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

Qianchen Yu.,[1] Irene Alisjahbana,[2] and Vivian W.H. 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, M.E. Rinker, Sr. School of Construction Management, 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 tests three scheduling algorithms (Longest Job First, Shortest Job First, and random assignment) under static and dynamic conditions. Using 2018 Lombok earthquake data, we find dynamic scheduling yields lower Root Mean Square Error between predicted and observed recovery timelines in five of seven regions, while static methods perform marginally better in two regions. Results indicate optimal scheduling depends on local damage patterns, highlighting the need for adaptive post-disaster reconstruction strategies.

INTRODUCTION
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 like Re-CoDeS and iRe-CoDeS have applied a demand-supply approach to balance repair needs with available resources Rahman et al. (2018); Suryanto et al. (2022). In PyRebuild, we simulate the gradual arrival of damage assessments and the dynamic allocation of contractors, building on these demand-supply concepts to support adaptive scheduling strategies. This review of existing literature underscores the potential of adaptive approaches and motivates our comparison of static and dynamic methods as a step toward more sophisticated, real-time policy adaptation.

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.

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—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),
- $r \in \{1,...,R\}$ 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 damage arrivals using a lognormal distribution. We use parameters $\mu = 100$ days and $\sigma = 0.3$ to capture this skewed behavior, which is supported by previous works. Damage reporting times are shaped by factors like severity, accessibility, and administrative delays, which together create a right-skewed pattern well-represented by a lognormal distribution—where most reports arrive early, and a few are substantially delayed.

Resource Constraints. Each region $\boldsymbol{r}$ has a fixed pool of contractors C_r, Each contractor works on one building at a time, and a pre-construction administrative phase must be completed before construction begins. Once assigned, a contractor remains engaged until the building is fully repaired. The total number of contractors C_r remains constant throughout the process.

Dynamic Event Processing and Queue Management. For each region r, a priority queue Qr manages contractor assignments. This queue is updated based on the selected scheduling policy as new damage assessments are processed.
Batch Arrival. 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 and studies on rapid damage assessment workflows. 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 Qr 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, 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 (Cr) 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

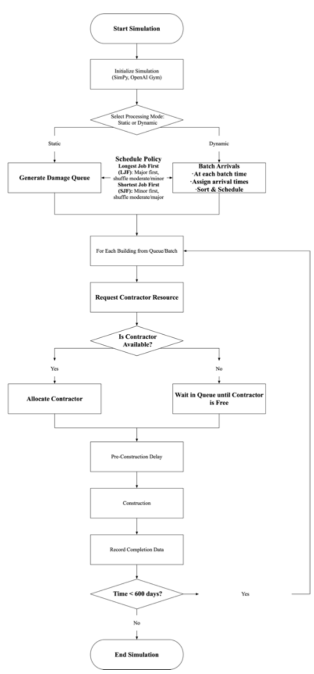

Figure 1 Flowchart of PyRebuild. The diagram illustrates PyRebuild's process from initial damage assessment through completion, including dynamic resource allocation and ongoing monitoring.

Figure 1 illustrates the workflow in PyRebuild. The simulation begins by initializing key parameters—for instance, 100 damaged houses and a pool of 20 construction workers.

In the **static approach**, all damage reports are assumed available at day 0. The simulator assigns contractors based on a predefined rule, such as Longest Job First (LJF) or Shortest Job First (SJF), and work proceeds accordingly.

In the **dynamic approach**, damage reports arrive in batches: 30% on day 0, 40% at day 60, and 30% at day 120. As new reports arrive and contractors become available, the system reprioritizes and assigns the next house based on the current scheduling policy.

Each house goes through a **pre-construction delay** (administrative processing) and an **active construction phase**, both modeled using lognormal distributions. Once completed, the contractor is released to take on a new assignment.

The simulation runs until all houses are completed or a time cap (e.g., 600 days) is reached. Throughout, performance is evaluated by comparing simulated recovery trajectories to observed data using metrics like RMSE.

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 Qr with unique identifiers and recorded damage levels.

Arrival Pattern. This session 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.

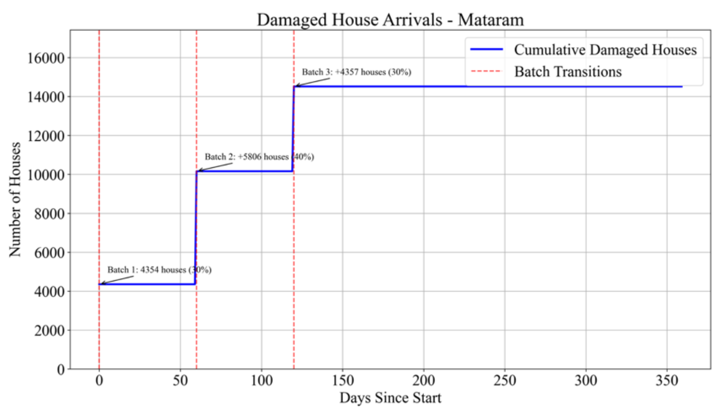

Figure 2 Damaged Houses Arrival in Mataram: The blue line represents the cumulative damage reports, with red dashed lines indicating the batch arrival times.

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:

Root Mean Squared Error (RMSE):

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

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.

Absolute Error:

$$AE(t) = |\hat{y}_t - y_t| \tag{2}$$

which measures the instantaneous deviation of the predicted completion ratio from the actual ratio at each time t.

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 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.

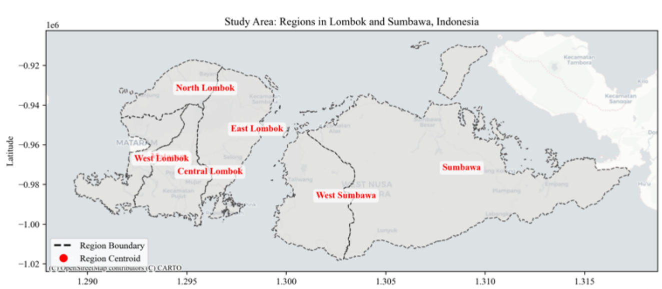

Figure 3: Map of the study area depicting the administrative regions in Lombok and Sumbawa, Indonesia. The dashed boundaries represent the regions, and the red labels indicate the English names of the regions. The grayscale basemap (CartoDB Positron) provides geographic context.

Regional Classification. The affected area comprises seven administrative regions:

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

Damage Distribution and Contractor Availability. Table 1 presents the distribution of damaged houses by region and severity. Table 2 shows contractor availability by region.

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 (Cr)
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**

Housing damage levels are defined as:

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

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.

Construction Duration Parameters. Construction times (τ_c) are modeled as:

$$\tau_c \sim LogNormal(\ln(\mu_c(d)), \beta_c) \quad (5)$$

with:

$$\mu_c(d) = \begin{Bmatrix} 30\ days\ for\ d = 0\ (minor) \\ 40\ \ days\ for\ d = 1\ (moderate)\ \ and\ \beta_C =\ 0.4. \\ 50\ \ days\ for\ d = 2\ (major) \end{Bmatrix} \tag{6}$$

Pre-construction Processing Parameters. Administrative delays (τ_p) are modeled as:

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

with:

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

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

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 3: RMSE by Region and Scheduling Strategy. Lower RMSE values indicate closer to recorded values. Multiple bold values indicate that no single strategy consistently outperforms 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

DISCUSSION

Our results across the seven regions indicate that while dynamic scheduling generally aligns better with observed recovery data, no single strategy consistently outperforms across all contexts. In certain regions or phases of recovery, static approaches may perform equally well or even better than dynamic methods. To illustrate these dynamics, we highlight the example of Mataram in Figures 4 and 5, while noting that similar patterns appear elsewhere, though with different timing and magnitude.

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)

Region	Resource Ratio (ρ)	Major Damage (δ)
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)**

Table 4 summarizes the regional resource ratios (ρ) and major damage percentages (δ) across the study area. While West Lombok and Sumbawa are the only regions where static scheduling outperformed dynamic methods, this outcome should not be overinterpreted. These regions did not have the highest contractor availability or the lowest damage levels. Instead, their relatively balanced conditions—moderate demand paired with adequate resources—meant that static scheduling was sufficient to maintain a stable reconstruction pace.

In contrast, regions like North Lombok, which experienced widespread severe damage and more limited resources, benefited significantly from dynamic scheduling. The flexibility to reprioritize tasks as new damage reports arrived helped improve alignment with observed recovery patterns.

These findings reveal that strategy effectiveness cannot be determined solely by resource ratio or damage severity. Instead, outcomes depend on the interaction of multiple factors—such as the timing of report arrivals, variability in task durations, and how resources are allocated over time. Relying only on static indicators oversimplifies what is, in practice, a highly dynamic process.

Illustrative Example: Mataram Region

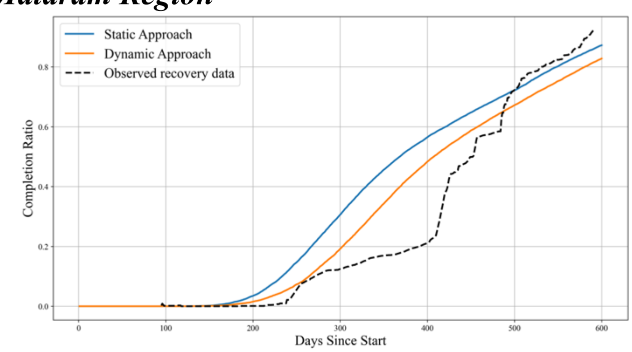

Figure 4: Recovery trajectory in Mataram as an example region using Longest Job First (LJF) strategy. Although the static method sometimes aligns with observed data early on, the dynamic approach shows closer agreement overall. Both methods lag behind actual recovery after day 500, indicating potential benefits of adaptive scheduling.

Figure 4 shows the recovery trajectories in Mataram under static and dynamic scheduling compared to observed data. While the static method aligns better early on (before day 350), the dynamic approach better captures the later recovery trend. Both underestimate recovery after day 500, likely due to unmodeled external factors such as policy shifts or sudden resource infusions—highlighting the limitations of fixed strategies.

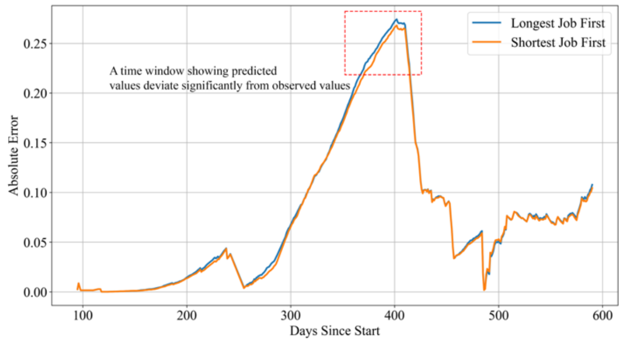

Figure 5: Absolute error analysis in Mataram. The highlighted period (days 350-400) shows a peak in error for both scheduling methods, underscoring the need for adaptive policies that can respond to changing recovery dynamics.

Figure 5 presents the absolute error over time. A clear spike between days 350–400 marks a period where both methods deviate significantly from observed outcomes. Although dynamic scheduling yields a lower RMSE overall, this error window reinforces that **no single strategy performs best at all times**.

These patterns—**late-stage underestimation** and **mid-stage error spikes**—also appear in other regions, including North and East Lombok, with varying timing and magnitude.

Need for Adaptive Strategies. These findings emphasize that fixed scheduling rules—whether static or dynamic—are often insufficient. To respond effectively to shifting conditions, **adaptive strategies are needed**. Reinforcement learning approaches, such as **Deep Double Q-Networks (DDQN)**, offer a promising path forward by allowing real-time adjustment between strategies like LJF and SJF based on current recovery dynamics.

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.S. (2021). "Modeling housing recovery after the 2018 Lombok earthquakes using a stochastic queuing model." *Earthquake Spectra*, 37(2), 587-611.

Comerio, M.C. (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." Retrieved from https://www.fema.gov/disaster/how-declared/preliminary-damage-assessments/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.A., 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. doi: 10.1109/ICIR.2018.00025

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." arXiv:2306.12589. Preprint; accessed March 2025.

SimPyDevelopmentTeam (2021). "SimPy: Discrete-Event Simulation for Python." Retrieved from https://simpy.readthedocs.io/. 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. doi: 10.1109/ICDMR.2022.00105

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. doi: 10.1016/j.jmsy.2023.08.001