Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

AN EVOLUTIONARY METHOD TO REFINE IMPERFECT SENSOR DATA FOR CONSTRUCTION SIMULATION

Prabhat Shrestha
Amir H. Behzadan

Department of Construction Science
Texas A&M University
College Station, TX 77843, USA

ABSTRACT

Construction simulation is used to analyze uncertainties inherent to project activities and variations in work packages. However, existing simulation systems often fail to meaningfully contribute to the decision-making process due to their inability to evolve with changing project conditions. Equipping simulation models with sensing and reality capture technologies has been investigated as possible remedies to this problem. This, however, requires meticulous effort to procure, set up, operate, synchronize, and calibrate peripheral devices for data collection, transmission, and mining. Furthermore, sensor readings are often noisy and imperfect. The chaos theory explains how small variations in sensor readings used as simulation model input can lead to relatively large volatility in the output even in simple linear systems. This paper investigates a scientific methodology for generating more stable simulation models using an evolutionary algorithm that produces clean datasets by processing and significantly reducing noise in imperfect data obtained from consumer-grade sensors.

1 INTRODUCTION

1.1 Value of Simulation to Project Planning

In planning project activities, the exact sequence of tasks (or events) that shape an activity may not be predetermined, and tasks may even be performed out of order or interchanged depending on resource availability and other constraints. This intrinsic fuzziness can further complicate the formulation of project attributes (e.g. cost, schedule, resource consumption) using mathematical terms. One potential solution to this problem is the use of discrete event simulation (DES) that is best suited to represent uncertainties in dynamic systems (AbouRizk et al. 2011; Martinez and Ioannou 1997). Despite recent advancements in simulation science, simulation models have not been yet fully accredited as an integral part of the decision-making process in construction projects due to barriers such as the intense initial effort required to set up and compile models (Oloufa, Ikeda, and Nguyen 1998), lack of flexibility (a.k.a. rigidity) of the simulation model, user incompetence, and specificity of the simulation environment (Hajjar and AbouRizk 2002). Recent studies have also discussed that most simulation systems in the construction domain are not capable of effectively processing construction-phase project data (Leite et al. 2016), thus rendering the model obsolete for decision-making shortly after the project starts. To this end, integrating field data in simulation modeling has been explored in recent years. Such studies, however, have been mostly carried out in fields outside architecture, engineering, construction, and facility management (AEC/FM). Akhavian and Behzadan (2013) identified some of the efforts in dynamic data-driven application simulation (DDDAS) as used in railway engineering simulation (Huang and Verbraeck 2009), and supply change modeling in aerospace engineering (Tannock et al. 2007). The work also described limited efforts in data-driven construction simulation that were to a large extent focused on

equipment location data (Song and Eldin 2012). Further investigation shows some recent applications of DDDAS including sporadic studies in construction where for instance, Akhavian and Behzadan (2015) created data-driven models for equipment activity recognition, and Vasenev et al. (2014) proposed a data collection framework for decision-making.

As previously discussed, barriers including the inability of existing simulation modeling techniques to integrate sensor data into the simulation, as well as the unregulated environment of sensing systems in AEC/FM still hinder the prospect of full adoption of high level data integration protocols in simulation modeling. The following Subsection explains the issue of inherent fuzziness in sensor data and potential problems it may cause for successful integration of data in simulation modeling.

1.2 Inherent Fuzziness of Sensor Data

In recent years, researchers have explored the potentials of advanced data sensing and computing technologies, and information modeling, to improve project planning and delivery in construction, and to establish new industry standards and paradigm shifts for decision-making throughout the life cycle of AEC/FM projects (Golparvar-Fard, Peña-Mora, and Savarese 2011; Leite et al. 2016). This proliferation of the use of sensor data in project planning, implementation, monitoring, and control (Spencer Jr, Ruiz-Sandoval, and Kurata 2004) can be found in almost every project type as the technology has become ubiquitous and more affordable. For instance, work by Chae et al. (2012) in structural health monitoring, Razavi and Hass (2010) in on-site material tracking, and Choe et al. (2014) in site safety have all demonstrated the versatile applications of these new technologies.

However, the abundance of data does not necessarily translate into effective data utilization. Barriers such as bias in data interpretation, high upfront costs (associated with procurement, installation, and maintenance), and inherent uncertainty in data compounded by the dynamic nature of construction projects can lead to technology gaps which reduce the reliability of results. Zamalloa and Krishnamachari (2007) identified several factors that cause variation and uncertainty in sensor reliability, and undermine the adoption of new technological advances in the construction industry, since handling, cleaning, and post-processing of raw sensor data requires special training and skillset that is otherwise not expected from a trained construction engineer or project manager (Lee et al. 2013).

1.3 Chaos Theory and Imperfect Sensor Data

Most of the data collected by sensors is not crisp and well differentiated; they present an image of the real world with uncertain progressions and states (Izadi et al. 2015). The resulting volatility from imperfect sensor data can be described using chaos theory, a branch of mathematics that deals with systems that appear to be deterministic (e.g. a construction schedule) but can experience chaotic events. Chaos theory states that despite its deterministic nature, a dynamic system can behave in an unpredictable (i.e. chaotic) manner with changes in initial conditions. Lorenz (1963) expressed this as “the present determines the future, but the approximate present does not approximately determine the future”. If uncertain data from a sensor network is fed into a model describing a dynamic construction system, chaos theory implies that model performance can randomly alter with small changes in the accuracy of sensor data.

The scenario in Figure 1 is used to show show the degree of volatility of a nondeterministic network to variations in the input data (Kiel and Elliott 1996). In this Figure, double arrows imply that the resource on a link can travel either way. Assuming that each node costs one resource unit, the total operation cost of moving 100 objects from node 1 to node 4 is calculated by adding the costs of individual activities (i.e. nodes). Initially (iteration 0), all outgoing links are assigned equal strength values, implying that each link is equally likely to be picked by a resource leaving a node. In general, these strength values determine the probability of any outgoing link on which a resource flows between activities. This operation is repeated for five more iterations (numbered 1 through 5). In each iteration, a random subset of links are selected and their strength values (model input) altered by 10%. It can be verified that even a slight alteration in the input creates a large volatility in the total operation cost (model output). For

instance, in iteration 3 the strength value of only one link is altered by 10% compared to benchmark (iteration 0). This small change, however, results in a 6% decrease in the output.

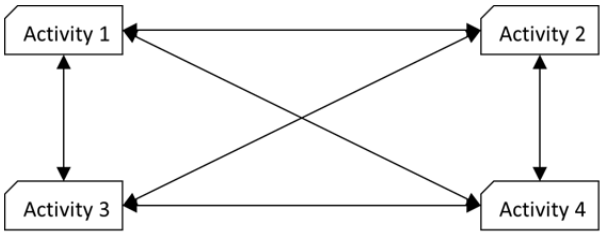

Figure 1: Sample nondeterministic network.

2 RESEARCH OBJECTIVE AND CONTRIBUTIONS

The work presented in this paper aims at designing a scientific methodology, inspired by chaos theory and built upon an evolutionary algorithm, capable of refining imperfect (noisy) sensor data and generating clean datasets that can be used for simulation input modeling. Practically, results of this work are sought to allow the use of low-cost sensors for data collection while minimizing the impact of inaccuracies on the overall quality of the simulation model. Ultimately, this approach promotes simulation-based decision-making by reducing the cost of data acquisition.

3 METHODOLOGY

In this Section, designed methodology of refining imperfect sensor data for simulation input modeling is explained. As seen in the activity cycle diagram (ACD) of Figure 2, the data collection experiment modeled a warehouse operation in which boxes were first transported from a loading area to an inspection area. The content of each box was inspected and if approved, the box was further moved to the unloading area. Otherwise, the box was removed from the system.

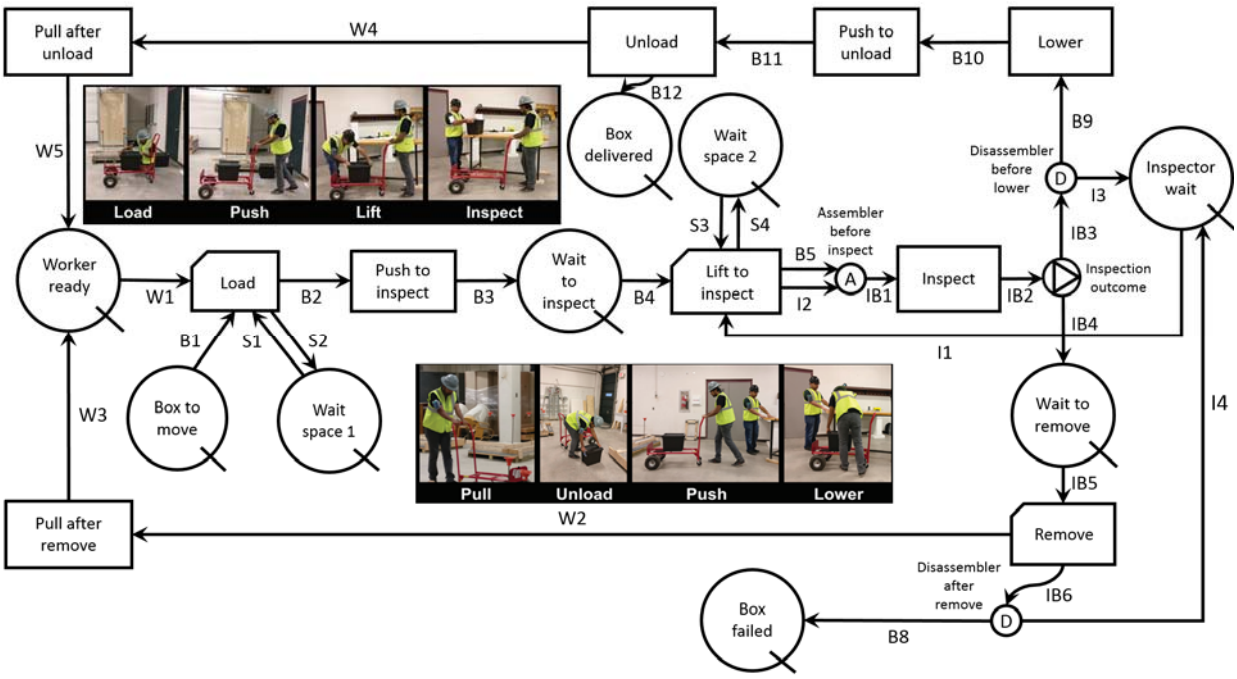

Figure 2: ACD diagram of the cyclic warehouse operation.

The cyclic operation starts with a worker loading a box onto a cart and then pushing it to the inspection area. Next, an inspector lifts the box and inspects it, while, the worker waits in the inspection area. At the conclusion of inspection, the inspector either accepts the box or rejects it. Upon acceptance, the worker lowers the box onto the cart, pushes it to the unloading area, unloads the box and then pulls the empty cart back to the loading area. If the box is rejected, however, the inspector removes the box from the system, and the worker pulls back to the loading area with an empty cart. In both cases, the worker moves back to the loading area and the cycle starts over. This operation was performed for a total of 30 cycles. Two smartphones were mounted on each performer's body (one on upper arm and another on waist). Data was collected from the built-in accelerometer, linear accelerometer, and gyroscope sensors of each smartphone. Next, a host of machine learning algorithms were used to recognize different activities and their durations as performed by each worker and inspector. Further details of the designed human activity recognition (HAR) process can be found in another publication by the authors (Nath and Behzadan 2017).

The output of the HAR step was then used to generate an activity transition matrix, called the dependency network assimilator (DNA). The DNA matrix corresponding to the sequence of activities in the real system as captured by the sensor data is denoted $DNA_{extracted}$. Using prior knowledge of the activity progression sequence, an ideal transition matrix, DNA_{clean}, was also created. As expected, $DNA_{extracted}$ and DNA_{clean} are not necessarily identical as the sensor data and the HAR algorithms used to create $DNA_{extracted}$ are prone to inaccuracy. As a result, the choice of the DNA matrix used to create the underlying ACD of a simulation model corresponding to the warehouse operation, will influence the quality of the simulation output. In theory, the model produces the most realistic results with DNA_{clean}. However, obtaining this DNA matrix is only possible under perfect conditions (i.e. 100% accuracy of sensor data and HAR algorithms). Therefore, the challenge is to use available information (i.e. extracted DNA matrix with intrinsic fuzziness) and still create a simulation model that can closely mimic the real system and predict its performance with high fidelity. Figure 3 shows the main building blocks of the designed methodology built upon a genetic algorithm (GA) to achieve this goal.

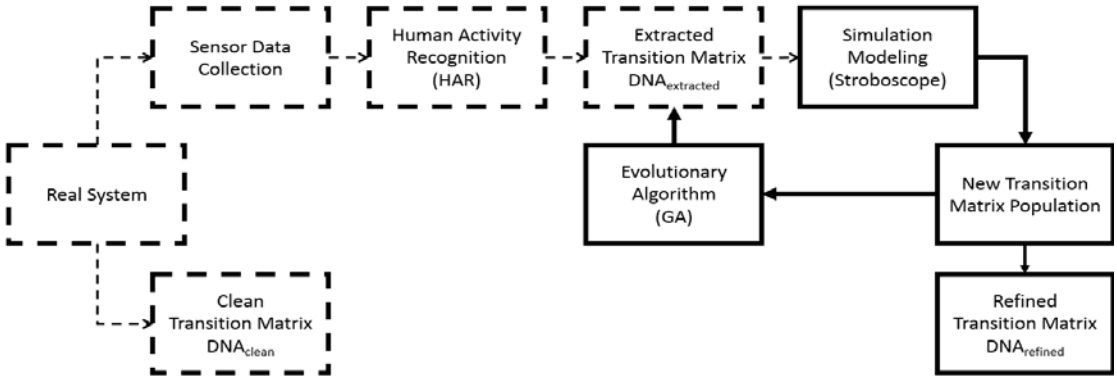

Figure 3: Block diagram of the designed sensor data refinement methodology.

In a nutshell, the $DNA_{extracted}$ matrix generated from sensor data and HAR algorithms is used to create a probabilistic ACD of the operation and the corresponding DES model in Stroboscope (Martinez 1996). Several iterations of the model are run, and given the probabilistic (uncertain) nature of the ACD, each iteration results in a new (and slightly different) DNA matrix (a.k.a. daughter population in GA terms). This new pool of DNA matrices undergoes fitness evaluation and new mother matrices are generated. New DNA matrices are then fed into the DES model, and the process is repeated until a satisfactory level of fitness is achieved. The combination of simulation and GA enables the production of a refined DNA matrix from the fuzzy sensor data. The following Subsections provide detailed discussions about this process.

4 THE SIMULATION MODEL

The warehouse operation experiment consists of independent activities, each with discrete start and end times. These activities can be defined as separate nodes in a DES network connected by links carrying resources. These resources (i.e. worker, inspector, boxes) are defined and stored in queues.

4.1 The Deterministic (Ideal) DES Model

The ACD shown in Figure 2 illustrates a deterministic DES model of the warehouse operation experiment. The model's accuracy in terms of logic and activity durations was validated through a point-by-point comparison with the video recordings of the real experiment at random times. This validation also ensures deterministic (non-probabilistic) transitions between successive activities thus yielding a clean DNA matrix. As previously described, each element in the DNA matrix represents the strength (i.e. likelihood) of transitioning from a preceding node to a succeeding node. It should be noted that while most activities shown in Figure 2 are both a predecessor and a successor, some are only of one type. For instance, activity 'load' is only a preceding activity as it starts a cycle whereas activities 'unload' and 'remove' are only succeeding activities as they end a cycle. Thus, the DNA matrix does not contain an equal number of preceding and succeeding activities, and consequently may not always be a square matrix. The DNA_{clean} matrix, as shown in Figure 4(a) shows ideal transitions between activities. Rows (i = 1…n) represent preceding activities and columns (j = 1…m) represent succeeding activities. As expected, most rows are binary (holding only one non-zero value) since each activity is only followed by one succeeding activity. The only exception to this rule is activity 'inspect', which can be followed by either activity 'lower' or activity 'reject', depending on the inspection result.

	Load	Push to inspect	Lift to inspect	Inspect	Lower	Remove	Push to unload	Unload	Pull after unload	Pull after remove	Box Failed	Box delivered
Load	0	30	0	0	0	0	0	0	0	0	0	0
Push to inspect	0	0	30	0	0	0	0	0	0	0	0	0
Lift to inspect	0	0	0	30	0	0	0	0	0	0	0	0
Inspect	0	0	0	0	10	20	0	0	0	0	0	0
Lower	0	0	0	0	0	0	30	0	0	0	0	0
Remove	0	0	0	0	0	0	0	0	0	0	10	0
Push to unload	0	0	0	0	0	0	0	30	0	0	0	0
Unload	0	0	0	0	0	0	0	0	0	0	0	20
Pull after unload	0	0	0	0	0	0	0	0	0	0	0	0
Pull after remove	0	0	0	0	0	0	0	0	0	0	0	0
Ready to load	30	0	0	0	0	0	0	0	0	0	0	0

(a)

	Load	Push to inspect	Lift to inspect	Inspect	Lower	Remove	Push to unload	Unload	Pull after unload	Pull after remove	Box Failed	Box delivered
Load	0	26	2	0	0	0	0	2	0	0	0	0
Push to inspect	2	0	23	0	0	0	0	0	2	2	0	0
Lift to inspect	0	0	0	25	0	0	0	3	2	0	0	0
Inspect	0	0	0	0	20	10	0	0	0	0	0	0
Lower	0	0	2	1	0	0	15	1	1	0	0	0
Remove	0	0	0	0	0	0	0	0	0	0	10	0
Push to unload	0	0	0	0	0	0	0	16	2	2	0	0
Unload	1	0	3	4	0	0	4	0	0	0	0	20
Pull after unload	0	2	0	0	0	0	1	4	0	0	0	0
Pull after remove	0	2	0	0	0	0	0	2	0	0	0	0
Ready to load	30	0	0	0	0	0	0	0	0	0	0	0

(b)

Figure 4: (a) Clean (ideal) DNA matrix (ground truth), and (b) extracted (fuzzy) DNA matrix of the warehouse operation.

4.2 The Nondeterministic (Probabilistic) DES Model

The ACD that defines the nondeterministic DES model is directly generated from the output of the HAR algorithms. As previously discussed, this ACD and the resulting DES model almost always contain fuzziness due to imperfect (noisy) sensor data and/or inaccuracy of the HAR algorithms. This fuzziness implies that unlike in the real system, each node is likely to be proceeded by more than one node. A hypothetical scenario showing a nondeterministic ACD with four activities was illustrated in Figure 1. The fuzzy precedence logic obtained for the warehouse operation experiment results in the $DNA_{extracted}$ of

Figure 4(b). In contrast to DNA_{clean} shown in Figure 4(a), some rows in $DNA_{extracted}$ contain multiple non-zero values indicating varying degrees of fuzziness.

4.3 Creating a Fuzzy DES Model in Stroboscope

The presence of uncertainty in the precedence logic requires that additional elements be introduced and used in Stroboscope to model the fuzzy ACD. Hence, a standard modeling element called fork was used in the DES model to allow multiple outgoing links from each activity resembling the fuzzy behavior. Each outgoing link emanating from a fork has a numerical strength (weight) value which determines the likelihood of that node to be selected at random upon the conclusion of a particular instance of the fork (Martinez 1996). For each fork element in the DES model, strength values were defined using the values in $DNA_{extracted}$ shown in Figure 4(b). To implement this fuzzy model, given the Stroboscope syntax, three types of nodes with different implementation mechanisms must be defined: initiation node, simple node, and termination node. An initiation node is used to start a new cycle (e.g. box moving), a termination node is used to end a cycle, and a simple node is used in all other cases. Figure 5 shows a partial ACD diagram in which these nodes are implemented. The cycle starts with initiation node 1, proceeds to simple node 1, and then randomly continues onto simple nodes 2, 3, or 4, or ends in termination node 1, which then closes the cycle. In case resources need to be regenerated at the closure of a cycle, an action event can be invoked in Stroboscope. In particular, the nondeterministic model created in Stroboscope to represent the warehouse operation experiment contains 1 initiation (modeling activity 'load'), 7 simple (modeling activities 'push to inspect', 'lift to inspect', 'inspect', 'lower', 'push to unload', 'pull after unload', and 'pull after remove'), and 2 termination nodes (modeling activities 'remove' and 'unload').

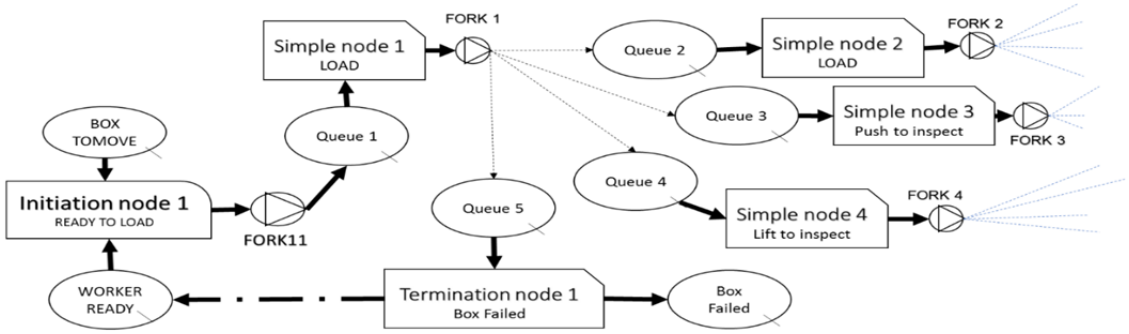

Figure 5: Partial fuzzy ACD diagram illustrating different node types.

5 REFINING THE EXTRACTED ACTIVITY TRANSITION MATRIX

5.1 Implementation of Evolutionary Algorithm

In order to transform $DNA_{extracted}$ to $DNA_{refined}$, a close-to-ideal DNA matrix, a new GA-based evolutionary technique is designed (Hassan et al. 2005) and implemented to enhance the existing capabilities of Stroboscope. GA (Reeves 2003) has been used in the past in different disciplines such as water contamination characterization (Preis and Ostfeld 2008), evaluating construction plans using data environment analysis (Torabi and Mahlooji 2017), site layout planning for construction projects (RazaviAlavi and AbouRizk 2016), and speech recognition based on random projections (Kataoka et al. 2016). In general, a GA-based method uses five key operations to reach an optimal solution from a number of possible (but not optimal) solutions (Poli et al. 2008). In this research, these five principles are implemented to refine the extracted DNA matrix, as briefly described in the following paragraphs.

- Stage 1 – Define the mother species: The $DNA_{extracted}$ generated from HAR is used as the initial mother species to create the first generation of daughter DNA matrices. Each element in the mother

matrix is regarded as the strength value of a link and is represented by $\gamma(ij)$, where i is the row index and j is the column index.

- Stage 2 – Create population of daughters: A DES model is built and run for multiple iterations to produce a population of daughter DNA matrices, producing one daughter matrix per launch. In each iteration, forks are evaluated given the strength values of their outgoing links. This results in anomalies in activity transitions leading to a population of daughter DNA matrices with a band of uncertainty, which perfectly represents the natural uncertainty in transitions.
- Stage 3 – Evaluate fitness: In this stage, daughter DNA matrices are assessed by the fitness function and receive fitness values. If this value is in the acceptable range, the corresponding daughter matrix is picked as the final matrix. In GA, this is termed the stopping condition.
- Stage 4 – Create mating pool: The mating pool is the collection of matrices used to generate the next generation of mother DNA matrices. Daughter matrices are ranked based on the fitness parameter, ω, and a subset is selected to generate the next group of mother matrices.
- Stage 5 – Produce a new generation: Once the mating pool is selected, the matrices in that pool are put through crossover, elitism, and mutation to derive six new mother DNA matrices (Davis 1991; Reeves 2003). Crossover combines parts of two or more daughter matrices, mutation changes random parts of certain daughter matrices, and elitism simply carries on daughter matrices that meet certain criteria to the next generation.

5.2 Fitness Function

The fitness function is selected based on the expected relationship between $DNA_{extracted}$ and $DNA_{refined}$. In the warehouse operation experiment, several logical observations are made to reduce the complexity of the GA and help translate and preserve the physical constraints in the intermediate transition matrices generated by the GA. In GA terms, such logical rules are called hard constraints (Chan, Chua, and Kannan 1996). As stated earlier, rows in DNA_{clean} are binary (only one non-zero element in each row) except where a decision is to be made as to where to move a resource after a decision node (e.g. inspector station). In that case, there may be more than one non-zero elements in one row. Thus, an overall binary matrix was taken as the final goal of the experiment. The node in which a decision is to be made is called a chance node. In each stage of the GA implementation, strength values of the outgoing links from a chance node are assumed to be known. For instance, in the warehouse operation experiment, the node 'inspect' is classified as a chance node; here, the inspector makes a decision on whether to accept or reject a box. Thus, in the row of the DNA matrix corresponding to this node, the number of accepted vs. rejected boxes (as observed in the experiment and recognized during the HAR step) were inserted as non-zero elements. In particular, HAR identified 20 instances of activity 'lower' (conducted by the worker immediately after the box was approved) and 10 instances of activity 'reject' (conducted by the inspector immediately after the box was rejected). Thus, 20 and 10 were used in the corresponding row of $DNA_{extracted}$.

The abovementioned hard constraints are used to define the mathematical boundaries for the GA when processing transition matrices using the steps described in previous Subsection. These constraints are expressed in Equations (1) through (4). In evaluating a DNA matrix generated by the GA, it is also noted that data collected by sensors and processed through HAR are not accurate but are still of quality sufficient to initiate the GA process. With this in mind, despite inherent fuzziness (uncertainty) in the input data, it can be assumed that in the presence of a large training dataset, when analyzing the $DNA_{extracted}$, the strongest links (large non-zero values in the matrix) are more likely to be statistically reliable, and thus should to the most extent survive (and not utterly diminish) during the refinement process. The objective of the GA is to obtain a matrix with the highest fitness parameter (Equation 3).

$$\omega_d(i) = \frac{\max(\gamma(i))}{\sum_{j=1}^{n} \gamma(ij)} \tag{1}$$

$$\omega_d = \bigvee_{i=1}^{n} \widehat{\omega_d(i)} \tag{2}$$

$$Z = \max \quad \omega_d \tag{3}$$

$$Z > 0.95 \text{ OR generation number} = 10 \tag{4}$$

5.3 Parameters of the Problem

The parameters used to produce a feasible solution mainly depend on the quality of input data, desired accuracy of final results, available computation time, and processor quality. Considering these factors, for the warehouse operation experiment discussed in this paper, the following parameters are selected:

Number of mothers in each generation: 3 Number of daughters generated by each mother: 5
Number of daughters in each generation: 15 Number of generations: 10
Acceptable parameter of fitness: 0.95

Increasing the number of mothers and/or daughters in each generation adds to the complexity and processing time of the GA. Similarly, increasing the number of GA generations can improve the accuracy of the resulting matrices while also adding to the processing time. Thus, this parameter is often the best way to control the overall computation time. Finally, since DNA_{clean} is a purely theoretical matrix and can be rarely achieved in practice, a realistic parameter of fitness is used to select the best possible $DNA_{refined}$. Once the desired fitness is reached, the process stops. It is worth noting that the criteria governing the selection of these parameters are mainly qualitative and expected to vary depending on the application.

6 RESULTS AND ANALYSIS

6.1 Evaluating the Effectiveness of GA Implementation

The developed GA implementation was applied to $DNA_{extracted}$ shown in Figure 4(b), which was used as the initial mother matrix. The first generation of mother matrices was created by processing the initial mother matrix using the GA-Stroboscope model. Each new daughter matrix was obtained by launching the GA-Stroboscope model using the proper mother matrix as input. Thus, in this case, the model was launched 5 times for each mother matrix and 15 times in each generation for 10 generations. After a total simulation time of 420 seconds and 10 generations, the $DNA_{refined}$ shown in Figure 6 is obtained.

Load	Push to inspect	Lift to inspect	Inspect	Lower	Remove	Push to unload	Unload	Pull after unload	Pull after remove	Box Failed	Box delivered	
0	30	0	0	0	0	0	0	0	0	0	0	Load
0	0	29	0	0	0	0	0	0	1	0	0	Push to inspect
0	0	0	28	0	0	0	0	1	0	0	0	Lift to inspect
0	0	0	0	20	10	0	0	0	0	0	0	Inspect
0	0	0	0	0	0	18	0	0	0	0	0	Lower
0	0	0	0	0	0	0	0	0	0	10	0	Remove
0	0	0	0	0	0	0	18	0	0	0	0	Push to unload
0	0	0	0	0	0	0	0	0	0	0	20	Unload
0	0	0	0	0	0	0	1	0	0	0	0	Pull after unload
0	0	0	0	0	0	0	1	0	0	0	0	Pull after remove
30	0	0	0	0	0	0	0	0	0	0	0	Ready to load

Figure 6: Refined (final) DNA matrix of the warehouse operation.

Comparing this DNA matrix with DNA_{clean} of Figure 4(a), it is inferred that $DNA_{refined}$ resembles DNA_{clean} more closely than $DNA_{extracted}$ of Figure 4(b). For example, most of the 17 erroneous transitions from $DNA_{extracted}$ have been treated, with only 4 fuzzy transitions remaining. Moreover, 8 out of the 11 rows are now perfectly binary as opposed to only 3 in $DNA_{extracted}$. As previously stated, a key factor in

evaluating matrices in each step of the GA is the average of the fitness function for the entire matrix. In $DNA_{extracted}$, this parameter was only 0.74 whereas in $DNA_{refined}$ it increased to 0.96, as shown in Table 1. This increase of 30% establishes the effectiveness of the GA implementation. Moreover, this value is sufficiently close to the fitness parameter of DNA_{clean} which was 0.97. Another measure of the effectiveness of the GA implementation is that with each new generation of daughter matrices, the average of the fitness parameter steadily increases, as shown in Table 1, thus implying that each iteration improves the fitness of the transition matrix.

Table 1: The Average Fitness Parameter for each generation

Generation	1	2	3	4	5	6	7	8	9	10
Average Fitness	0.74	0.8	0.84	0.85	0.87	0.89	0.92	0.95	0.96	0.96

6.2 Investigating the Impact of DNA Refinement on DES Results

The ultimate goal of refining imperfect sensor data is to produce a more reliable input for simulation modeling from a noisy input. To assess the success of achieving this goal, each of the three DNAs (clean, extracted, and refined) is used to create a DES model of the warehouse operation experiment. Each model is then run for multiple iterations (while incrementing the number of boxes to inspect) and simulation results from all three models are compared to check if the DES model built using $DNA_{refined}$ does in fact resemble the real system more closely. As presented in Figure 7 and Figure 8, two quantifiable parameters (i.e. total time to process each box, and variations in unit cost) are selected to evaluate these DES models.

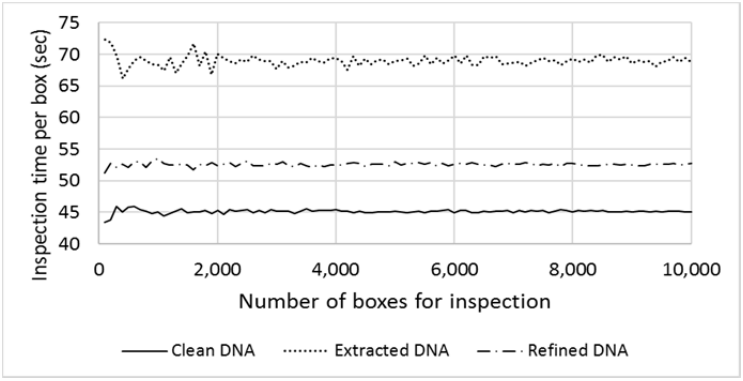

Figure 7: Analysis of inspection time per box obtained from clean, extracted, and refined DNAs.

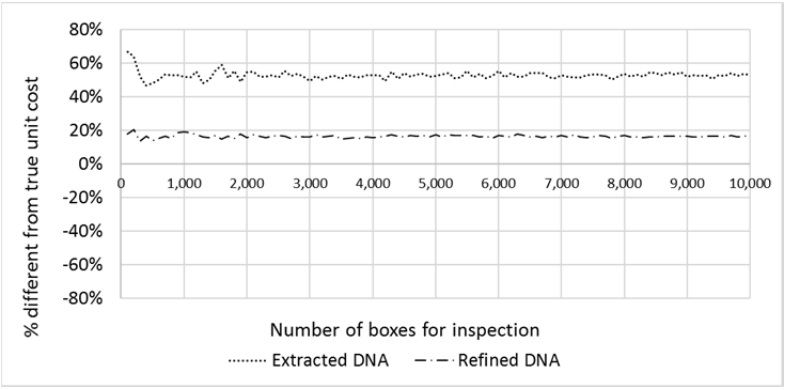

Figure 8: Analysis of unit cost discrepancy obtained from extracted and refined DNAs (baseline of 0% represents the unit cost obtained using the clean DNA).

In Figure 8, cost is obtained by considering total labor cost (one worker and one inspector) according to the Bureau of Labor Statistics (BLS) (2015) data at $15.34/hour for the worker and $33.92/hour for the inspector. As illustrated in Figure 7 and Figure 8, for both parameters, the output of the simulation model created from $DNA_{refined}$ is in closer agreement with the output of the simulation model built using DNA_{clean}, while the output of the simulation model built from $DNA_{extracted}$ is far off from the ground truth. For instance, per Figure 7, the average discrepancy in inspection time (in seconds) per box reduces from 23.8 between DNA_{clean} and $DNA_{extracted}$ to only 7.4 between DNA_{clean} and $DNA_{refined}$. Similarly, as seen in Figure 8, the discrepancy in unit cost is reduced from 52.8% on average to 16.5%. Both parameters show major improvement in the accuracy of the simulation output compared to ground truth values.

7 SUMMARY AND CONCLUSIONS

Sensing technologies have significantly expanded and become more ubiquitous thus creating a potentially expanded role in project planning, implementation, monitoring, and control. However, since most construction simulation systems lack the ability to fully capture and incorporate process-level data, the plethora of data captured from construction sites often goes unused. Another major challenge in working with sensor data is the noise in data causing a massive underutilization of data in decision-making. If used to create simulation inputs, this innate fuzziness can potentially propagate in the model and result in volatile outputs, further contributing to unreliable and inaccurate simulation results.

In this paper, the authors investigated whether low quality data captured by consumer-grade sensors can be reliably used to generate stable simulation input models. In particular, human activity data in a warehouse operation experiment was collected by smartphone sensors, and processed using machine learning methods. Next, evolutionary techniques and DES modeling was deployed to refine the noise in collected data. This resulted the fitness of the activity transition matrix (a.k.a. DNA) of the warehouse operation experiment to improve from 0.76 to 0.96 (compared with 0.97ground truth value of 0.97). The refined data was then used to create a simulation model of the operation, and the model output was compared with the ground truth in terms of two measures: total time to process each box, and variation in unit cost. Results showed that the model built from $DNA_{refined}$ outperformed the model built from $DNA_{extracted}$.

It must be added that data accuracy can be also affected by ambient factors. For instance, in outdoor applications, GPS signals can be disrupted, and data leakage can occur which in turn, induces exponential error propagation when individual sensors feed data to a large network. Given that most such issues would not be dealt effectively with improvements in sensor technology, the work presented in this paper can be of significant value to refining imperfect sensor data for a variety of data-dependent platforms. This research contributes to the body of knowledge by enabling the transformation of imperfect sensor data to cleaner datasets for generating more stable simulation models of real systems.

8 ACKNOWLEDGEMENTS

The presented work has been supported by the U.S. National Science Foundation (NSF) through grant CMMI 1602236. The authors gratefully acknowledge the support from the NSF. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily represent those of the NSF.

9 REFERENCES

AbouRizk, S., D. Halpin, Y. Mohamed, and U. Hermann. (2011). "Research in Modeling And Simulation for Improving Construction Engineering Operations." *Journal of Construction Engineering and Management*, 137(10): 843–852.

Akhavian, R. and A. H. Behzadan. (2013). "Design Requirements of an Automated Data-Driven Simulation Model Generator for Construction Operations." *Proceedings of the International Conference on Civil and Building Engineering Informatics (ICCBEI)*, 114–124.

Akhavian, R. and A. H. Behzadan. (2015). "Construction Equipment Activity Recognition for Simulation Input Modeling Using Mobile Sensors and Machine Learning Classifiers." *Advanced Engineering Informatics*, 29(4): 867–877.

Bureau of Labor Statistics (BLS). (2015). "Occupational Employment and Wages." http://www.bls.gov/oes/current/oes472061.htm Accessed: Mar. 2, 2017.

Chae, M. J., H. S. Yoo, J. Y. Kim, and M. Y. Cho. (2012). "Development of a Wireless Sensor Network System for Suspension Bridge Health Monitoring." *Automation in Construction*, 21: 237–252.

Chan, W.-T., D. K. Chua, and G. Kannan. (1996). "Construction Resource Scheduling with Genetic Algorithms." *Journal of Construction Engineering and Management*, 122(2): 125–132.

Choe, S., F. Leite, D. Seedah, and C. Caldas. (2014). "Evaluation of Sensing Technology for the Prevention of Backover Accidents In Construction Work Zones." *Journal of Information Technology in Construction (ITcon)*, 19(1): 1–19.

Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.

Golparvar-Fard, M., F. Peña-Mora, and S. Savarese. (2011). "Integrated Sequential As-Built and As-Planned Representation With D 4 AR Tools in Support of Decision-Making Tasks in the AEC/FM Industry." *Journal of Construction Engineering and Management*, 137(12): 1099–1116.

Hajjar, D. and S. M. AbouRizk. (2002). "Unified Modeling Methodology for Construction Simulation." *Journal of Construction Engineering and Management*, 128(2): 174–185.

Hassan, R., B. Cohanim, O. De Weck, and G. Venter. (2005). "A Comparison of Particle Swarm Optimization and the Genetic Algorithm." *Proceedings of 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Austin, TX.

Huang, Y. and A. Verbraeck. (2009). "A Dynamic Data-Driven Approach for Rail Transport System Simulation." In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 2553–2562. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Izadi, D., J. H. Abawajy, S. Ghanavati, and T. Herawan. (2015). "A Data Fusion Method in Wireless Sensor Networks." *Sensors*, 15(2): 2964–2979.

Kataoka, Y., T. Nakashika, R. Aihara, T. Takiguchi, and Y. Ariki. (2016). "Selection of an Optimum Random Matrix Using a Genetic Algorithm for Acoustic Feature Extraction." *Prooceeding of the 2016 Conference on Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference*, IEEE, 1–6.

Kiel, L. D. and E. W. Elliott. (1996). *Chaos theory in the social sciences: Foundations and applications*. University of Michigan Press, Ann Arbour, MI.

Lee, S., A. Behzadan, A. Kandil, and Y. Mohamed. (2013). "Grand Challenges in Simulation for the Architecture, Engineering, Construction, and Facility Management Industries." *Computing in Civil Engineering (2013)*, 773–785.

Leite, F. *et al.* (2016). "Visualization, Information Modeling, and Simulation: Grand Challenges in the Construction Industry." *Journal of Computing in Civil Engineering*, 30(6): 4016035.

Lorenz, E. N. (1963). "Deterministic Non-periodic Flow." *Journal of the Atmospheric Sciences*, 20(2): 130–141.

Martinez, J. C. (1996). *Stroboscope: State and Resource Based Simulation of Construction Processes*. University of Michigan, Ann Arbour, MI.

Martinez, J. C. and P. G. Ioannou. (1997). "State-based Probabilistic Scheduling using STROBOSCOPE's CPM Add-on." *Proceedings of the 1997 Construction Congress V, ASCE, Stuart D. Anderson, ed, Minneapolis, MN*, 438–445.

Nath, N. and A. H. Behzadan. (2017). "Construction Productivity and Ergonomic Assessment Using Mobile Sensors and Machine Learning." In *Proceedings of the ASCE International Workshop on Computing in Civil Engineering 2017:Smart Safety, Sustainability and Resilience, Seattle, WA*, 434-441.

Oloufa, A. A., M. Ikeda, and T.-H. Nguyen. (1998). "Resource-based Simulation Libraries for Construction." *Automation in Construction*, 7(4): 315–326.

Poli, R., W. B. Langdon, N. F. McPhee, and J. R. Koza. (2008). *A Field Guide to Genetic Programming*. Lulu.com.

Preis, A. and A. Ostfeld. (2008). "Genetic Algorithm for Contaminant Source Characterization using Imperfect Sensors." *Civil Engineering and Environmental Systems*, 25(1): 29–39.

Razavi, S. N. and C. T. Haas. (2010). "Multisensor Data Fusion for On-Site Materials Tracking in Construction." *Automation in Construction*, 19(8): 1037–1046.

RazaviAlavi, S. and S. AbouRizk. (2016). "Genetic Algorithm–Simulation Framework for Decision Making in Construction Site Layout Planning." *Journal of Construction Engineering and Management*, 143(1): 4016084.

Reeves, C. (2003). "Genetic Algorithms." *Handbook of Metaheuristics*, Springer, Berlin, Germany, 55–82.

Song, L. and N. N. Eldin. (2012). "Adaptive Real-Time Tracking and Simulation of Heavy Construction Operations for Look-Ahead Scheduling." *Automation in Construction*, 27.

Spencer Jr, B., M. E. Ruiz-Sandoval, and N. Kurata. (2004). "Smart Sensing Technology: Opportunities and Challenges." *Journal of Structural Control and Health Monitoring*, 11(4): 349–368.

Tannock, J., B. Cao, R. Farr, and M. Byrne. (2007). "Data-Driven Simulation of the Supply-Chain-Insights From the Aerospace Sector." *International Journal of Production Economics,* 110(1): 70–84.

Torabi, M. and H. Mahlooji. (2017). "An Integrated Simulation-DEA Approach to Multi-Criteria Ranking of Scenarios for Execution of Operations in a Construction Project." *Iranian Journal of Management Studies*, 9(4): 801–827.

Vasenev, A., T. Hartmann, and A. G. Dorée. (2014). "A Distributed Data Collection and Management Framework for Tracking Construction Operations." *Advanced Engineering Informatics*, 28(2): 127–137.

Zamalloa, M. Z. and B. Krishnamachari. (2007). "An Analysis of Unreliability and Asymmetry In Low-Power Wireless Links." *ACM Transactions on Sensor Networks (TOSN)*, 3(2): 7.

AUTHOR BIOGRAPHIES

PRABHAT SHRESTHA is a Master's student in the Department of Construction Science at Texas A&M University. He received his BSc. in Civil Engineering from Technion-Israel Institute of Technology. His current interests include construction informatics, dynamic data-driven application simulation (DDDAS), and decision-making science. His email address is prabhat1993@tamu.edu.

AMIR H. BEHZADAN is an Associate Professor in the Department of Construction Science at Texas A&M University. He received his Ph.D. in Civil Engineering (Construction Engineering and Management) in 2008 and his Master's degree in Construction Engineering and Management in 2005 both from the University of Michigan, Ann Arbor. He also holds a B.Eng. degree in Civil Engineering from Sharif University of Technology (Tehran, Iran). His current research interests include construction informatics and data analytics, autonomous construction systems, and data-driven simulation and visualization. He is a member of the American Society of Civil Engineers (ASCE) and serves on the editorial board of the ASCE Journal of Construction Engineering and Management. His email address is abehzadan@tamu.edu and his web page is http://people.tamu.edu/~abehzadan.