PyRebuild: A Python-Based Simulator for the Dynamic Post-Earthquake Housing Reconstruction Problem

QIANCHEN YU[1], IRENE ALISJAHBANA[2], and VIVIAN WONG[3]

[1]Department of Urban and Regional Planning, College of Design, Construction and Planning, University of Florida. Email: yu.qianchen@ufl.edu
[2]d.School, Stanford University. Email:ialsjbn@stanford.edu
[3]Department of Urban and Regional Planning, College of Design, Construction and Planning, University of Florida. Email: vivian.wong@ufl.edu

ABSTRACT

Post-earthquake housing reconstruction faces a key challenge: damage assessments are reported gradually over time, while conventional scheduling methods assume that complete information is available at the start. This discrepancy can lead to inefficient resource allocation and delays in recovery. We introduce PyRebuild, a Python-based simulation framework that models the temporal dynamics of damage report arrivals and dynamically allocates reconstruction resources. The framework employs a discrete event simulation approach and tests three scheduling algorithms: Longest Job First (LJF), Shortest Job First (SJF), and random assignment. Using data from the 2018 Lombok earthquakes reconstruction initiative, we evaluate these strategies under both static (complete initial information) and dynamic (gradual information arrival) conditions. Our analysis shows that in five of seven regions dynamic scheduling yields lower Root Mean Square Error (RMSE) between predicted and observed recovery timelines, while in two regions static methods perform slightly better. These results indicate that the optimal scheduling approach depends on local damage patterns, emphasizing the need for adaptive strategies in post-disaster reconstruction planning.

INTRODUCTION

Research Context

Rebuilding after an earthquake is challenging because damage assessments are received gradually over weeks or months rather than all at once. Traditional scheduling methods assume that complete information is available immediately, which may lead to suboptimal resource allocation and delays in recovery. This issue is particularly important since housing reconstruction represents about 50% of total disaster-related losses Comerio (2014).

Resource Allocation Fundamentals

Resource allocation methods in post-disaster reconstruction can be divided into static and dynamic approaches Lawrence and Sewell (1997). Static methods assume that all damage data is available at the outset and assign contractors accordingly, whereas dynamic methods update these assignments as new damage reports arrive. Dynamic scheduling is vital in real disasters, where both damage information and resource availability evolve over time.

Static vs. Dynamic Approaches

Traditional planning tools, such as HAZUS, illustrate the limitations of static methods Federal Emergency Management Agency (2010). While static scheduling is straightforward, it lacks flexibility when new reports emerge. In contrast, dynamic scheduling updates priorities in real time, an approach that has proven effective in domains such as manufacturing and healthcare Pinedo (2012); Green et al. (2004). Recent studies indicate that prioritizing major damage repairs using dynamic methods can lead to improved recovery outcomes Alisjahbana and Kiremidjian (2021); Wang et al. (2023).

Dynamic Scheduling and Demand-Supply Perspectives

Frameworks such as iRe-CoDeS (Suryanto et al., 2022) and Re-CoDeS (Rahman et al., 2018) adopt a demand-supply approach to quantify disaster resilience by modeling recovery as a time-stepping process, where the interaction between resource demand and available supply is assessed at each step. While iRe-CoDeS evaluates community recovery and Re-CoDeS aggregates resilience indicators for civil infrastructure, both frameworks assume that the necessary data are provided in an aggregated form rather than arriving incrementally. In contrast, PyRebuild extends these approaches by simulating the gradual, phased arrival of damage assessments and dynamically allocating reconstruction resources at each time step.

Research Framework

In this paper, we introduce PyRebuild, a simulator that updates scheduling priorities as new damage reports are received. We compare static scheduling (assuming complete data) with dynamic scheduling (updating as data arrives) to assess their impact on reconstruction timeline predictions. Our research framework is designed to simulate recorded recovery trajectories as accurately as possible, thereby providing a benchmark for evaluating scheduling strategies. By calibrating our simulation with real-world data, we can quantify discrepancies between simulated and observed outcomes. These insights lay the groundwork for future enhancements, particularly through the incorporation of reinforcement learning techniques to dynamically adapt scheduling policies in real time.

PROBLEM SETTING

The reconstruction problem after an earthquake requires that contractors be assigned as damage assessments are reported. Unlike static methods—which assume complete data and do not update assignments when new reports arrive Alisjahbana and Kiremidjian (2021)—our approach continuously adjusts the queue based on incoming information (i.e., quantity and severity of incoming houses).

Input Data and Parameters

Let $i \in \mathbb{N}$ index the set of damaged buildings. Each building is represented by the vector (d, r, t) where:

- $d \in \{0, 1, 2\}$ is the damage state (0 = minor, 1 = moderate, 2 = major),
- $r \in \{1, \ldots, R\}$ is the region identifier (with R being the total number of regions),
- t is the time at which the damage assessment is reported.

We model the arrival times of damage assessments using a lognormal distribution. Data from the Lombok earthquakes indicate that most assessments are submitted early, with a few arriving later. We set the parameters $\mu = 100$ days and $\sigma = 0.3$ to capture this skewed behavior. Prior work Alisjahbana and Kiremidjian (2021) supports the use of a lognormal model for damage reporting processes. The observed reporting times are influenced by several independent factors—including the severity of damage, accessibility of the affected site, and administrative delays—that interact in a multiplicative manner, resulting in a right-skewed distribution where most reports occur early and only a few are significantly delayed. For example, Limpert et al. (2001) demonstrated that multiplicative processes typically yield lognormal behavior in various natural and engineered systems. This property makes the lognormal model particularly suitable for time-to-event data, such as damage reporting times, which are inherently positive and often exhibit right-skewed distributions.

Resource Constraints

Each region r has a fixed pool of contractors, C_r, with the following constraints:

- Each contractor works on one building at a time.
- A pre-construction administrative phase must be completed before construction begins, following findings by Alisjahbana and Kiremidjian, 2021.
- Once assigned, a contractor remains engaged until the building is fully repaired.
- The total number of contractors C_r remains constant.

Dynamic Event Processing and Queue Management

For each region r, a priority queue Q_r manages contractor assignments. This queue is updated based on the selected scheduling policy as new damage assessments are processed.

Batch Arrival

In our simulation, we assume:

- The first batch arrives at $t = 0$ (30% of reports),
- The second batch at $t = 60$ days (40% of reports),
- The third batch at $t = 120$ days (30% of reports).

Our simulation assumes that damage assessments are reported in three batches, with a 30–40–30 split (i.e., 30% at $t = 0$, 40% at $t = 60$ days, and 30% at $t = 120$ days). This assumption is guided by operational recommendations such as FEMA's Preliminary Damage Assessment Guide Federal Emergency Management Agency (2024) and studies on rapid damage assessment workflows Robinson et al. (2023). Although actual reporting may vary due to local practices or event severity, this split serves as a useful baseline. Likewise, the chosen lognormal parameters reflect average behavior observed in the Lombok data, though local differences may exist. Future work could refine these parameters for specific regions.

Dynamic Priority Updates and Contractor Allocation

The system recalculates priorities for buildings in Q_r at fixed intervals or when repairs are completed, reordering the queue. When a contractor becomes available, the highest-priority building is selected for repair.

PYREBUILD SIMULATOR

PyRebuild is a discrete event simulation tool that dynamically allocates contractors during post-disaster reconstruction. Built on the SimPy framework SimPyDevelopment-Team (2021), the simulator updates contractor assignments as new damage assessments are reported.

Architecture Overview

The simulator consists of three main components:

- **Region Management:** The `Region` class uses SimPy's `PriorityResource` to manage a fixed pool of contractors (C_r) and track repair completion times.
- **Building Recovery Process:** Each building undergoes a pre-construction administrative delay and an active construction phase, both modeled using lognormal distributions with damage-specific parameters.
- **Policy Implementation:** The simulator implements three scheduling strategies—Longest Job First (LJF), Shortest Job First (SJF), and Random Assignment—and tests them under static and dynamic conditions.

Processing Workflow

Figure 1 shows the overall reconstruction process in PyRebuild. The simulation begins by initializing the environment with fixed parameters. For example, consider a scenario with 100 houses in total and a pool of 20 construction workers available to work in parallel. In the static approach, all 100 damage reports are assumed to be known at the start (day 0), and the simulator applies a scheduling rule—such as Longest Job First (LJF) or Shortest Job First (SJF)—to prioritize which houses receive a contractor first. In contrast, the dynamic approach reveals the 100 houses gradually in batches. In this example, the first batch comprises 30 houses (30% of the total), the second batch 40 houses (40%), and the final batch 30 houses (30%). As soon as a contractor becomes available (with up to 20 houses being worked on simultaneously), the simulator assigns the highest-priority house from the queue. Each house then undergoes a pre-construction phase, experiencing a delay (modeled using a lognormal distribution) that represents administrative processing, followed by an active construction phase—also modeled by a lognormal distribution—which represents the actual repair work. Once a house's repairs are complete, its contractor is released and becomes available for the next house in the queue. The simulation continues until all 100 houses have been processed or a predetermined time limit (e.g., 600 days) is reached. Throughout this process, performance metrics such as the Root Mean Squared Error (RMSE) between simulated and observed recovery trajectories are computed to evaluate how well each scheduling strategy replicates real-world recovery dynamics.

Batch Generation and Arrival Processing

For each region r, damaged buildings are divided into k batches. The first batch arrives at $t = 0$, with subsequent batches arriving according to a lognormal distribution. Each batch is integrated into the priority queue Q_r with unique identifiers and recorded damage levels.

Arrival Pattern

This paper uses Mataram as an example to show batch arrival pattern. Figure 2 shows the assumed arrival pattern for damage assessments in Mataram. In this scenario, 30% of reports arrive immediately, 40% after two months, and the remaining 30% after four months.

Building State Tracking

The simulator tracks each building through three phases: the administrative delay, active construction, and completion. These timestamps, combined with damage levels and regional assignments, enable detailed analysis of recovery progress and contractor utilization.

Performance Metrics

Our simulator computes three indicators to evaluate the accuracy of different scheduling strategies:

1. **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2}, \tag{1}$$

   where $\hat{y}_i$ is the predicted completion ratio on day i, and y_i is the observed completion ratio on day i. This metric quantifies how closely the simulated reconstruction progress matches the actual data, with lower values indicating better alignment.
2. **Absolute Error:**

$$\text{AE}(t) = \left|\hat{y}(t) - y(t)\right|, \tag{2}$$

   which measures the instantaneous deviation of the predicted completion ratio from the actual ratio at each time t.
3. **Completion Ratio Over Time:** The simulator tracks the fraction of houses completed on each day by calculating a cumulative sum of finished repairs. Although this ratio is not a standalone metric, it forms the basis for RMSE and Absolute Error calculations. Visual comparisons of predicted vs. actual completion trajectories also provide a qualitative assessment of how well each scheduling policy replicates the real-world reconstruction pace.

The RMSE provides an overall measure of how well the simulation replicates observed data, while the Absolute Error metrics give a more granular look at deviations at specific points in time. Together, they help identify which scheduling strategies best capture the complex dynamics of post-disaster recovery.

DATA SOURCE

Study Area

Figure 3 presents a map of our study area, which encompasses seven administrative regions of Lombok and Sumbawa in West Nusa Tenggara, Indonesia. The map is generated using OpenStreetMap data via OSMnx and reprojected into the Web Mercator projection (EPSG:3857) to ensure compatibility with standard online mapping services and the CartoDB Positron basemap. Dashed lines indicate the boundaries of each region, while red annotations display their corresponding English names (e.g., Mataram and Sumbawa).

Our analysis uses data from the 2018 Lombok earthquakes. Initial damage assessments were completed by Indonesia's National Disaster Management Authority (BNPB) in September 2018, and daily reconstruction progress was tracked by the Regional Disaster Management Authority of West Nusa Tenggara (BPBD NTB) from October 2018 to March 2020.

Regional Classification

The affected area comprises seven administrative regions:

$$r = \begin{cases} 1 & \text{Mataram} \\ 2 & \text{West Lombok} \\ 3 & \text{North Lombok} \\ 4 & \text{Central Lombok} \\ 5 & \text{East Lombok} \\ 6 & \text{West Sumbawa} \\ 7 & \text{Sumbawa} \end{cases} \tag{3}$$

Damage Distribution and Contractor Availability

Table 1 presents the distribution of damaged houses by region and severity. Damage levels are defined as:

$$d = \begin{cases} 0 & \text{Minor damage (partial structural damage, repairable)} \\ 1 & \text{Moderate damage (significant damage, temporarily uninhabitable)} \\ 2 & \text{Major damage (severe damage, complete reconstruction required)} \end{cases} \tag{4}$$

Table 2 shows contractor availability by region.

Processing Time Parameters

Based on observations from the Lombok reconstruction program Alisjahbana and Kiremidjian (2021), we model construction and pre-construction processing times using lognormal distributions.

Table 1. Distribution of Damaged Houses by Region (r) and Severity Level (d)

Region	Major	Moderate	Minor	Total
Mataram	1,345	3,672	9,500	14,517
West Lombok	14,069	13,556	45,218	72,843
North Lombok	42,049	4,772	8,889	55,710
Central Lombok	4,483	3,096	16,639	24,218
East Lombok	10,104	4,657	12,209	26,970
West Sumbawa	1,283	3,803	13,078	18,164
Sumbawa	1,374	2,756	9,652	13,782
Total	74,707	36,312	115,185	226,204

Table 2. Total Available Construction Contractors by Region (r)

Region	Number of Contractors (C_r)
Mataram	9,917
West Lombok	45,028
North Lombok	22,996
Central Lombok	15,048
East Lombok	15,404
West Sumbawa	10,200
Sumbawa	10,360
Total	128,953

Construction Duration Parameters

Construction times (τ_c) are modeled as:

$$\tau_c \sim \text{LogNormal}(\ln(\mu_c(d)), \beta_c) \tag{5}$$

with:

$$\mu_c(d) = \begin{cases} 30 & \text{days for } d = 0 \text{ (minor)} \\ 40 & \text{days for } d = 1 \text{ (moderate)} \\ 50 & \text{days for } d = 2 \text{ (major)} \end{cases} \quad \text{and} \quad \beta_c = 0.4. \tag{6}$$

Pre-construction Processing Parameters

Administrative delays (τ_p) are modeled as:

$$\tau_p \sim \text{LogNormal}(\ln(\mu_p(d)), \beta_p(d)) \tag{7}$$

with:

$$\mu_p(d) = \begin{cases} 275 & \text{days for } d = 0 \text{ (minor)} \\ 300 & \text{days for } d = 1 \text{ (moderate)} \\ 225 & \text{days for } d = 2 \text{ (major)} \end{cases} \quad \text{and} \quad \beta_p(d) = 0.3. \tag{8}$$

These parameters capture average behavior observed in the Lombok data, though local variations may occur.

Table 3. RMSE by Region and Scheduling Strategy. Lower RMSE values indicate closer to recorded values. Multiple bold values indicate that no single strategy consistently outforms than the other one.

Region	Static LJF	Static SJF	Dynamic LJF	Dynamic SJF
Mataram	0.1667	0.1577	**0.1125**	**0.1101**
West Lombok	**0.0772**	**0.0719**	0.0931	0.0960
North Lombok	0.1572	0.1247	**0.1256**	**0.1140**
Central Lombok	0.1360	0.1285	**0.1104**	**0.1083**
East Lombok	0.1227	0.1008	**0.0851**	**0.0797**
West Sumbawa	0.1078	0.1045	**0.1006**	**0.1002**
Sumbawa	**0.1165**	**0.1247**	0.1943	0.1957

RESULTS

The results show that dynamic scheduling generally produces predictions closer to the actual recovery timelines than static scheduling. In five of the seven regions (Mataram, North Lombok, Central Lombok, East Lombok, and West Sumbawa), dynamic methods yield lower RMSE. In two regions (West Lombok and Sumbawa), static scheduling slightly outperforms dynamic methods.

Table 4. Regional Resource and Damage Profile. Bolded values highlight regions where static methods perform better. These regions have higher resource ratios ($\rho > 0.5$) and lower major damage percentages ($\delta < 30\%$), conditions that may favor static scheduling.

Region	Resource Ratio (ρ)	Major Damage (δ)
Mataram	0.68	9.3% (1,345)
West Lombok	**0.62**	**19.3% (14,069)**
North Lombok	0.41	75.5% (42,049)
Central Lombok	0.45	18.5% (4,483)
East Lombok	0.51	37.5% (10,104)
West Sumbawa	0.55	7.1% (1,283)
Sumbawa	**0.48**	**10.0% (1,374)**

DISCUSSION

Our results across the seven regions indicate that while dynamic scheduling generally aligns better with observed recovery data, certain phases or local conditions can favor a static approach. To illustrate these points, we highlight the example of Mataram, shown in Figures 4 and 5, noting that other regions exhibit similar patterns but may differ in timing or magnitude.

Regional Profiles and Their Impact on Scheduling Performance

Table 4 summarizes the regional resource ratios (ρ) and the percentage of major damage (δ) for the study area. Notably, regions such as West Lombok and Sumbawa exhibit higher resource ratios (i.e., $\rho > 0.5$) alongside relatively lower percentages of major damage (i.e., $\delta < 30\%$). These characteristics indicate that in these regions, the available contractor resources are relatively abundant compared to the number of severely damaged buildings. As a result, a static scheduling approach—which assigns contractors based solely on initial data—may perform adequately because the repair tasks are less complex and the resource availability is sufficient to handle the workload.

In contrast, regions with lower resource ratios and higher percentages of major damage require a more flexible, dynamic scheduling strategy to accommodate the evolving repair needs. This variability across regions underscores that no single scheduling policy is universally optimal. The observed differences motivate the potential development of adaptive strategies, such as reinforcement learning methods, which can dynamically adjust scheduling policies in response to real-time changes in resource availability and damage severity.

Illustrative Example: Mataram Region

Figure 4 compares the predicted reconstruction trajectories from static and dynamic approaches with the observed recovery data in Mataram. In the early and mid stages (up to about day 350), the static method occasionally tracks the real data more closely. However, the dynamic approach matches observed progress better at later stages. After day 500, both approaches underestimate the final acceleration in reconstruction, suggesting that real-world factors (e.g., additional resources or policy changes) occurred late in the process.

Figure 5 provides the corresponding absolute error analysis, highlighting when each scheduling strategy deviates most from the observed data. Around days 350–400, both static and dynamic methods exhibit a spike in error. While the dynamic approach maintains a lower RMSE overall, this peak reveals that no single scheduling policy is optimal under every condition. Similar mid-stage error spikes appear in other regions as well, reinforcing the broader conclusion that a flexible or mixed scheduling policy could be more robust.

Implications for Other Regions

Although these figures focus on Mataram, we observed similar tendencies in regions like North Lombok and East Lombok, albeit with varying timelines and severity levels. Specifically:

- **Late-stage underestimation:** In several regions, both static and dynamic methods fell behind actual recovery speed after a certain point, indicating that unmodeled factors (e.g., sudden resource infusions) accelerated rebuilding.
- **Mid-stage spikes in error:** Most regions showed a period where both approaches deviated significantly from real data. The timing of this spike differed by region but consistently revealed that no single policy performed best at all times.

Need for Adaptive Strategies

These observations suggest that incorporating real-time adaptive scheduling could address the limitations of a single fixed policy. Specifically, a reinforcement learning

(RL) approach, such as Deep Double Q-Networks (DDQN), could switch between scheduling rules (e.g., Longest Job First and Shortest Job First) as local conditions change. Our findings motivate such an approach, indicating that flexible policies are likely to capture late-stage surges in recovery more effectively and reduce mid-stage error spikes.

FUTURE WORK

Dynamic Contractor Management

Future research should focus on developing methods to manage fluctuations in contractor availability during reconstruction. This may include systems that account for gradual workforce changes or sudden contractor dropouts, as well as inter-regional sharing of contractors.

Real-time Strategy Adaptation

Another promising direction is the application of reinforcement learning techniques, such as Deep Double Q-Networks (DDQN), to develop adaptive scheduling systems. These systems could switch between scheduling strategies in real time, potentially leading to hybrid approaches that outperform fixed strategies.

REFERENCES

Alisjahbana, I. and Kiremidjian, A. (2021). "Modeling housing recovery after the 2018 lombok earthquakes using a stochastic queuing model." *Earthquake Spectra*, 37(2), 587–611.

Comerio, M. (2014). "Disaster recovery and community renewal: Housing approaches." *Cityscape*, 16(2), 51–68.

Federal Emergency Management Agency (2010). "Hazus-mh mr5 technical manual." *Report no.*, Department of Homeland Security, Washington, DC.

Federal Emergency Management Agency (2024). "2024 operational draft preliminary damage assessment guide. Accessed: March 15, 2025.

Green, L., Soares, J., Giglio, J., and Green, R. (2004). "Using queueing theory to increase emergency department provider staffing." *Academic Emergency Medicine*, 11(5), 162–168.

Lawrence, S. R. and Sewell, E. C. (1997). "Heuristic, optimal, static, and dynamic schedules when processing times are uncertain." *Journal of Operations Management*, 15(1), 71–82.

Limpert, E., Stahel, W., and Abbt, M. (2001). "Log-normal distributions across the sciences: keys and clues." *BioScience*, 51(5), 341–352.

Pinedo, M. (2012). "Scheduling: Theory, algorithms, and systems." *Operations Research*, 60(2), 253–274.

Rahman, A., Nugroho, S., and Setiawan, H. (2018). “A compositional demand/supply framework to quantify the resilience of civil infrastructure systems (re-codes).” *Proceedings of the International Conference on Infrastructure Resilience*, 150–160.

Robinson, C., Nsutezo, S. F., Ortiz, A., Sederholm, T., Dodhia, R., Birge, C., Richards, K., Pitcher, K., Duarte, P., and Lavista Ferres, J. M. (2023). “Rapid building damage assessment workflow: An implementation for the 2023 rolling fork, mississippi tornado event. Preprint; accessed March 2025.

SimPyDevelopmentTeam (2021). *SimPy: Discrete-Event Simulation for Python*. Accessed: 2024-11-29.

Suryanto, R., Pratama, D., and Wulandari, E. (2022). “A demand-supply framework for evaluating the effect of resource and service constraints on community disaster resilience (ire-codes).” *Proceedings of the International Conference on Disaster Management and Resilience*, 75–85.

Wang, H., Peng, T., Nassehi, A., and Tang, R. (2023). “A data-driven simulation-optimization framework for generating priority dispatching rules in dynamic job shop scheduling with uncertainties.” *Journal of Manufacturing Systems*, 70, 288–308.

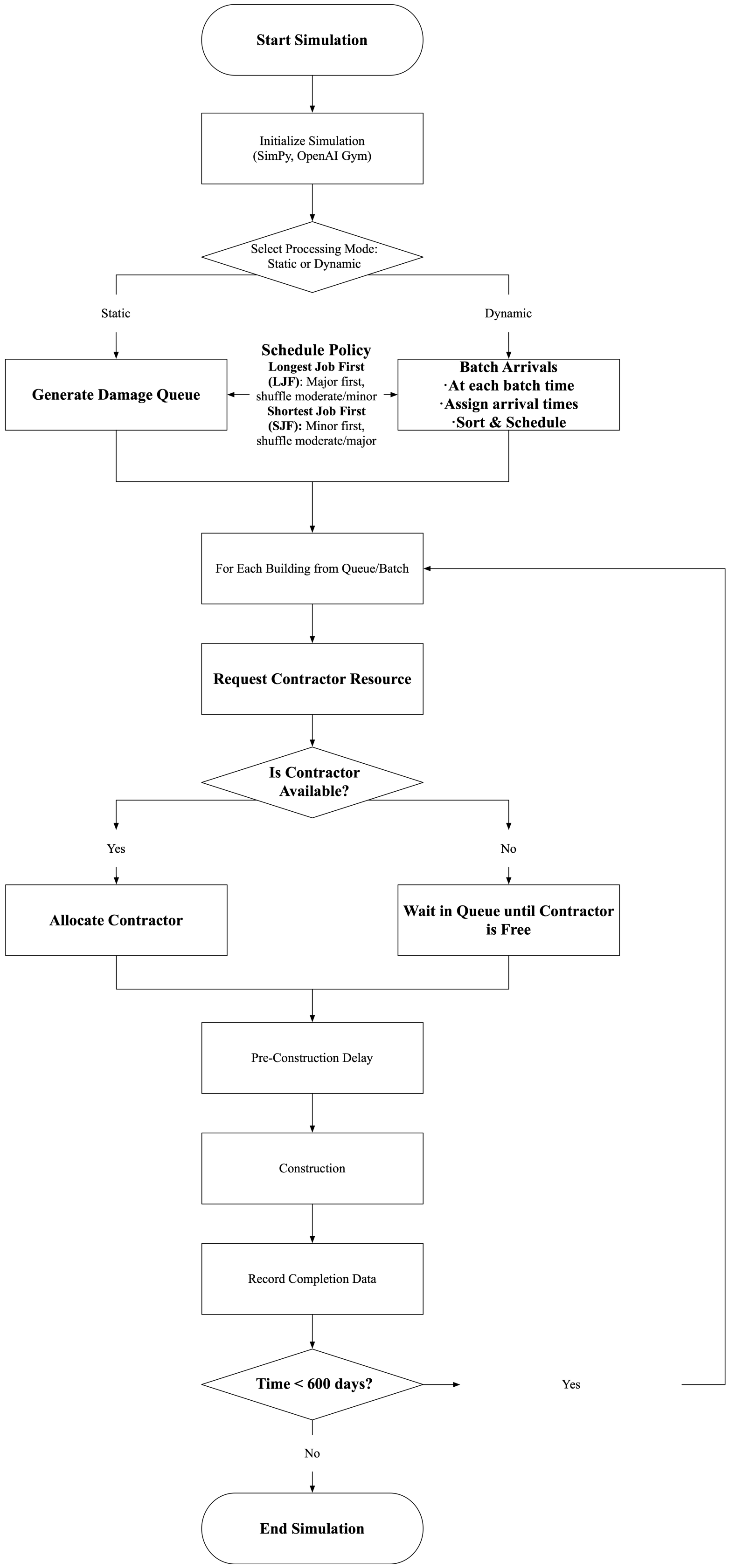

Fig. 1. Reconstruction Process. The diagram illustrates PyRebuild's process from initial damage assessment through completion, including dynamic resource allocation and ongoing monitoring.

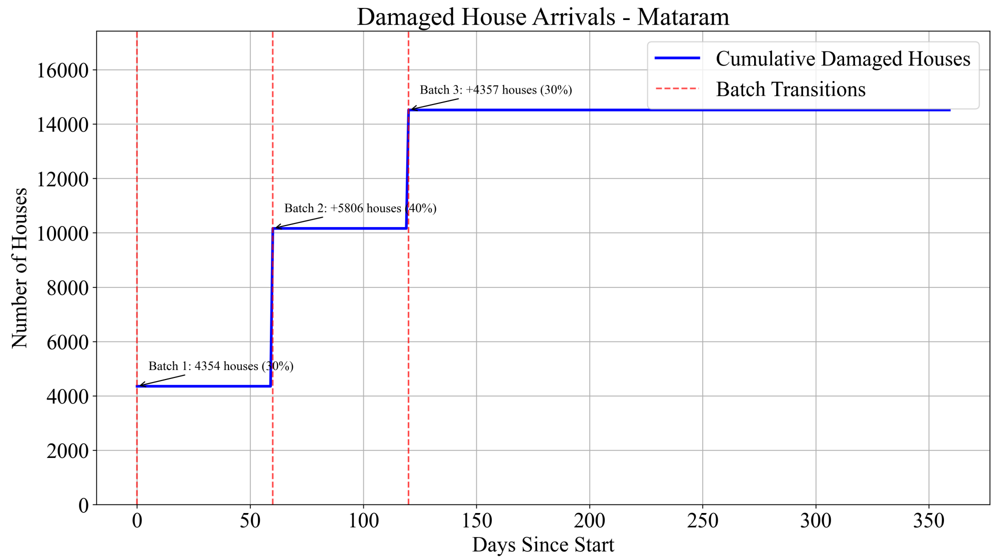

Fig. 2. Damaged Houses Arrival in Mataram: The blue line represents the cumulative damage reports, with red dashed lines indicating the batch arrival times.

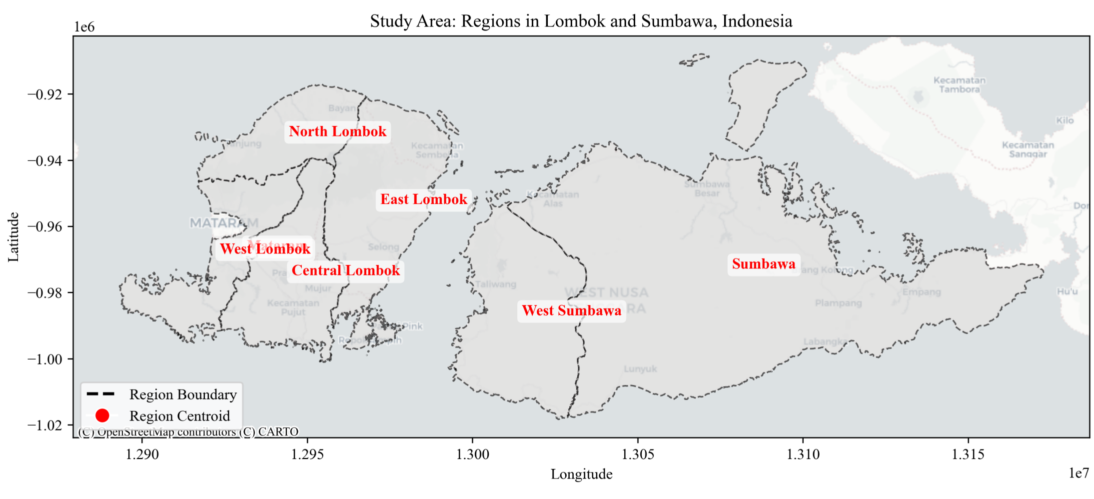

Fig. 3. Map of the study area depicting the administrative regions in Lombok and Sumbawa, Indonesia. The dashed boundaries represent the delineated regions, and the red labels indicate the English names of the regions as determined from local administrative queries. The grayscale basemap (CartoDB Positron) provides geographic context.

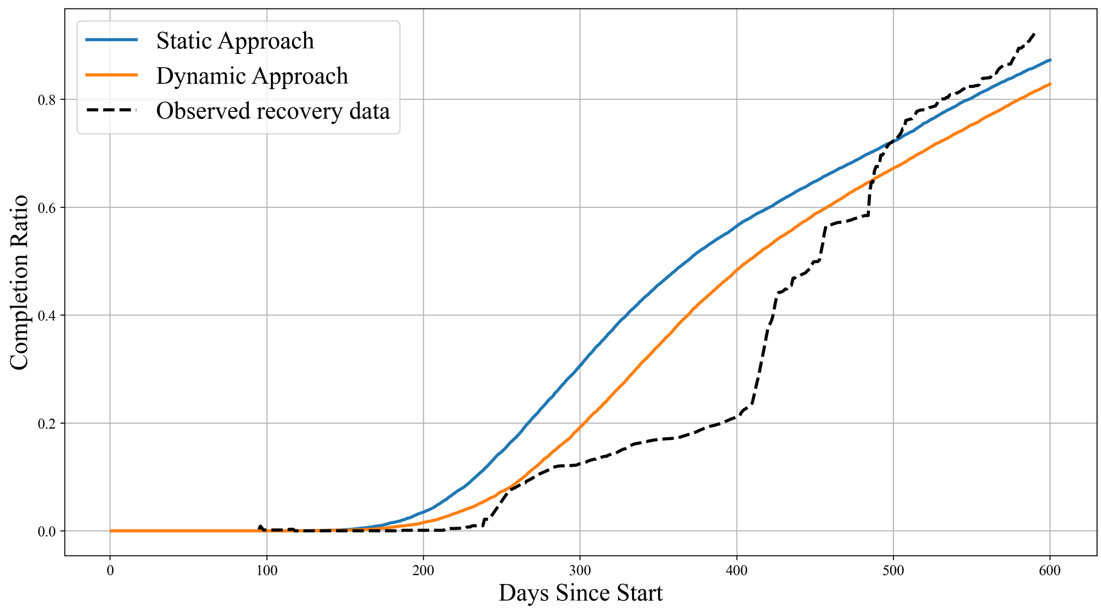

Fig. 4. Recovery trajectory in Mataram as an example region. Although the static method sometimes aligns with observed data early on, the dynamic approach shows closer agreement overall. Both methods lag behind actual recovery after day 500, indicating potential benefits of adaptive scheduling.

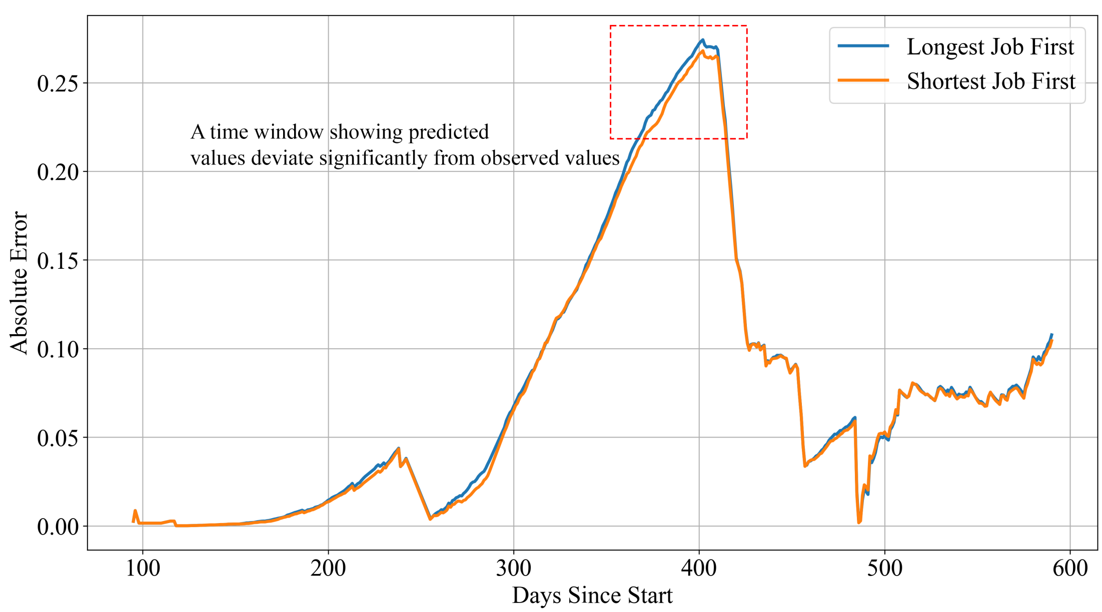

Fig. 5. Absolute error analysis in Mataram. The highlighted period (days 350–400) shows a peak in error for both scheduling methods, underscoring the need for adaptive policies that can respond to changing recovery dynamics.