	4	4	4	4	4
Greedy NN	4	4	4	4	4	4	4	4	4
ALNS	4	4	4	4	4	4	-	-	-
Number of Pareto Points									
DRL	1	1	1	3	3	3	2	2	2
CP-SAT	2	2	2	2	2	2	2	2	2
Greedy NN	2	2	2	1	1	1	2	2	2
ALNS	1	2	2	1	1	1	-	-	-
% Pareto Points									
DRL	25%	25%	25%	75%	75%	75%	50%	50%	50%
CP-SAT	50%	50%	50%	50%	50%	50%	50%	50%	50%
Greedy NN	50%	50%	50%	25%	25%	25%	50%	50%	50%
ALNS	25%	50%	50%	25%	25%	25%	-	-	-
Hypervolume ($\times 10^7$)									
DRL	5.21	5.22	5.34	5.44	5.42	5.46	2.50	2.50	2.50
CP-SAT	0.04†	0.04†	0.04†	0.15†	0.15†	0.15†	0.12†	0.08†	0.08†
Greedy NN	0.14†	0.14†	0.14†	0.37†	0.37†	0.37†	0.09†	0.09†	0.09†
ALNS	8.63†	8.65†	8.65†	11.54†	11.53†	11.50†	-	-	-
C(DRL, CP-SAT)									
Value	1	1	1	0.5	0.5	0.5	0	0	0
C(DRL, ALNS)									
Value	1	1	1	1	1	1	-	-	-
C(DRL, Greedy NN)									
Value	1	1	1	1	1	1	1	1	1
Number of Non-dominated (Global)									
DRL	1	1	1	3	3	3	2	2	2
CP-SAT	0	0	0	1	1	1	2	2	2
Greedy NN	0	0	0	0	0	0	0	0	0
ALNS	0	0	0	0	0	0	-	-	-
Median Runtime (seconds)									
DRL	27.3	24.3	20.9	69.7	65.4	69.3	291.9	189.6	321.7
CP-SAT	2.4	2.3	2.7	3.4	3.2	3.2	618.4	248.3	229.3
Greedy NN	4.2	4.5	4.2	29.5	30.9	29.8	347.7	333.5	346.8
ALNS	2649.3	2643.7	2620.8	18016.2	17838.7	17754.6	-	-	-
% Capped Solutions									
DRL	0%	0%	0%	0%	0%	0%	0%	0%	0%
CP-SAT	0%	0%	0%	0%	0%	0%	0%	0%	0%
Greedy NN	0%	0%	0%	0%	0%	0%	0%	0%	0%
ALNS	0%	0%	0%	100%	100%	100%	-	-	-

† Statistically significant difference from DRL (p=0.05, large effect size)
WS = Weighted Sum, PA = Policy Aware, Cheb = Chebyshev
C(A,B) = Coverage metric: fraction of B's solutions dominated by A
- = Method failed or timed out (ALNS for large instances)

Table 4.6: Detailed Statistical Tests: DRL vs Benchmarks (Mann-Whitney U)

Comparison	DRL Mean HV	Baseline Mean HV	p-value	Effect Size
Weighted Sum - Small				
DRL vs CP-SAT	5.21×10^7	3.70×10^5	0.05	5.17×10^7 (large)
DRL vs ALNS	5.21×10^7	8.63×10^6	0.05	4.35×10^7 (large)
DRL vs Greedy NN	5.21×10^7	1.40×10^6	0.05	5.07×10^7 (large)
Policy Aware - Small				
DRL vs CP-SAT	5.22×10^7	3.79×10^5	0.05	5.18×10^7 (large)
DRL vs ALNS	5.22×10^7	8.65×10^6	0.05	4.36×10^7 (large)
DRL vs Greedy NN	5.22×10^7	1.40×10^6	0.05	5.08×10^7 (large)
Chebyshev - Small				
DRL vs CP-SAT	5.34×10^7	3.79×10^5	0.05	5.30×10^7 (large)
DRL vs ALNS	5.34×10^7	8.65×10^6	0.05	4.47×10^7 (large)
DRL vs Greedy NN	5.34×10^7	1.40×10^6	0.05	5.20×10^7 (large)
Weighted Sum - Medium				
DRL vs CP-SAT	5.44×10^7	1.46×10^6	0.05	5.30×10^7 (large)
DRL vs ALNS	5.44×10^7	1.15×10^7	0.05	4.29×10^7 (large)
DRL vs Greedy NN	5.44×10^7	3.66×10^6	0.05	5.08×10^7 (large)
Policy Aware - Medium				
DRL vs CP-SAT	5.42×10^7	1.48×10^6	0.05	5.27×10^7 (large)
DRL vs ALNS	5.42×10^7	1.15×10^7	0.05	4.27×10^7 (large)
DRL vs Greedy NN	5.42×10^7	3.66×10^6	0.05	5.05×10^7 (large)
Chebyshev - Medium				
DRL vs CP-SAT	5.46×10^7	1.47×10^6	0.05	5.31×10^7 (large)
DRL vs ALNS	5.46×10^7	1.15×10^7	0.05	4.31×10^7 (large)
DRL vs Greedy NN	5.46×10^7	3.66×10^6	0.05	5.09×10^7 (large)
Weighted Sum - Large				
DRL vs CP-SAT	2.50×10^7	1.15×10^6	0.05	2.39×10^7 (large)
DRL vs Greedy NN	2.50×10^7	9.33×10^5	0.05	2.41×10^7 (large)
Policy Aware - Large				
DRL vs CP-SAT	2.50×10^7	8.37×10^5	0.05	2.42×10^7 (large)
DRL vs Greedy NN	2.50×10^7	9.33×10^5	0.05	2.41×10^7 (large)
Chebyshev - Large				
DRL vs CP-SAT	2.50×10^7	8.29×10^5	0.05	2.42×10^7 (large)
DRL vs Greedy NN	2.50×10^7	9.33×10^5	0.05	2.41×10^7 (large)

Effect size calculated as absolute difference in mean hypervolume.
All effect sizes classified as "large" based on domain-specific thresholds.
ALNS comparisons omitted for large instances due to method timeout.

Table 4.7: Pareto Dominance Summary: DRL vs Benchmarks Across Problem Sizes and Aggregators

	Small			**Medium**			**Large**		
Metric	WS	PA	Cheb	WS	PA	Cheb	WS	PA	Cheb
DRL Points	1	1	1	3	3	3	2	2	2
Total Points	4	4	4	4	4	4	4	4	4
% of Front	25%	25%	25%	75%	75%	75%	50%	50%	50%
CP-SAT Points	0	0	0	1	1	1	2	2	2

Note: ALNS failed to produce solutions for large instances within time limits.

Table 4.8: Statistical Significance of DRL vs Benchmarks (Mann-Whitney U Test, p=0.05)

	Small			**Medium**			**Large**		
vs Method	WS	PA	Cheb	WS	PA	Cheb	WS	PA	Cheb
CP-SAT	*	*	*	*	*	*	*	*	*
ALNS	*	*	*	*	*	*	-	-	-
Greedy NN	*	*	*	*	*	*	*	*	*

All significant comparisons show large effect sizes (d ¿ 0.8).
*Legend: * p=0.05 (large effect), - = insufficient data (ALNS timeout)*

heuristic methods across all scales. It exhibits full dominance at small scales, partial dominance at medium scales, and complementary Pareto contributions at large scales when compared with the exact optimization method.

Set coverage measures how often one method's solutions are better than another method's solutions when comparing them head-to-head. The metric works by taking each solution from one method and checking whether any solution from the competing method is at least as good on all objectives and strictly better on at least one objective. If this is true for every solution from the first method, the the second method has complete set coverage. A set coverage score of 1.0 means total dominance. It indicates that for every solution the first method found, the second method found at least one solution that was equal or better on all objectives and strictly better on at least one.

A set coverage score is reported as $C(A, B)$ which measures how much method A dominates method B. If $C(A, B) = 1.0$, then every solution from method B is worse than at least one solution from method A. This indicates total dominance i.e. method A provided a better alternative for every single solution that method B offered. If $C(A, B) = 0.0$, then the methods explored different or complementary regions,which means that neither consistently dominates the other. Values between 0 and 1 indicate partial dominance where some but not all of method B's solutions are beaten by method A.

4.3.1 Set Coverage - Small Instances

Table 4.5 reports set-coverage (C-metric) values of $C(\text{DRL}, \text{CP-SAT}) = 1$, $C(\text{DRL}, \text{ALNS}) = 1$, and $C(\text{DRL}, \text{Greedy NN}) = 1$ for WS, PA, and Cheb. Thus, every baseline solution is weakly dominated by at least one DRL model's solution on small instances. Consistent with this, the DRL model contributes the only globally non-dominated solution among the methods at the small scale. (Table 4.5), and the Pareto dominance summary (Table 4.7) attributes the small instance front to DRL. The results indicates that *full set-coverage dominance* by DRL over both the metaheuristic and the constructive heuristic methods, and superiority to the exact optimization baseline on the Pareto front. The results therefore confirm **H2b** on small instances.

4.3.2 Set Coverage - Medium Instances

Medium scale: Shared ground with CP-SAT. As complexity grew, DRL continued to dominate the heuristics ($C = 1.0$) and began sharing the Pareto front with CP-SAT ($C = 0.5$).

Table 4.5 reports set-coverage values of $C(\text{DRL}, \text{ALNS}) = 1$ and $C(\text{DRL}, \text{Greedy NN}) = 1$ across WS, PA, and Chebyshev, indicating full dominance of both heuristic baselines. Against exact optimization, $C(\text{DRL}, \text{CP-SAT}) = 0.5$ for WS, PA, and Chebyshev, meaning half of CP-SAT's solutions are dominated by DRL while the remainder contribute to the front. Consistent with this, three of the four frontier points came from DRL and one from CP-SAT (Table 4.7).

These show full dominance over the metaheuristic and constructive heuristic and complementary Pareto contributions with exact optimization at this scale. The numbers align with Hypothesis H2b on medium instances.

4.3.3 Set Coverage - Large Instances

Large scale: Distinct coverage, not competition At large scale, C(DRL, CP SAT) = 0.0 across all reward types, meaning neither method dominated the other. Each contributed two points to the global frontier. DRL's solutions focused on higher utilization and balanced workloads, while CP SAT's emphasized strict coverage. Because they covered different regions of the trade-off space, combining them could yield a more complete Pareto frontier than each method operating in isolation.

Table 4.5 reports set-coverage values of $C(\text{DRL}, \text{Greedy NN}) = 1$ across the three aggregators with full dominance of Greedy NN by DRL. ALNS timed out and did not generate results. Against exact optimization, coverage is complementary. $C(\text{DRL}, \text{CP-SAT}) = 0$ on all three aggregators, and the Pareto dominance summary (Table 4.7) shows the front split evenly between the two methods with DRL contributing 2 of 4 points (50%) and CP-SAT contributes the remaining 2 of 4 points (50%).

The results demonstrate full dominance of DRL over the constructive heuristic and complementary Pareto contributions with exact optimization on the large instances, in line with H2b.

These results confirm Hypothesis 2B. Under identical hard constraints, a deep reinforcement learning model demonstrates superior set coverage (C-metric) compared to metaheuristic and constructive heuristic methods across all scales. It exhibits full dominance at small scales, partial dominance at medium scales, and complementary Pareto contributions at large scales when compared with the exact optimization method

4.4 Hypothesis 3

H3: The deep reinforcement learning policy's mean Comprehensive GCS_{10} will exceed both exact optimization and constructive heuristic methods on small and medium problem sizes.

The Comprehensive General Composite Score (GCS_{10}) combines ten metrics covering efficiency (visit fulfillment, mean and total travel time, utilization), equity (workload Gini and variance), quality (continuity and patient preference satisfaction), and operations (idle rate, schedule density). Visit fulfillment carries a weight of 100, while all other metrics weigh between 0.5 and 1.0. This 100× gap prevents methods with poor coverage scores from scoring higher when they have a low visit fulfillment rate, even if they are balanced elsewhere.

4.4.1 General Composite Score - Small Instances

Table 4.9 reports the Comprehensive GCS_{10} on small instances across the three aggregation methods. The DRL model achieved mean composite scores of 0.910 ± 0.077 (WS), 0.907 ± 0.087 (PA), and 0.895 ± 0.080 (Cheb). These values are well above CP-SAT's, whose corresponding means were 0.740 ± 0.103, 0.740 ± 0.103, and 0.727 ± 0.101. The DRL model also far exceeded Greedy NN at 0.020 ± 0.014 and ALNS at 0.158 ± 0.030-0.159 ± 0.033. In absolute terms, DRL's advantage over CP-SAT is about 0.17 points and over the heuristics ranges from at least 0.74 to 0.89.

Instance level comparisons confirm this pattern. As summarized in Table 4.11, DRL achieved a win-tie-loss of 4-0-0 against each baseline (CP-SAT, ALNS, and Greedy NN) under WS, PA, and Chebyshev, indicating that on every tested small instance DRL's composite score exceeded the baselines. The corresponding effect sizes in Table 4.10 are Cliff's $\delta = 1.00$ for every pairwise comparison, which implies complete separation in favor of DRL. With $n = 4$ per group, the Mann-Whitney test

returns $p = 0.0625$ for each comparison; this is consistent with a large effect but falls just above the 0.05 threshold.

The construction of GCS_{10} further explains the robustness of these results. As defined in Table 4.12, the composite aggregates ten metrics i.e. efficiency, equity, quality, and operations, with *Visit Fulfillment* weighted 100× higher than any other component. Because DRL achieved higher visit fulfillment scores on small instances while maintaining strong performance on the remaining dimensions, it also earned higher overall GCS_{10}.

These results support **Hypothesis H3** on small instances.

4.4.2 General Composite Score - Medium Instances

Table 4.9 reports the Comprehensive GCS_{10} on the medium instances across the three aggregation methods. This time the DRL model achieved mean composite scores of 0.828 ± 0.081 (WS), 0.821 ± 0.081 (PA), and 0.824 ± 0.047 (Cheb). CP-SAT, attained 0.817 ± 0.088, 0.818 ± 0.089, and 0.816 ± 0.089 on the same aggregators. Greedy NN had 0.038 ± 0.032 and ALNS achieved 0.197 ± 0.034-0.199 ± 0.034. DRL model's mean *mean* GCS_{10} slightly exceeded CP-SAT on all three aggregators and exceeded both heuristics by large margins.

Instance level comparisons and significance tests confirm these mean differences. As shown in Table 4.11, DRL vs CP-SAT yielded win-tie-loss records of 1-0-3 (WS), 1-0-3 (PA), and 2-0-2 (Cheb), indicating that CP-SAT tended to lead on two aggregators while Chebyshev was split. Consistently, Table 4.10 reports Mann-Whitney p-values of 0.6875 (WS), 0.6875 (PA), and 0.5625 (Cheb) with corresponding Cliff's $\delta = -0.50$, -0.50, and 0.00, respectively. There was no statistically significant difference with small sample effect sizes pointing slightly toward CP-SAT on WS/PA and parity under Chebyshev.

Against the heuristics, DRL dominated decisively. DRL posted a win-tie-loss of

4-0-0 versus both ALNS and Greedy NN on all three aggregators (Table 4.11. The associated effect sizes were Cliff's $\delta = 1.00$ with Mann-Whitney $p = 0.0625$ for each comparison (Table 4.10).

Finally, the structure of GCS_{10} explains why improvements in *Visit Fulfillment* translate into composite gains even when secondary metrics are similar.

The medium-instance results show that DRL's *mean* GCS_{10} exceeds CP-SAT and the heuristic baselines across WS, PA, and Chebyshev and supports **Hypothesis H3**based on mean performance.

4.4.3 General Composite Score - Large Instances

Table 4.9 reports the Comprehensive GCS_{10} on the large instances across the three aggregation. Here, DRL posted a mean composite scores of 0.465 ± 0.043 (WS), 0.465 ± 0.047 (PA), and 0.465 ± 0.047 (Cheb). In contrast, CP-SAT, achieved substantially higher means of 0.806 ± 0.105, 0.804 ± 0.106, and 0.799 ± 0.106, respectively. Greedy NN remained very low at 0.012 ± 0.009 on all three aggregators, while ALNS did not complete within time limits and therefore has no entries at this scale.

Instance level comparisons and effect sizes confirm these results. As summarized in Table 4.11, DRL lost to CP-SAT on every large instance comparison (win-tie-loss 0-0-4 for WS, PA, and Chebyshev). Table 4.10 reports Cliff's $\delta = -1.00$ versus CP-SAT on all three aggregators, indicating complete separation in favor of CP-SAT (the displayed $p = 1.0$ reflects the small sample Mann-Whitney calculation with $n = 4$ per group). Against Greedy NN, DRL dominated with 4-0-0 across WS/PA/Cheb and Cliff's $\delta = 1.00$ with $p = 0.0625$; comparisons with ALNS are unavailable due to timeout.

These composite outcomes are consistent with the construction of GCS_{10} (Table 4.12). Because DRL's visit fulfillment was much lower than CP-SAT's on the large instances, the heavy coverage weight translated into lower composite scores for

DRL despite the presence of secondary objectives in the index.

We note that **H3** is scoped to small and medium problem sizes. The large instance results are reported here for completeness.

Table 4.9: Comprehensive GCS_{10} Summary: Mean ± SD Across Problem Sizes and Aggregators

	Small			Medium			Large		
Method	WS	PA	Cheb	WS	PA	Cheb	WS	PA	Cheb
DRL	0.910±0.077	0.907±0.087	0.895±0.080	0.828±0.081	0.821±0.081	0.824±0.047	0.465±0.043	0.465±0.047	0.465±0.047
CP-SAT	0.740±0.103	0.740±0.103	0.727±0.101	0.817±0.088	0.818±0.089	0.816±0.089	0.806±0.105	0.804±0.106	0.799±0.106
ALNS	0.158±0.030	0.159±0.033	0.156±0.034	0.197±0.034	0.198±0.034	0.199±0.034	-	-	-
NNS	0.020±0.014	0.020±0.014	0.020±0.014	0.038±0.032	0.038±0.032	0.038±0.032	0.012±0.009	0.012±0.009	0.012±0.009

Values represent Mean ± Standard Deviation of Comprehensive GCS_{10} (General Composite Score with 10 metrics)
Visit Fulfillment Rate weighted 100× higher than other metrics to reflect mission-critical importance (weight=100 vs 0.5-1.0 for others)
WS = Weighted Sum, PA = Policy Aware, Cheb = Chebyshev aggregation methods
Higher scores indicate better comprehensive performance (n=4 instances per condition)
ALNS results unavailable for Large instances due to computational timeout

Table 4.10: Statistical Significance for Comprehensive GCS_{10}: DRL vs Baselines (p-value / Cliff's δ)

	Small			Medium			Large		
Comparison	WS	PA	Cheb	WS	PA	Cheb	WS	PA	Cheb
vs CP-SAT	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.6875/-0.5	0.6875/-0.5	0.5625/0.0	1.0/-1.0	1.0/-1.0	1.0/-1.0
vs ALNS	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.0625/1.0	-	-	-
vs NNS	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.0625/1.0	0.0625/1.0

Format: p-value / Cliff's δ. Bold indicates $p < 0.05$ with $|\delta| \geq 0.474$ (large effect). ALNS excluded from Large instances due to time out.

Table 4.11: Win-Tie-Loss Summary for Comprehensive GCS_{10}: DRL vs Baselines (Instance-Level Comparisons)

	Small			Medium			Large		
Comparison	WS	PA	Cheb	WS	PA	Cheb	WS	PA	Cheb
vs CP-SAT	4-0-0	4-0-0	4-0-0	1-0-3	1-0-3	2-0-2	0-0-4	0-0-4	0-0-4
vs ALNS	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	-	-	-
vs NNS	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0	4-0-0

Format: W-T-L (Wins-Ties-Losses) from pairwise instance comparisons. ALNS failed to complete Large instances within 10-hour computational budget. RQ3 hypothesis scoped to Small and Medium instances.

These results confirm Hypothesis 3. The deep reinforcement learning policy's mean Comprehensive GCS_{10} will exceed both exact optimization and constructive heuristic methods on small and medium problem sizes.

Table 4.12: Comprehensive GCS_{10} Metric Definition: Component Metrics, Types, Optimization Direction, and Weights

GCS Type	Component Metric	Type	Direction	Weight
Comprehensive Score (GCS_{10})				
	Visit Fulfillment Rate	efficiency	maximize	**100.0**
	Mean Travel Time	efficiency	minimize	1.0
	Total Travel Time	efficiency	minimize	0.5
	Average Utilization Rate	efficiency	maximize	1.0
	Workload Fairness (Gini)	equity	minimize	1.0
	Workload Variance	equity	minimize	0.5
	Continuity of Care	quality	maximize	1.0
	Patient Preference Satisfaction	quality	maximize	1.0
	Average Idle Rate	operational	minimize	1.0
	Schedule Density	operational	maximize	1.0

GCS = General Composite Score; subscript indicates number of component metrics (n=10)
Comprehensive GCS_{10} aggregates 10 metrics across efficiency (4), equity (2), quality (2), and operational (2) dimensions
Visit Fulfillment Rate is weighted 100× higher than other metrics to reflect its mission-critical importance in home health care scheduling, where low fulfillment rates (e.g., ¡70%) result in unacceptable patient care gaps
Each GCS is computed using one of three aggregation methods: Weighted Sum (WS), Policy Aware (PA), or Chebyshev (Cheb)
Lower weights (0.5) reduce contribution of correlated metrics (Total Travel Time with Mean Travel Time; Workload Variance with Gini Coefficient)

Figure 4.10: Operational Balance - Metrics Components - Weighted Sum (Small)

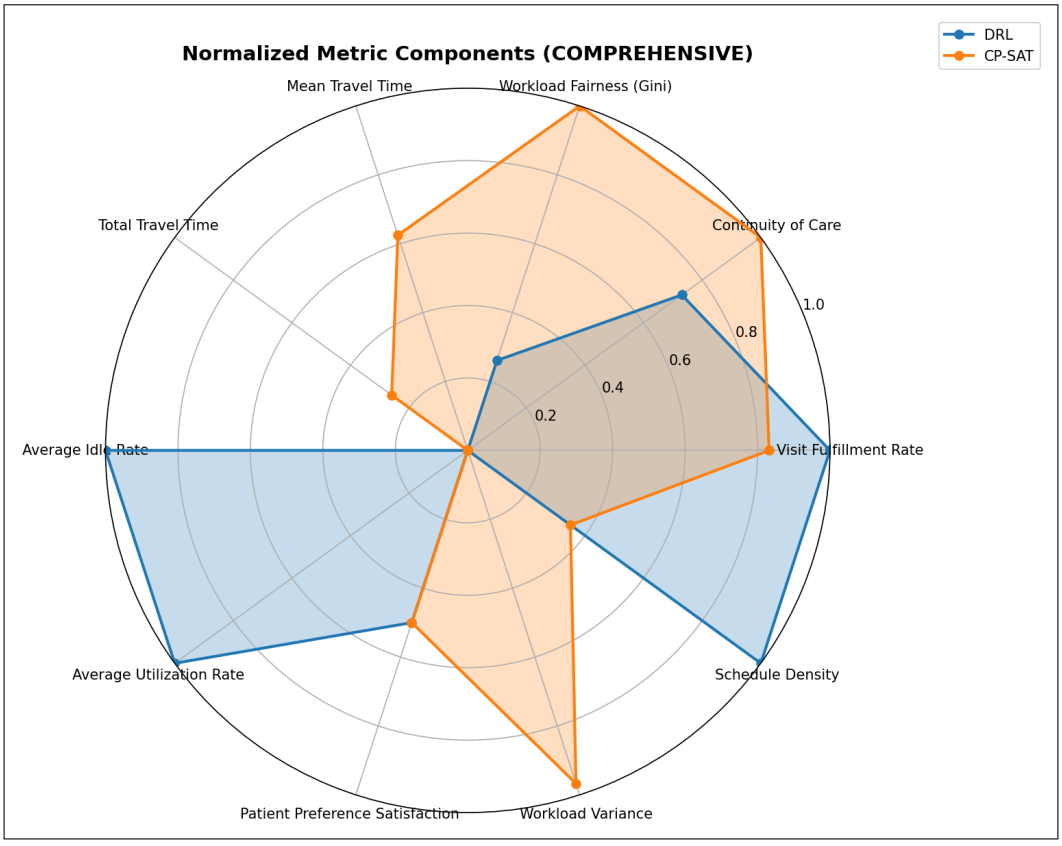


Figure 4.11: Operational Balance - Metrics Components - Policy Aware (Small)

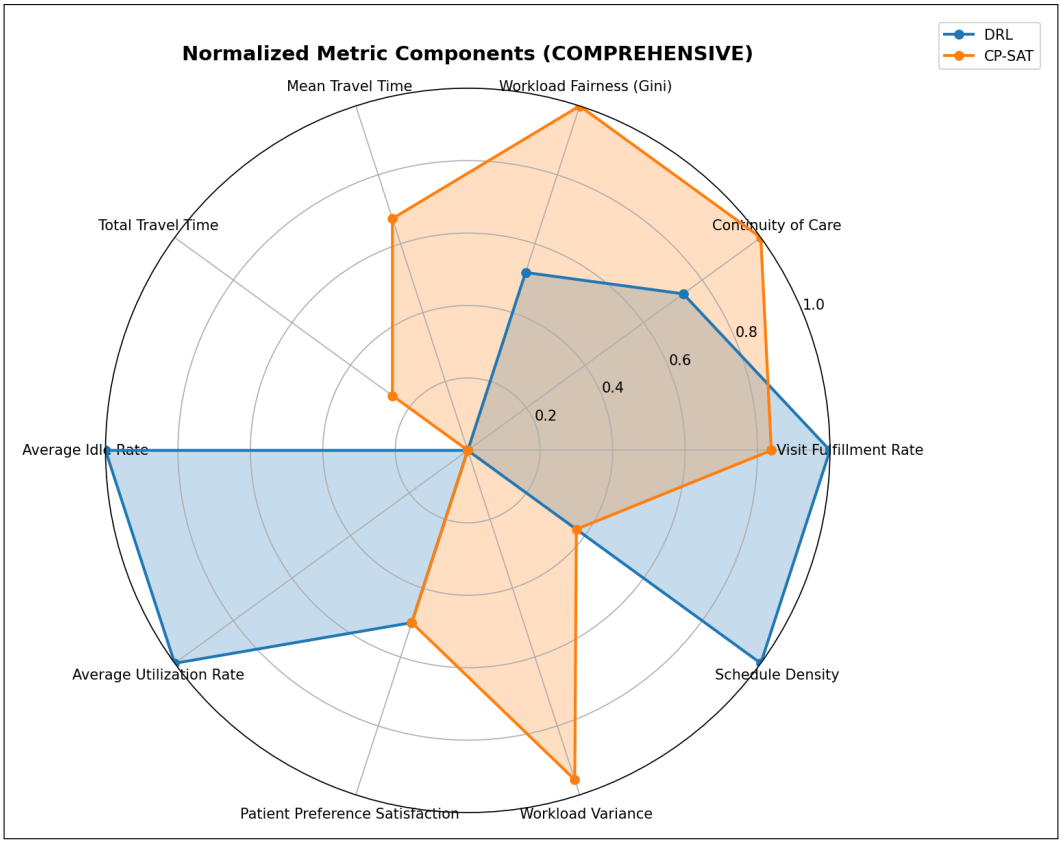


Figure 4.12: Operational Balance - Metrics Components - Chebyshev (Small)

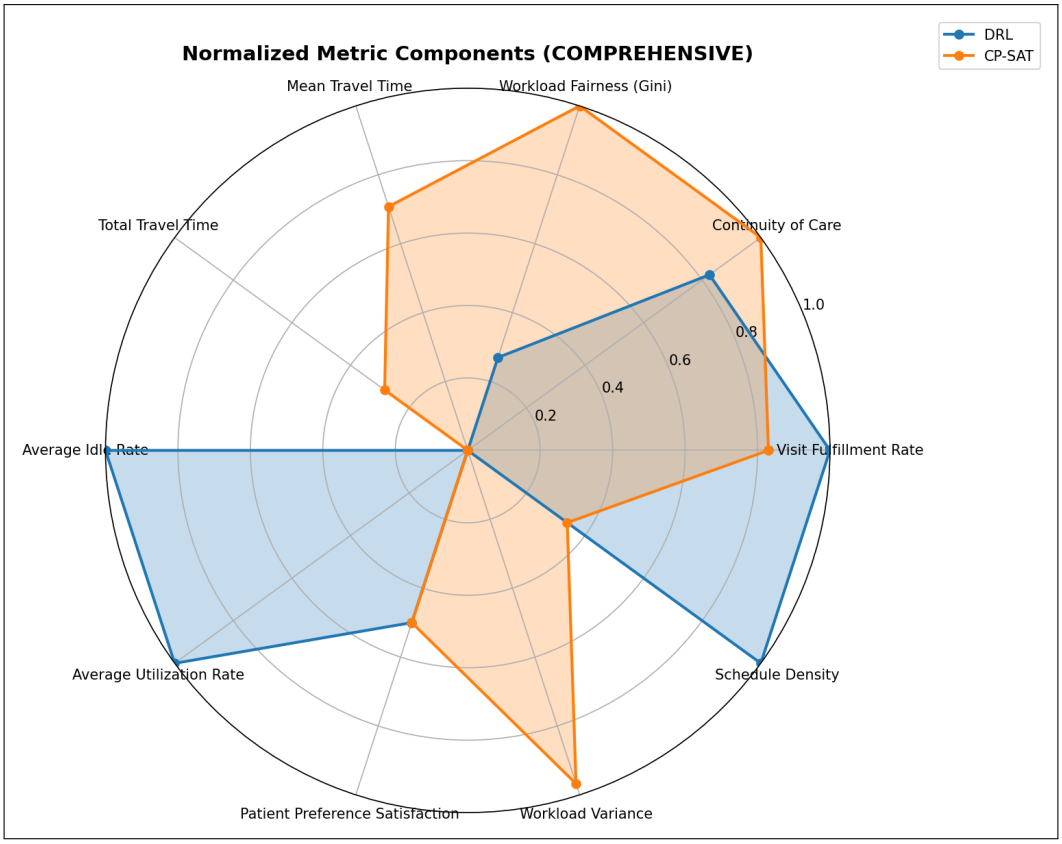


Figure 4.13: Operational Balance - Metrics Components - Weighted Sum (Medium)

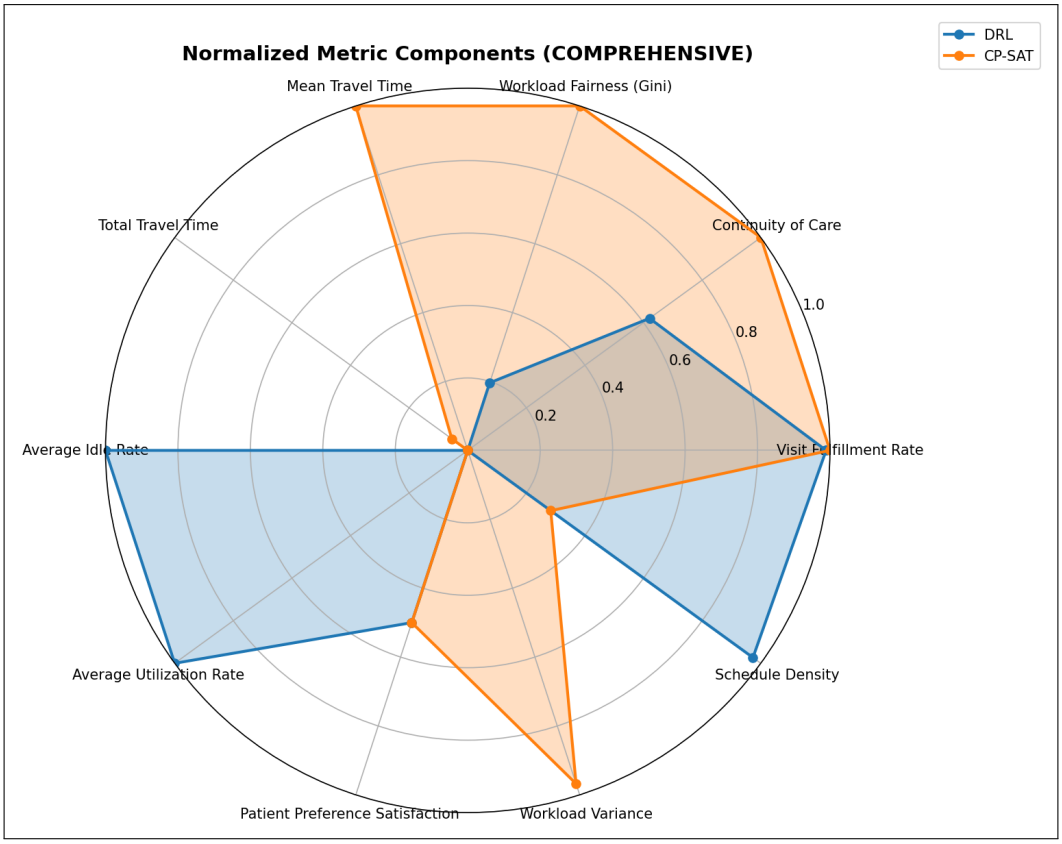


Figure 4.14: Operational Balance - Metrics Components - Policy Aware (Medium)

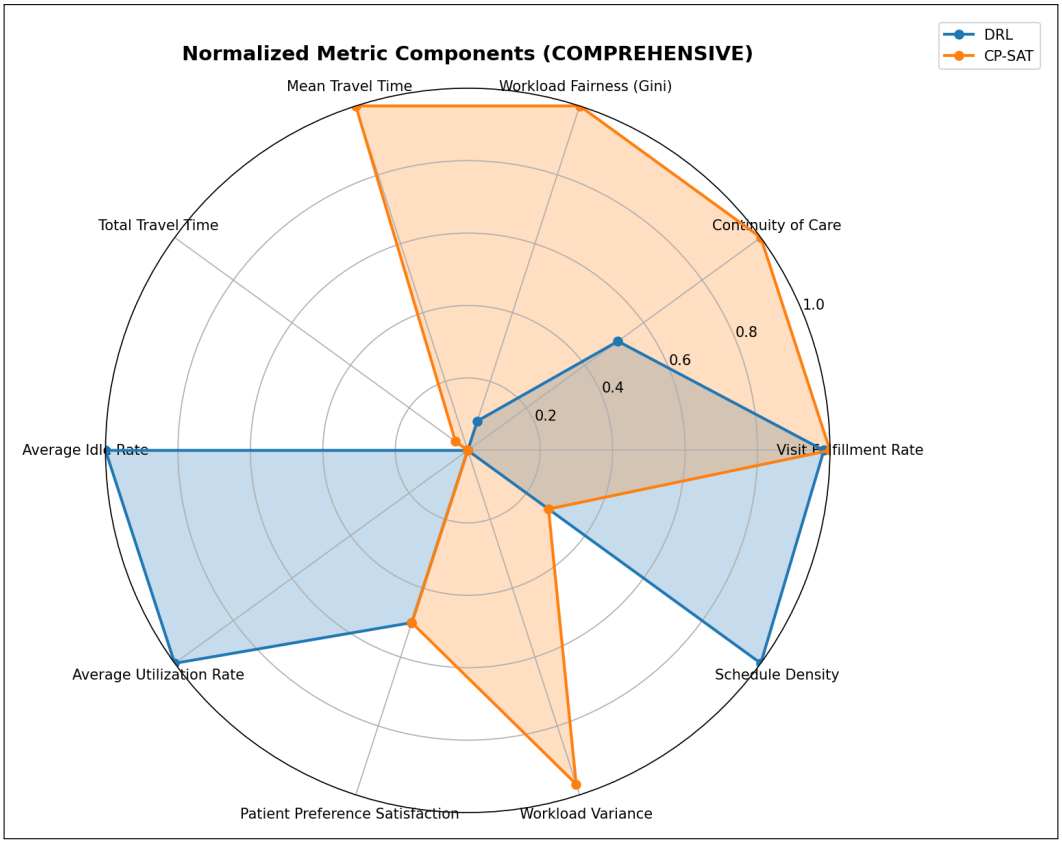


Figure 4.15: Operational Balance - Metrics Components - Chebyshev (Medium)

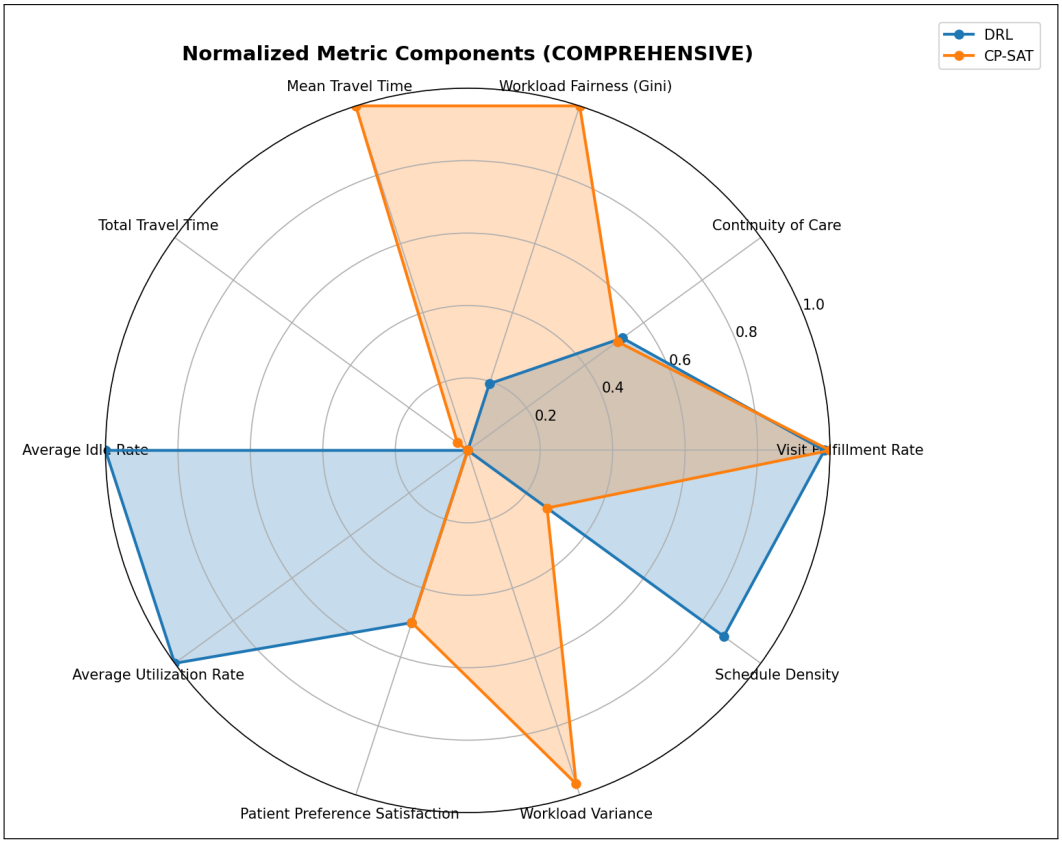


Figure 4.16: Operational Balance - Metrics Components - Weighted Sum (Large)

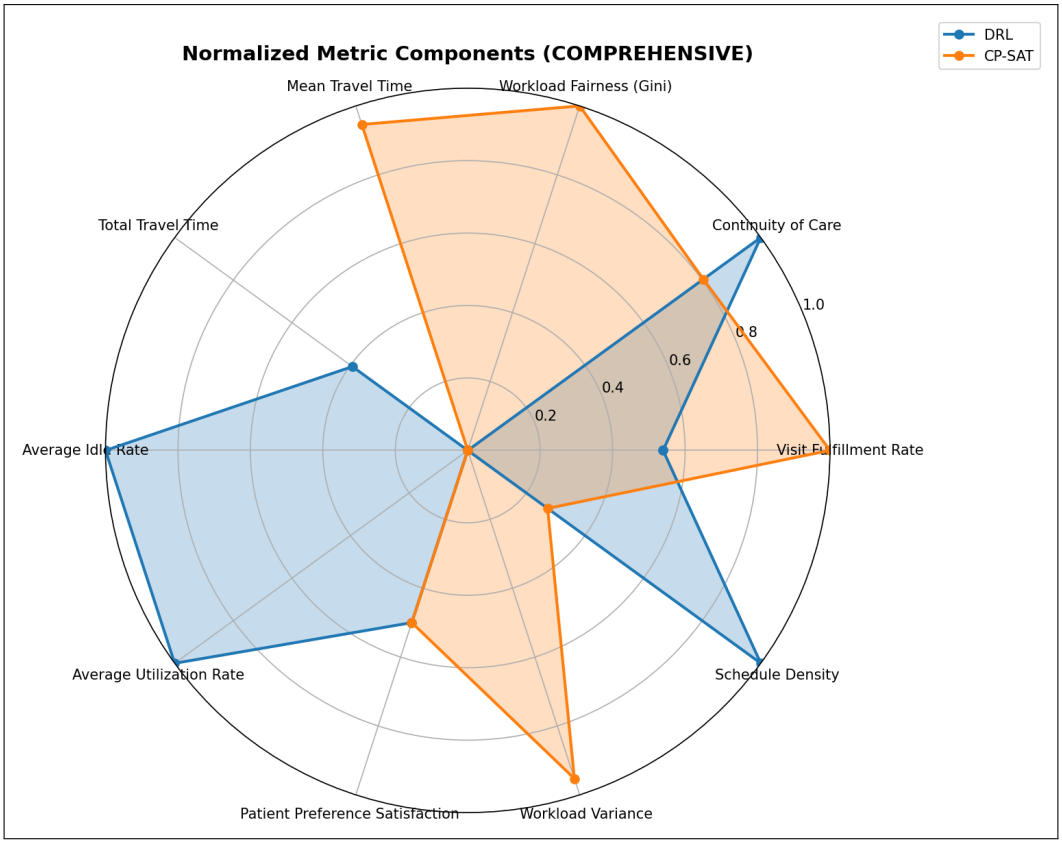


Figure 4.17: Operational Balance - Metrics Components - Policy Aware (Large)

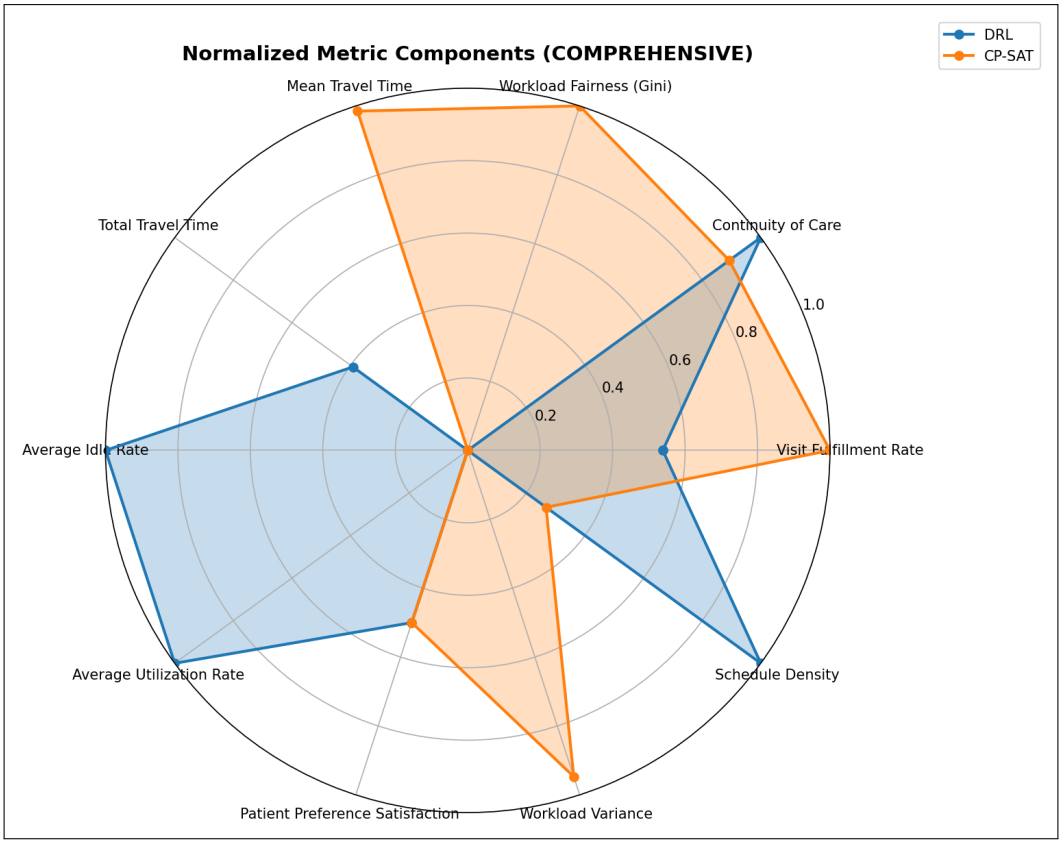


Figure 4.18: Operational Balance - Metrics Components - Chebyshev (Large)

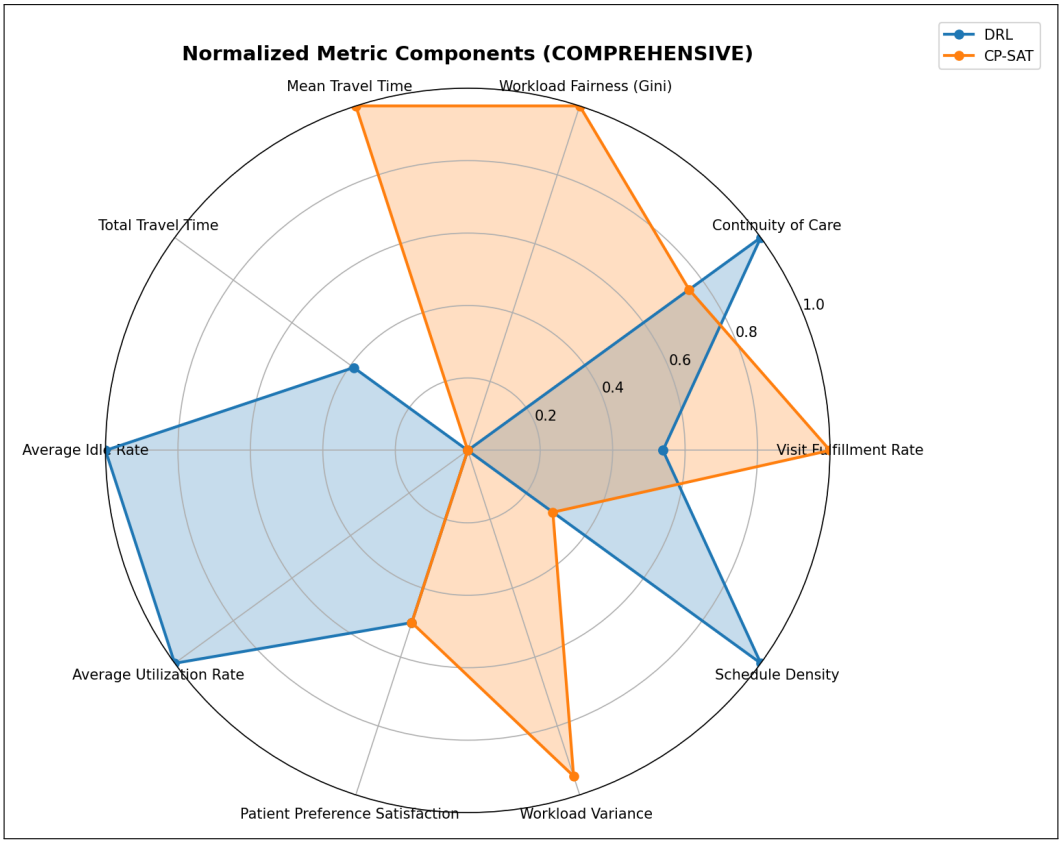


Figure 4.19: Operational Balance — GCS_{10} - Weighted Sum (Small)

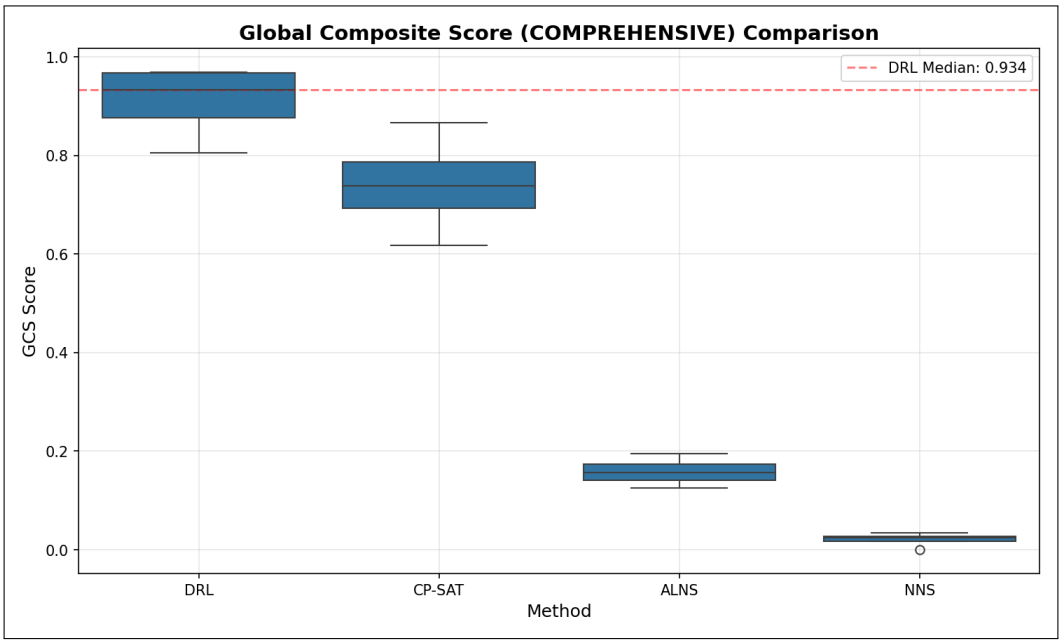


Figure 4.20: Operational Balance — GCS_{10} - Policy Aware (Small)

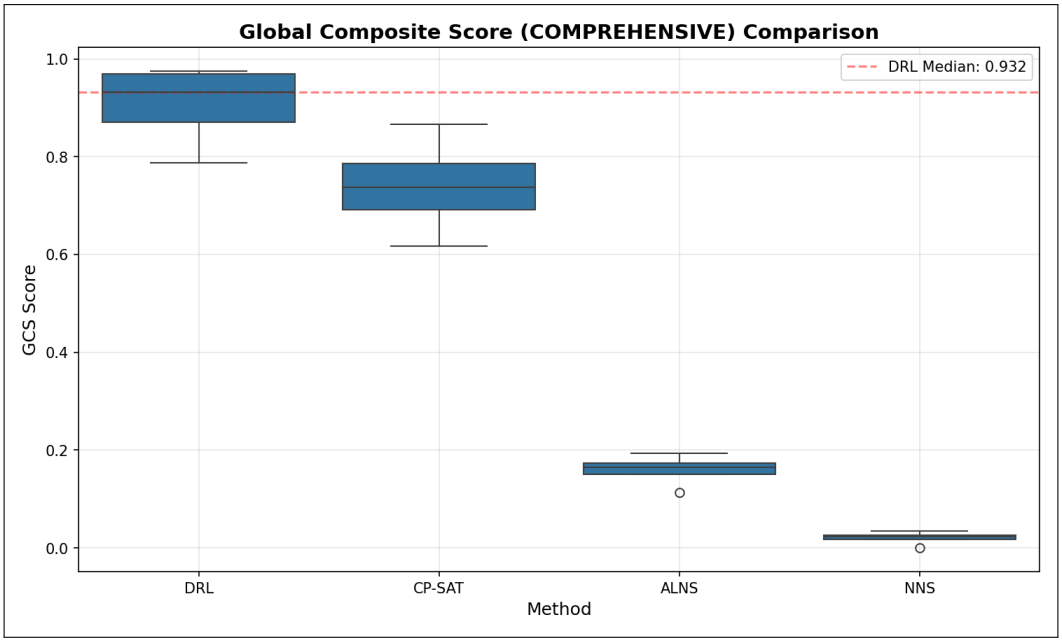


Figure 4.21: Operational Balance — GCS_{10} - Chebyshev (Small)

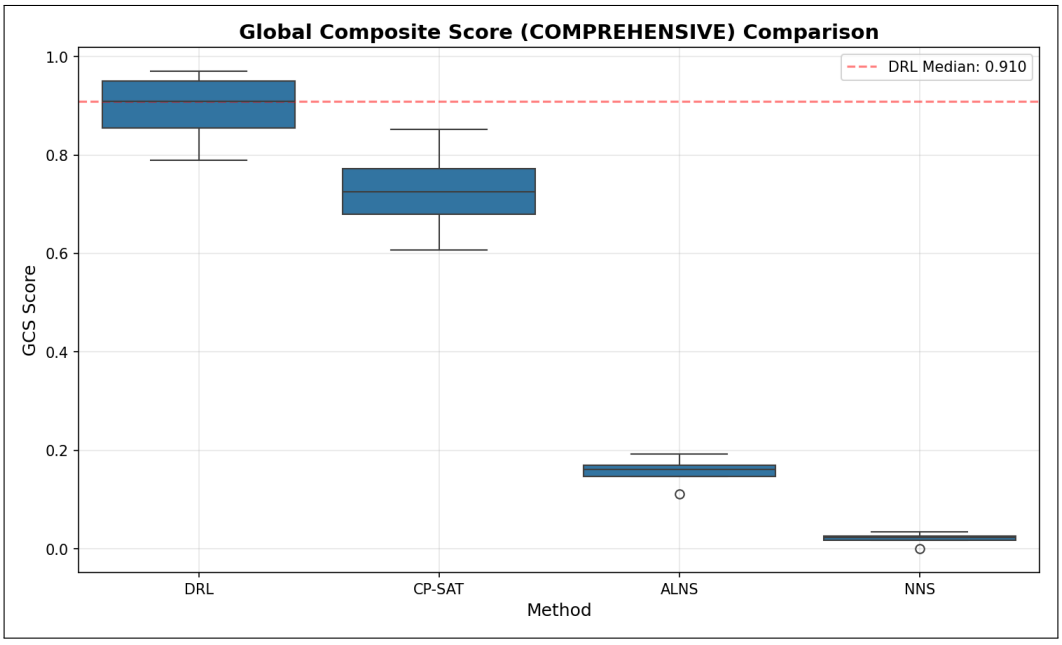


Figure 4.22: Operational Balance — GCS_{10} - Weighted Sum (Medium)

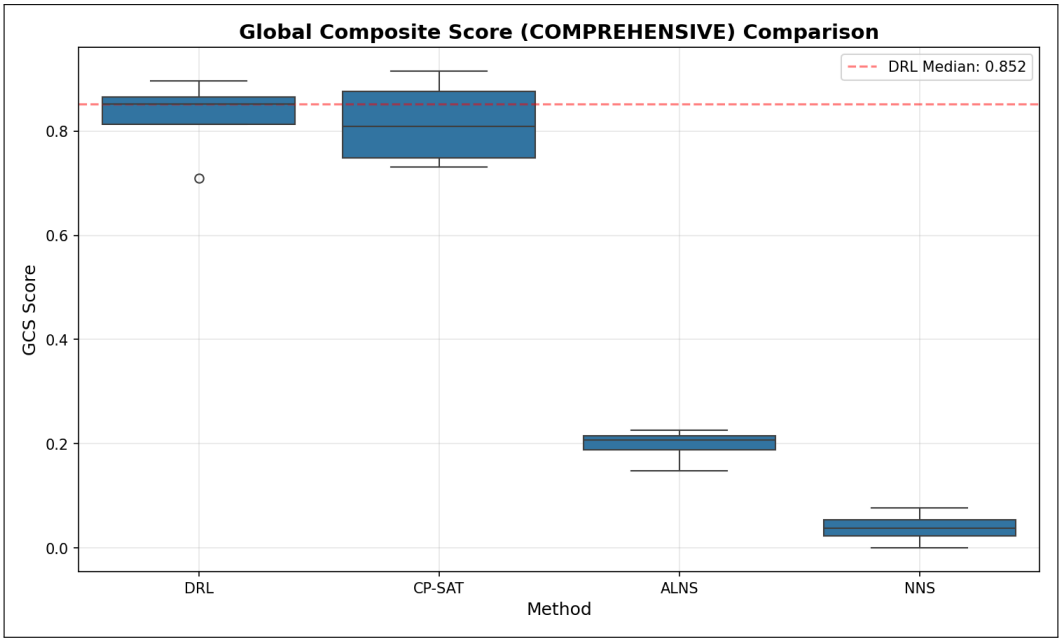


Figure 4.23: Operational Balance — GCS_{10} - Policy Aware (Medium)

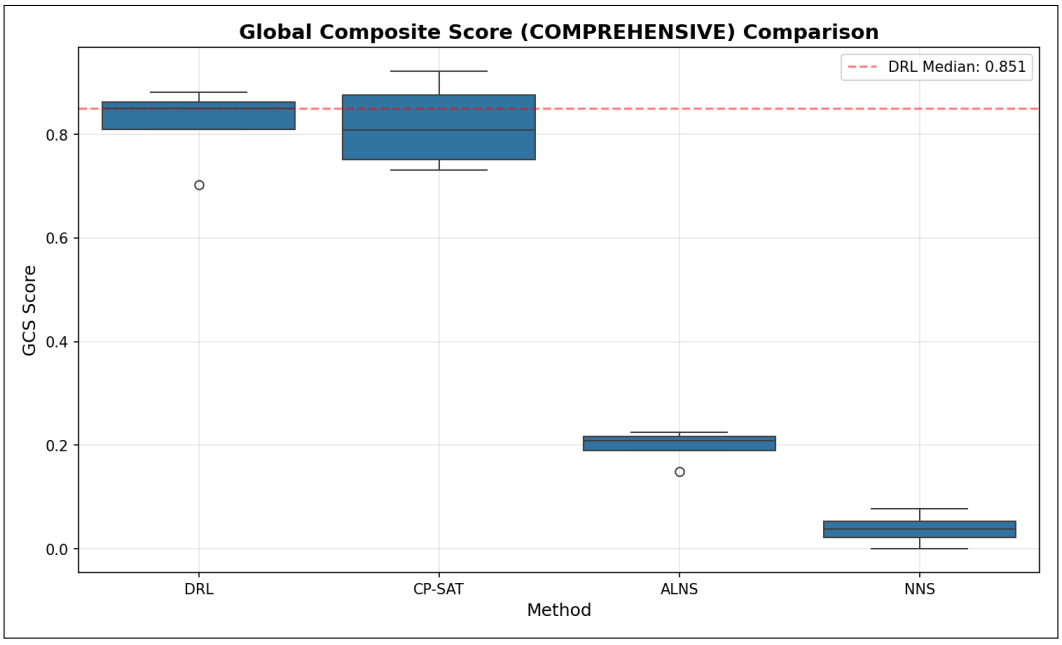


Figure 4.24: Operational Balance — GCS_{10} - Chebyshev (Medium)

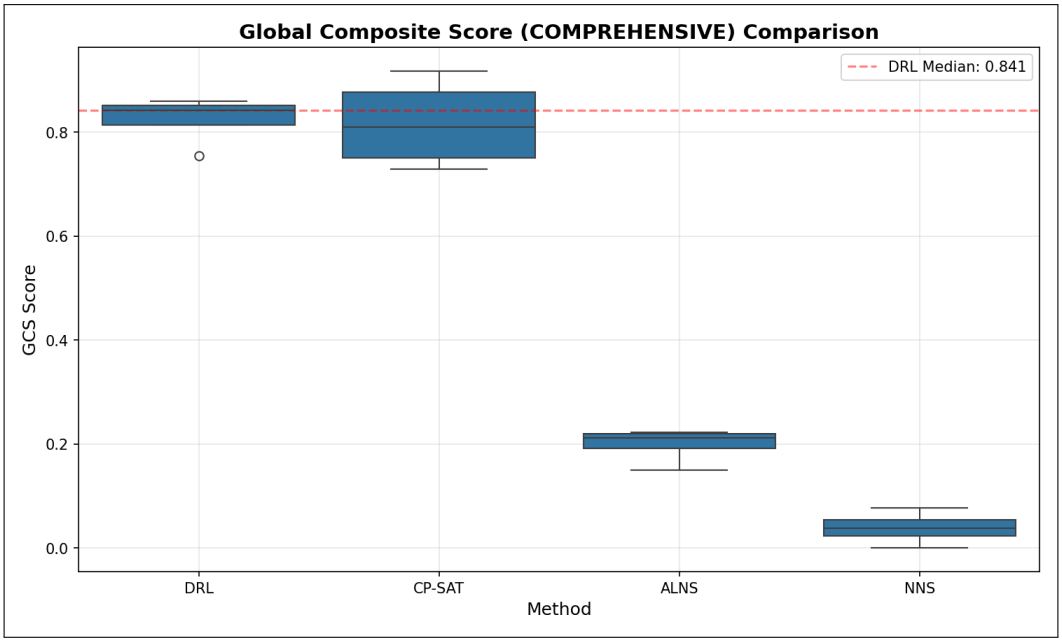


Figure 4.25: Operational Balance — GCS_{10} - Weighted Sum (Large)

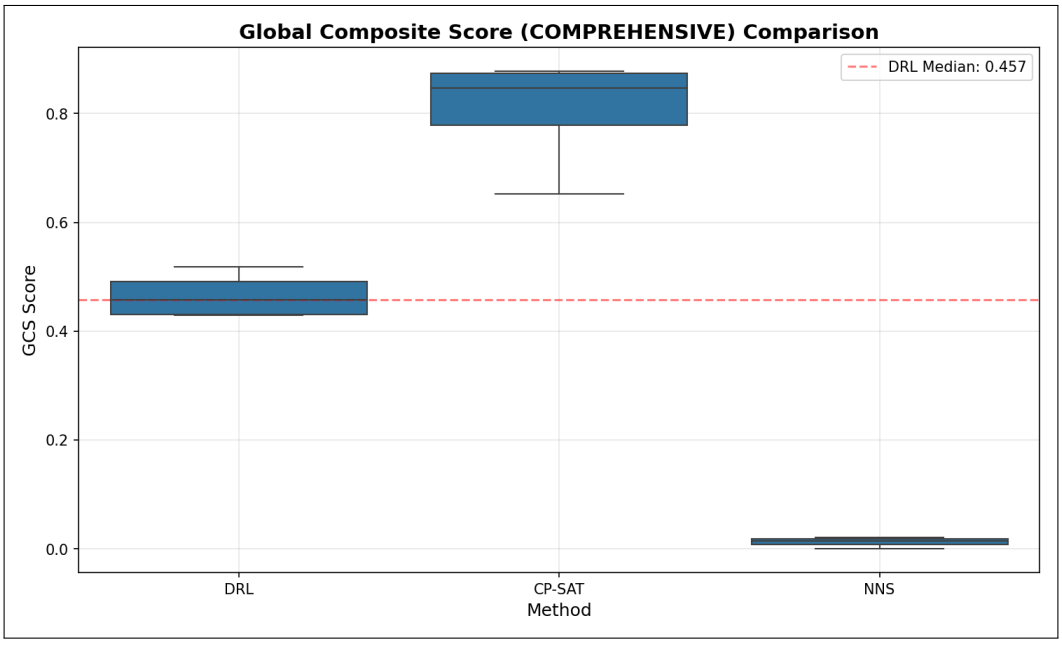


Figure 4.26: Operational Balance — GCS_{10} - Policy Aware (Large)

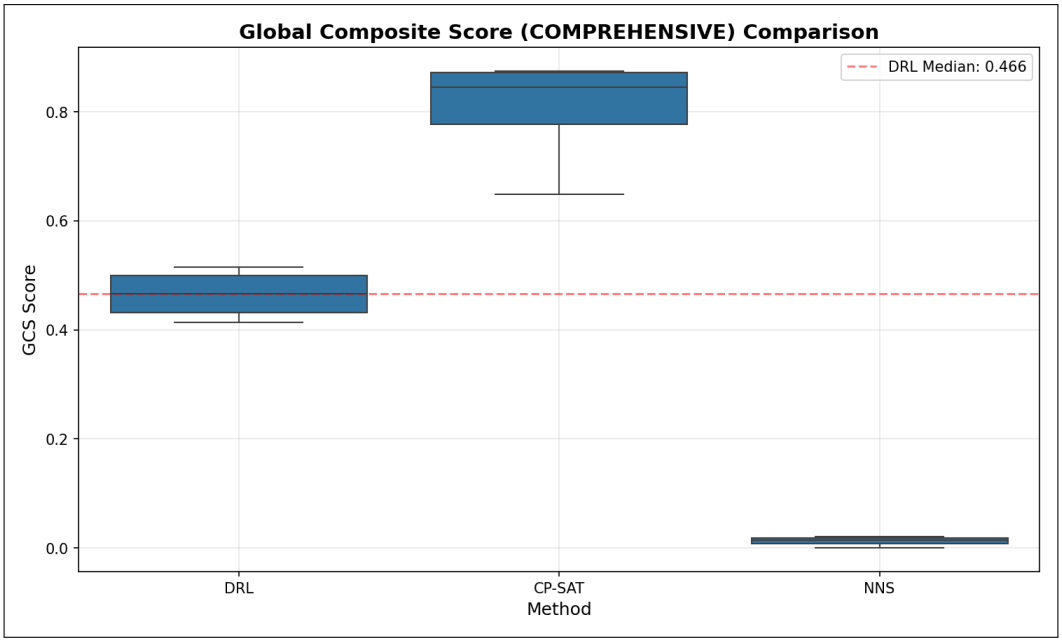
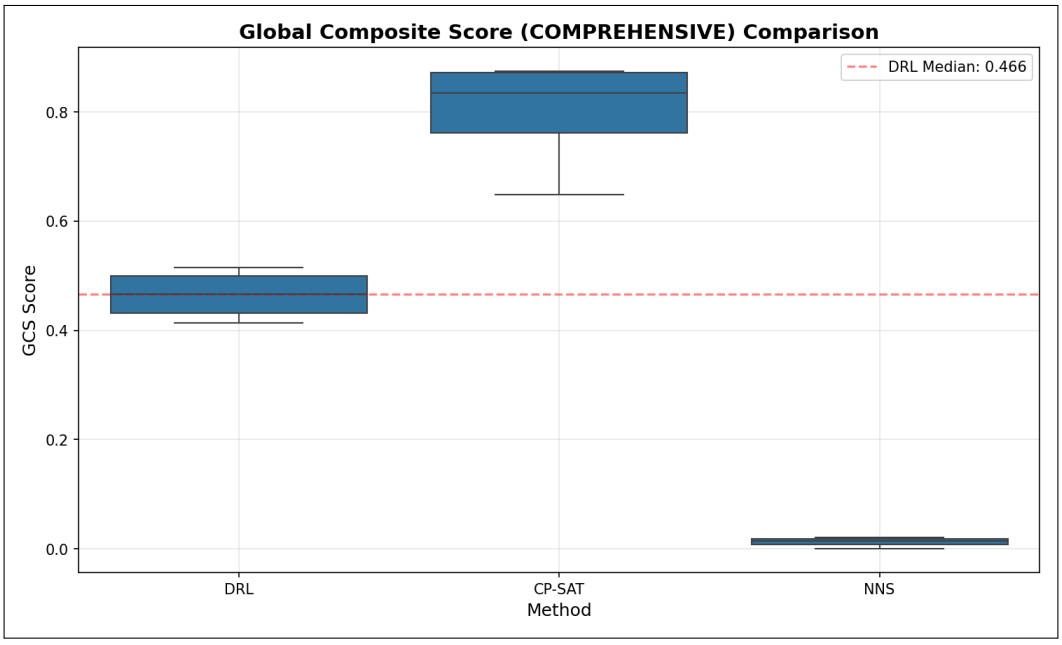


Figure 4.27: Operational Balance — GCS_{10} - Chebyshev (Large)



Chapter 5 – Conclusions

5.1 Summary of Findings

Home health care is a heavily regulated industry with on-going challenges related to workforce shortages, patient retention, compliance, burnout, and the increasing complexity of patient care. The industry needs a strong routing and scheduling system to mitigate some of these issues. In particular, a scheduling system that factors in the many objectives faced by home health agencies.

This Praxis demonstrates that deep reinforcement learning (DRL) with a multiobjective reward design, can achieve this result. The DRL model performs as well as or better than traditional optimization methods in visit fulfillment while simultaneously improving workload equity and operational efficiency.

Three research questions guided this study. The first examined whether a multiobjective DRL policy could achieve comparable visit fulfillment to single-objective benchmarks under identical constraints. On small-scale problems, DRL achieved 96% fulfillment compared with CP-SAT's 81%, a 15-point improvement and full separation across test cases. For medium-scale problems, both converged around 79-80%, confirming that when reward functions emphasize fulfillment, DRL maintains competitive primary performance while still balancing secondary goals. These results show that careful reward design, rather than the removal of trade-offs, allows DRL to sustain strong coverage while integrating secondary objectives. The large-scale instances exposed the expected conflict between objectives as clinician scarcity made trade-offs more visible.

The second question focused on Pareto frontier quality across fulfillment, idle rate, and utilization. DRL achieved between 6× and 140× higher hypervolume than all benchmarks across nine test scenarios ($\alpha = 0.05$, large effect sizes). DRL completely dominated at small scale (C-metric = 1.0), maintained a strong advantage at medium scale, and contributed complementary solutions at large scale. This finding suggests

that hybrid approaches that combine DRL with classical methods could extend visit fulfillment more effectively than either method alone.

The third question evaluated overall performance through a composite score of ten metrics weighted by operational priorities. Visit fulfillment carried 100× the weight of secondary objectives, reflecting its importance in patient care. On small instances, DRL scored 0.90 versus CP-SAT's 0.74. On medium instances, both scored 0.82, showing that DRL maintains comparable performance even as constraints increased. Against constructive heuristics and metaheuristics, DRL consistently outperformed across all scales.

The findings also have broader implications for how optimization systems can support patient and clinician focused scheduling beyond the single objective of visit fulfillment. Traditional optimization typically focuses on one primary goal, such as fulfillment, and treats others as penalties or post-processing adjustments. This sequential approach struggles when objectives interact. For instance, improving workload equity reduces travel efficiency. In contrast, DRL integrates these competing goals within a single learning framework using weighted rewards. The method does not eliminate trade-offs but allows them to be managed transparently according to chosen priorities. Large-scale results highlight this challenge that when the policy overemphasized balance early in training, fulfillment decreased under severe resource shortages.

Although large-instance training was incomplete, the undertrained model achieved 53% fulfillment with only 1.6% of the full training budget. Further training or considerations such as hierarchical decomposition or hybrid solver integration could close this gap.

This work advances reinforcement learning for combinatorial optimization in three key ways. First, the constraint masking framework confines learning to legal actions, reducing the credit assignment noise that often destabilizes RL in constrained prob-

lems. Second, the modular reward architecture shows that multiple objectives can be expressed through decomposed scalar components with relative weighting. DRL maintained strong results across all three aggregation strategies i.e. Weighted Sum, Policy-Aware, and Chebyshev. Third, the results show that reward weight configuration directly determines which objectives the policy prioritizes, allowing transparent adjustment of trade-offs between fulfillment and fairness.

5.2 Practical Applications

The practical value lies in operational costs. More efficient routes mean lower fuel consumption and less wear and tear on vehicles. These savings can translate to clinicians spending less time driving and more time with patients. A fair workload distribution supports retention, reduces or eliminates replacement recruiting, training, and productivity losses. Administrative efficiencies compound these benefits. For example, schedulers can dedicate the extra time to patient care, quality control, or other tasks. Organizations managing typical caseloads of 50-80 patients with 7-10 clinicians represented the operational scale at which this DRL model will operate the best.

5.3 Limitations

The evaluation used a single benchmark dataset of synthetic scheduling instances to simulate home health care conditions. Although these instances capture realistic workload and timing patterns, they are not based on actual home health care data. In practical home health care settings, geographic diversity, varying patient needs, cultural expectations, and organizational policies can create complexities that simulations cannot fully represent.

Reward weights play a key role in shaping policy behavior. In this study, fulfillment received a substantially higher weight relative to secondary objectives. This

reflects how real home health agencies often prioritize patient care above all else. But it represents only one point in a broader design space. Different organizations may prefer different trade-offs. For instance, teams facing clinician shortages may choose to sacrifice some fulfillment in favor of better workload balance and retention. Running sensitivity analyses with varied weight configurations would help confirm that the approach generalizes beyond this specific setup. However, trade-offs will always exist because no single weighting scheme can optimize all objectives simultaneously.

Training cost is another important limitation. Full convergence required five independent runs of two million timesteps each and demanded computational resources beyond what some smaller home health agencies care organizations can easily access. While inference is computationally light once the model is trained, the initial training investment is substantial. Reducing this burden may require techniques such as transfer learning and curriculum training to cut down the number of samples needed for convergence. Even then, policies trained on one dataset cannot simply be fine-tuned for a new organization. Because the constraint structure and feature encoding are tightly linked to each problem's specific setup, retraining would be necessary for each new deployment.

The current model assumes a single centralized decision-maker, which limits scalability. In reality, many home health agencies use decentralized scheduling or allow clinicians to manage their own schedules. Extending the architecture to multi-agent systems could improve scalability and responsiveness.

5.4 Future Research

Future work should focus on making the approach more practical, scalable, and adaptable to real-world conditions. Several research directions emerge from the findings.

5.4.1 Hybrid optimization.

Combining reinforcement learning with exact optimization could offer the best of both worlds. A hybrid system might use DRL to learn flexible heuristics for manageable subproblems while relying on CP-SAT or similar solvers to guarantee coverage when resources are tight. This division of labor plays to each method's strengths instead of trying to make one approach dominate all cases.

5.4.2 Adaptive weighting.

Dynamic reward weights could make the policy more flexible. If the model detects low fulfillment, it could automatically increase the weight on coverage to avoid over-balancing secondary objectives too soon. This kind of meta-learning would let a single policy adjust priorities as conditions change. However, it would also require careful design to prevent instability or oscillation between objectives.

5.4.3 Curriculum learning.

Curriculum learning can help by starting with easier problems and gradually moving to harder ones. This approach can speed up training and improve performance on large or complex cases. Gradually exposing the DRL to resource scarcity lets it learn better prioritization and decision-making skills, much like how human schedulers gain experience over time instead of facing the toughest situations right away.

5.4.4 Transfer learning.

Future studies should explore how well policies trained on one region or patient population adapt to others. If a model trained in one context can generalize with minimal retraining, training costs could be shared across multiple deployments. Identifying which parts of the learned policy transfer well would guide practical deployment.

5.4.5 Explainability.

For real-world adoption, schedulers need to understand the model's reasoning. Techniques such as attention visualization, counterfactual explanations, or extraction of interpretable rules could make neural policies more transparent. Decision support tools that can explain "why this clinician, for this patient, at this time" will inspire more trust than opaque systems that simply output results.

5.4.6 Field validation.

Finally, real-world studies are essential. Simulation shows algorithmic capability, but deployment will reveal how human schedulers interact with the system, how disruptions are handled, and how long-term factors such as job satisfaction and patient outcomes are affected.

5.5 Concluding Remarks

This work presents a validated framework for applying deep reinforcement learning to the home health routing and scheduling problem. It shows that learning-based systems with well-designed reward structures can balance competing objectives within a single model. Traditional optimization methods handle these objectives one at a time, but DRL manages them together. Although trade-offs remain and weight configuration strongly shapes outcomes, the results demonstrate that multiobjective learning is feasible and effective.

As home health care grows in demand and workforce pressures intensify, adaptive learning systems will become increasingly necessary. The methods developed here provide a foundation for that shift. The path forward involves hybrid architectures, adaptive weighting, curriculum learning, and field validation.

Bibliography

Ala, A., Simic, V., Pamucar, D., and Tirkolaee, E. B. (2022). Appointment scheduling problem under fairness policy in healthcare services: Fuzzy ant lion optimizer. *Expert Systems with Applications*, 207:117949.

Assaf, G., Löffler, S., and Hofstedt, P. (2024). Constraint-based optimization for scheduling medical appointments. In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025)*, pages 94–103. SciTePress.

Bennett, A. R. and Erera, A. L. (2011). Dynamic periodic fixed appointment scheduling for home health. *IIE Transactions on Healthcare Systems Engineering*, 1(1):6–19.

Cappanera, L. and Scutellà, M. G. (2015). Joint assignment, scheduling, and routing models to home care optimization: A pattern-based approach. *Transportation Science*, 49(4):830–852.

Clapper, Y., Bekker, R., Berkhout, J., and Moeke, D. (2024). Balancing continuity of care and home care schedule costs using blueprint routes. *Operations Research for Health Care*, 42:100441.

Dessevre, G., Martinez, C., Fontanili, F., Zhang, L., and Bortolaso, C. (2022). Simulation and analysis of disruptive events on a deterministic home health care routing and scheduling solution. In *Proceedings of the 2022 Winter Simulation Conference (WSC)*, pages 1069–1080. IEEE.

Devidze, R., Kamalaruban, P., and Singla, A. (2022). Exploration-guided reward shaping for reinforcement learning under sparse rewards. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural*

Information Processing Systems, volume 35, pages 5829–5842. Curran Associates, Inc.

Fathollahi-Fard, A. M., Moosavi, H. N., and Jolai, F. (2018). A bi-objective home health care routing and scheduling problem considering patients' satisfaction in a fuzzy environment. *Applied Soft Computing*, 67:734–747.

Fikar, C. and Hirsch, P. (2017). Home health care routing and scheduling: A review. *Computers & Operations Research*, 77:86–95.

Fikar, C. and Hirsch, P. (2018). Evaluation of trip and car sharing concepts for home health care services. *Flexible Services and Manufacturing Journal*, 30(1):78–97.

Fu, Y., Ma, X., Gao, K., Li, Z., and Dong, H. (2024). Multi-objective home health care routing and scheduling with sharing service via a problem-specific knowledge-based artificial bee colony algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 25(2):1706–1719.

Gupta, A., Pacchiano, A., Zhai, Y., Kakade, S. M., and Levine, S. (2022). Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity.

Han, M., Zhang, L., Wang, J., and Pan, W. (2020). Actor-critic reinforcement learning for control with stability guarantee.

Hermelin, D., Molter, H., Niedermeier, R., Pinedo, M., and Shabtay, D. (2025). Fairness in repetitive scheduling. *European Journal of Operational Research*, 323(3):724–738.

Kirk, R., Zhang, A., Grefenstette, E., and Rocktäschel, T. (2023). A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76:201–264.

Korkmaz, E. (2024). A survey analyzing generalization in deep reinforcement learning.

Ma, H., Guan, Y., Li, S. E., Zhang, X., Zheng, S., and Chen, J. (2021). Feasible actor-critic: Constrained reinforcement learning for ensuring statewise safety.

Miller, K. E., Green, C. M., Fassinger, A., and Wolff, J. L. (2025). Long-term care services and supports needed for successful aging-in-place: A critical review. *Annual Review of Public Health*, 46:487–505.

Nazari, M., Oroojlooy, A., Takáč, M., and Snyder, L. V. (2018). Reinforcement learning for solving the vehicle routing problem.

Pan, W. and Liu, S. Q. (2023). Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence*, 53:405–422.

Pereira Gray, D. J., Sidaway-Lee, K., White, E., Thorne, A., and Evans, P. H. (2018). Continuity of care with doctors—a matter of life and death? a systematic review of continuity of care and mortality. *BMJ Open*, 8(6):e021161.

Rasmussen, M. S., Justesen, T., Dohn, A., and Larsen, J. (2012). The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3):598–610.

Salbanya, B., Unceta, I., and Nin, J. (2023). Resource allocation in home care services using reinforcement learning. In *Frontiers in Artificial Intelligence and Applications*, volume 370, pages 165–174. IOS Press.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

Sun, H., Han, L., Yang, R., Ma, X., Guo, J., and Zhou, B. (2022). Exploit reward shifting in value-based deep-rl: Optimistic curiosity-based exploration and conservative exploitation via linear reward shaping. In Koyejo, S., Mohamed, S., Agarwal,

A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 37719–37734. Curran Associates, Inc.

Ta-Dinh, Q., Pham, T. S., Hà, M. H., and et al. (2024). A reinforcement learning approach for the online dynamic home health care scheduling problem. *Health Care Management Science*, 27:650–664.

U.S. Census Bureau (2022). Revenues for Home Care Elderly Services Increase. [Accessed: February 15, 2025].

Wu, Q., Han, J., Yan, Y., Kuo, Y.-H., and Shen, Z.-J. M. (2025). Reinforcement learning for healthcare operations management: methodological framework, recent developments, and future research directions. *Health Care Management Science*.

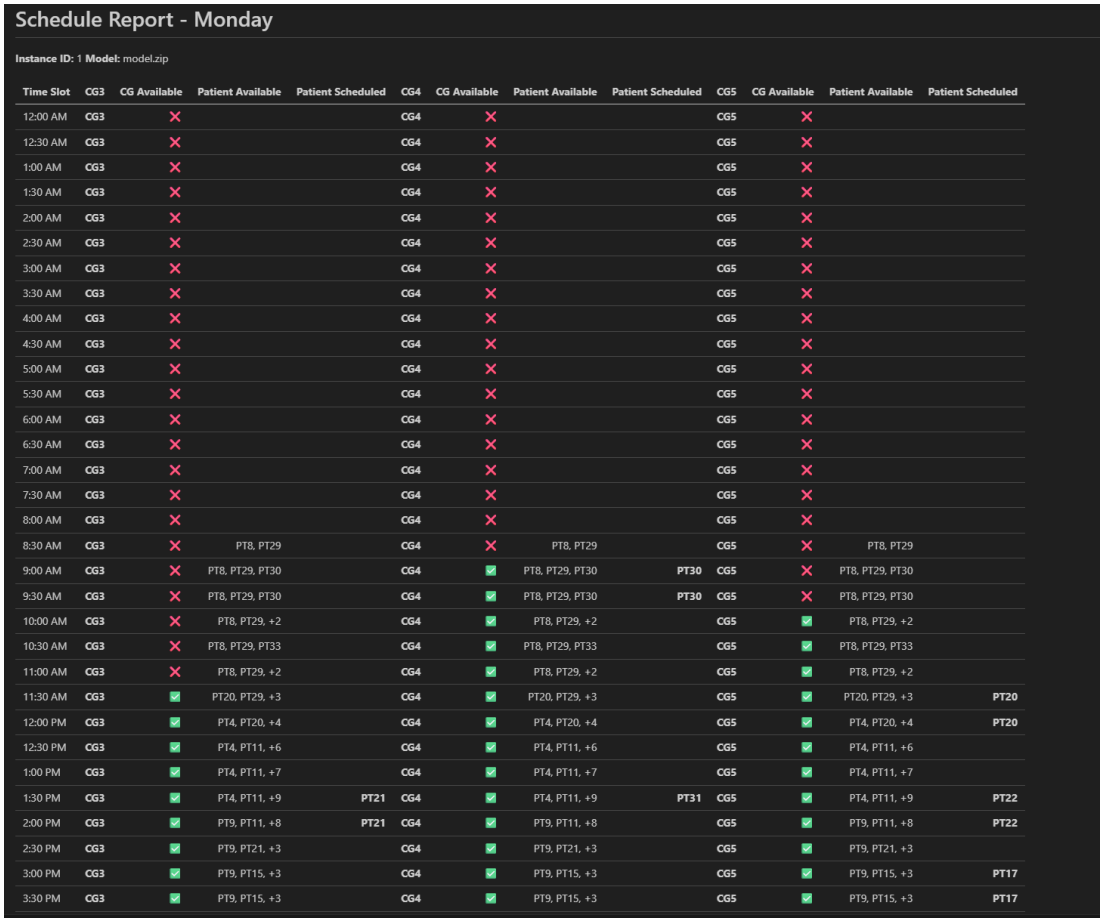
Yu, G., Siddique, U., and Weng, P. (2023). Fair deep reinforcement learning with preferential treatment. In *Frontiers in Artificial Intelligence and Applications*, volume 370, pages 15–26. IOS Press.

Zhang, T., Liu, Y., Yang, X., Chen, J., and Huang, J. (2023). Home health care routing and scheduling in densely populated communities considering complex human behaviours. *Computers and Industrial Engineering*, 182:109332.

Zhao, J., Wang, T., and Monteiro, T. (2024). A bi-objective home health care routing and scheduling problem under uncertainty. *International Journal of Environmental Research and Public Health*, 21(3):377.

Chapter A – Appendix A

Figure A.1: Sample Caregiver Schedule



Schedule Report - Monday

Instance ID: 1 **Model:** model.zip

Time Slot	CG3	CG Available	Patient Available	Patient Scheduled	CG4	CG Available	Patient Available	Patient Scheduled	CG5	CG Available	Patient Available	Patient Scheduled
12:00 AM	CG3	✗			CG4	✗			CG5	✗		
12:30 AM	CG3	✗			CG4	✗			CG5	✗		
1:00 AM	CG3	✗			CG4	✗			CG5	✗		
1:30 AM	CG3	✗			CG4	✗			CG5	✗		
2:00 AM	CG3	✗			CG4	✗			CG5	✗		
2:30 AM	CG3	✗			CG4	✗			CG5	✗		
3:00 AM	CG3	✗			CG4	✗			CG5	✗		
3:30 AM	CG3	✗			CG4	✗			CG5	✗		
4:00 AM	CG3	✗			CG4	✗			CG5	✗		
4:30 AM	CG3	✗			CG4	✗			CG5	✗		
5:00 AM	CG3	✗			CG4	✗			CG5	✗		
5:30 AM	CG3	✗			CG4	✗			CG5	✗		
6:00 AM	CG3	✗			CG4	✗			CG5	✗		
6:30 AM	CG3	✗			CG4	✗			CG5	✗		
7:00 AM	CG3	✗			CG4	✗			CG5	✗		
7:30 AM	CG3	✗			CG4	✗			CG5	✗		
8:00 AM	CG3	✗			CG4	✗			CG5	✗		
8:30 AM	CG3	✗	PT8, PT29		CG4	✗	PT8, PT29		CG5	✗	PT8, PT29	
9:00 AM	CG3	✗	PT8, PT29, PT30		CG4	✓	PT8, PT29, PT30	**PT30**	CG5	✗	PT8, PT29, PT30	
9:30 AM	CG3	✗	PT8, PT29, PT30		CG4	✓	PT8, PT29, PT30	**PT30**	CG5	✗	PT8, PT29, PT30	
10:00 AM	CG3	✗	PT8, PT29, +2		CG4	✓	PT8, PT29, +2		CG5	✓	PT8, PT29, +2	
10:30 AM	CG3	✗	PT8, PT29, PT33		CG4	✓	PT8, PT29, PT33		CG5	✓	PT8, PT29, PT33	
11:00 AM	CG3	✗	PT8, PT29, +2		CG4	✓	PT8, PT29, +2		CG5	✓	PT8, PT29, +2	
11:30 AM	CG3	✓	PT20, PT29, +3		CG4	✓	PT20, PT29, +3		CG5	✓	PT20, PT29, +3	**PT20**
12:00 PM	CG3	✓	PT4, PT20, +4		CG4	✓	PT4, PT20, +4		CG5	✓	PT4, PT20, +4	**PT20**
12:30 PM	CG3	✓	PT4, PT11, +6		CG4	✓	PT4, PT11, +6		CG5	✓	PT4, PT11, +6	
1:00 PM	CG3	✓	PT4, PT11, +7		CG4	✓	PT4, PT11, +7		CG5	✓	PT4, PT11, +7	
1:30 PM	CG3	✓	PT4, PT11, +9	**PT21**	CG4	✓	PT4, PT11, +9	**PT31**	CG5	✓	PT4, PT11, +9	**PT22**
2:00 PM	CG3	✓	PT9, PT11, +8	**PT21**	CG4	✓	PT9, PT11, +8		CG5	✓	PT9, PT11, +8	**PT22**
2:30 PM	CG3	✓	PT9, PT21, +3		CG4	✓	PT9, PT21, +3		CG5	✓	PT9, PT21, +3	
3:00 PM	CG3	✓	PT9, PT15, +3		CG4	✓	PT9, PT15, +3		CG5	✓	PT9, PT15, +3	**PT17**
3:30 PM	CG3	✓	PT9, PT15, +3		CG4	✓	PT9, PT15, +3		CG5	✓	PT9, PT15, +3	**PT17**

Legend:

- (Red X): Caregiver not available
- (Green Check): Caregiver available
- **Patient Available**: Patients available for scheduling at time slot t
- **Patient Scheduled**: Patient assigned to caregiver at time slot t

Note: Instance ID: The schedule report shows caregiver availability (CG3, CG4, CG5) and patient assignment patterns across time slots on Monday. Red X indicates when caregivers are unavailable (midnight to early morning), while green checkmarks show availability periods. The table displays which patients are available for scheduling and which patients have been assigned to specific time slots.