Inverse Generative Approach for Identifying Agent-Based Models from Stochastic Primitives

Gayani P.D.P. Senanayake[1], Minh Kieu[1]

Department of Civil and Environmental Engineering, University of Auckland, Auckland 1010, New Zealand[1]
http://www.cee.auckland.ac.nz/

Abstract. Genetic programming evolves computer programs to solve problems while showing promising avenues for discovering complex behavioural rules in agent-based models (ABMs). Since existing efforts to learn ABM structures in the field of Inverse Generative Social Science (IGSS) have concentrated on combining domain-specific primitives for deterministic rule generation, this paper evolves interpretable agent logic from scratch comprised of stochastic primitives for decision tree nodes. We show the adaptability of our approach by applying it to discover models representing human behaviour targeting data generated from existing conceptual models. Our results show that the IGSS can identify the same model as the reference model to be the most fitted rule with pseudo-truth data and other evolved solutions that closely resemble the data. Additionally, permutation accuracy proves that random selection is the most important comparator primitive, where random selection based on similarity by age was the sole variable determining exit selection for data collection. Contrary to this original rule, the selection of the closest exit and similarity with gender are found to be more important terminal primitives (factors) than the similarity by age. Our study illustrates the capacity of genetic programming to extract understandable decision rules from data for agents in ABMs that simulate behaviours across domains.

Keywords: Genetic Programming, Agent-Based Modelling, Inverse Generative Social Science, Pedestrian Behaviour

1 Introduction

In the field of agent-based simulations and generative social science, it is crucial to specify all possible intra-agent equations in order to create accurate models of social systems that can replicate actual dynamics to provide causal inference. This requires a careful understanding of human behavioural dynamics and decision-making processes. Agent-based models (ABMs) take a bottom-up approach to modelling systems, with macro-level behaviours emerging from the interactions of agents operating at the micro-level. While ABMs are capable of capturing the heterogeneity of human behaviour and providing in-depth analysis of agent interactions, the decisions made by individual agents are often guided by a predefined set of rules [1, 2]. The formulation of these rules for individual behaviours typically requires manual effort by the modeller, which can be time-consuming and heavily dependent on the domain expertise and assumptions. This leads to modeller bias and loss of simulation realism and reduces the possibility of investigating actual trends [3, 4, 5].

To address this limitation, several approaches, including modelling protocols known as pattern-oriented modelling (POM) and Machine Learning (ML) based simulation methodologies, have been proposed. POM is a technique that involves testing patterns of different theories by manually creating multiple candidate models to match a real-world scenario, which can introduce implementation risks and inconsistencies across models. Further, the repetitive implementation of multiple versions of the model to explore the rule space is laborious and lacks modularisation to clearly identify the decision-making elements that generate the desired patterns [6, 7]. Beyond simulating potential macro-level behaviours of complex systems, ABMs offer the ability to work backwards from real-world data, determining lower-level agent rules that reconstruct higher-level observed dynamics [8]. ML, on the other hand, is a data-driven approach that can enable the reverse implementation of ABM. It uses algorithms to discover patterns and make predictions based on past experiences. Many machine learning techniques act as black boxes, such as artificial neural networks, support vector machines, and random forests. Although they are powerful for predictions, these opaque models do not elucidate the decision processes or causal mechanisms that produce their outputs [9, 10, 11, 12]. They are mainly focused on predictive accuracy rather than interpretation of results. In recent years, a promising evolutionary approach to addressing this challenge has been the use of genetic programming in agent-based simulations. Genetic programming, a subset of machine learning, provides a method for automatically discovering and evolving the rules or mechanisms that govern individual agent behaviour in an ABM [13, 14]. This concept of generating rules rather than deductively testing theories to work backwards from data to underlying data-generating processes in an inductive manner while building explanations from the bottom up is called Inverse Generative Social Science (IGSS).

This emerging field of IGSS offers a novel approach to model discovery by generating potential explanatory ABMs for a target phenomenon, as opposed to modelling a singular pattern for micro-behavioural behaviour and testing its validity as is done in generative social science models [8]. This approach entails treating agent architecture as an output of the model rather than an input, and utilising machine learning algorithms to discern the rules or equations governing agent behaviour [15]. However, existing IGSS papers have also focused primarily on evolving deterministic agent rules and models keeping deterministic nodes in the decision tree [8, 16, 15, 17]. This study facilitates the development of highly stochastic ABMs properly capturing behavioural stochasticity, for instance, pedestrian behaviour and movements are inherently stochastic. There is a gap in extending IGSS to search the wider space of stochastic, not just deterministic, generative models. This paper aims to develop stochastic IGSS using strongly typed genetic programming to test pedestrian movement and navigation behaviours as the case study. The most challenging aspect of pedestrian modelling is to capture the stochastic behaviour of pedestrians when interacting with other individuals. We hypothesise that stochastic IGSS can be a more data-driven and systematic way to develop stochastic ABMs, such as a pedestrian model, rather than solely relying on predefined rules and modeller assumptions. This will expand the scope of IGSS into more behaviourally realistic mechanisms driving empirically observed class of simulation models.

This paper aims to explore the search space of decision trees in IGSS with stochastic nodes. This aim is supported by using the case study of pedestrian behaviour of people leaving the train station. In Section 2, background information on the research area and case study model is introduced in sufficient detail for the subsequent manipulation of its structure to be understood. Then, in Section 3, an overall process of model discovery is introduced, with a particular focus on the evolutionary computing tools used to search through the space of model structures. Results of the model discovery process for the case study are provided in Section 4. Finally, the findings are discussed in detail in Section 5 with future directions for research on this exciting topic.

2 Background

2.1 Related Work

Pedestrian models that leverage ABM have become an invaluable tool for analysing and predicting human movement and crowding behaviours for domains ranging from architecture and urban planning to emergency response and pandemic modelling [18, 19]. Such models capture pedestrian navigation at the individual level by specifying simple movement rules, allowing complex system-level dynamics with collision, exit selection, and congestion to emerge [20]. Most pedestrian ABMs rely on hand-crafted deterministic or probabilistic rules dictating acceleration, heading changes, collision avoidance, route choices, etc. [21]. A significant advances have been made in pedestrian modelling, while exploring the integration of stochastic elements into these agent-based behavioural decision-making systems. Early studies demonstrated adding stochastic components for navigation decisions or wait times at bottlenecks/intersections as it can improve model realism [22]. Recent empirical findings on crowd disasters highlight complex dynamics between stochastic panic behaviours and peer influence effects. This approach has been applied in various ways, such as in simulating counter flow through bottlenecks [23], enhancing wayfinding in pedestrian simulation [24], developing a stochastic transition model for pedestrian dynamics [25], and developing utility threshold model of herding-panic behaviour in evacuation [26]. However, most prevailing models take narrow approaches, modifying one or two isolated rules rather than searching the possibility space of stochastic pedestrian mechanisms [27]. This generally lacks an empirical grounding for their stipulated behavioural mechanisms [28].

The generative social sciences paradigm argues that explaining macro-scale societal patterns requires modelling the micro-level mechanisms by which they arise from individual interactions [29]. Agent-based modelling is axiomatic to this approach. Recently, the vision for "Inverse Generative Social Science" (IGSS) has been articulated, applying evolutionary computation to automate the discovery of generative ABMs. IGSS employs ML and other computational techniques to uncover the underlying mechanisms that give rise to social phenomena [30]. Evolutionary algorithms have gained traction in recent decades for automated scientific discovery across disciplines. Methodologies ranging from genetic programming to classifiers leverage Darwinian selection to evolve free-form equations, simulations, or programs according to fitness objectives [31, 32].

Existing IGSS research demonstrations have focused on matching observable patterns in residential segregation, alcohol consumption, flocking dynamics and other domains. [8, 33, 16, 34] proposed a promising framework for IGSS; Evolutionary Model Discovery (EMD), for automated discovery of explanatory individual-level mechanisms behind emergent social phenomena using genetic programming. They evolved agent decision rules to generate mixed patterns of residential segregation and integration in an ABM. Rules were represented as mathematical equations combining hypothesised causal factors like racial bias and isolation. A genetic program optimised combinations of these factors over generations to maximise a metric indicating mixed patterns. [35] applied EMD for automated rule discovery in agent-based models of irrigation cooperation dilemmas. They define micro-level decision factors and use genetic programming to search for accurate behavioural theories, evaluating predictive fit to experimental data. [15, 36, 37] has advanced the application of multi-objective evolutionary computation for the automated discovery of agent-based model structures in IGSS. Using multi-objective grammar-based genetic programming (MOGGP), they treat social science theories as modular components that are combined in different arrangements to match historical targets on alcohol consumption trends. Trade-offs emerge between model fit, complexity, and theoretical coherence. Their approach searches over and tests alternative formulations of theories like social norms and social roles, promoting theoretical diversity. Model structures balance explanatory power against interpretability, with the goal of generating empirically grounded mechanistic explanations that can inform policy. [38] demonstrated an approach to automatically learn interpretable symbolic agent-based models from basic mathematical building blocks using genetic programming. Without relying on domain knowledge, they evolve agent update rules as mathematical/logical functions that accurately replicate target flocking and opinion dynamics patterns.

Prior notable works in IGSS have focused on evolving deterministic decision rules, whether domain-specific [34] or based on basic operators [38, 37]. [35] introduced inherent stochasticity by evolving rules with random factor (terminal comparator). This paper introduces both deterministic and stochastic rules by introducing a random comparator node other than argmax and argmin nodes to the search space. Agents choose between behavioural exits deterministically or stochastically based on a probability calculated from distance, crowd and other factors. While these factors combine deterministically, final exit selection incorporates randomness - probabilistically picking min, max, or random exit. This explores whether uncertainty in strategy selection, rather than strategy execution, better explains human decisions. This approach retains the interpretability of distinct rules while capturing unpredictability. The difference is conceptually subtle - randomness enters from non-systematic switching between strategic options, not inherent randomness within strategies. But model discovery can shed light on these nuances to advance generative sufficiency. Comparing mechanisms for embedding stochasticity advances IGSS techniques for robust, generalisable social explanations.

In summary, stochasticity is critical for accurately representing the variability inherent to pedestrian behaviours. The emergent vision for IGSS offers newfound promise to automate the discovery of stochastic generative models across social science domains. However, pioneering applications have only begun targeting simpler deterministic search spaces (Epstein 2019). This paper's proposed case study, applying IGSS techniques to induce fully stochastic pedestrian movement from pseudo-truth data, will help advance the state-of-the-art. The expected contributions are manifold: substantiating the feasibility of automated stochastic model induction, enhancing the empirical validity of generative pedestrian simulations, and revealing novel behavioural insights to better design built environments.

2.2 Agent-Based Model: Station-Sim

To demonstrate IGSS concept with stochasticity, a simplified ABM of pedestrian behavior modeling, StationSim was employed, depicting generic human crowd flow. The StationSim model aims to capture key factors influencing pedestrian movement and congestion in train stations. The station layout is abstracted as a 2D grid with configurable entrances, exits, and sample agent trajectories exhibiting movement and interaction dynamics as shown in the Figure 1.

In StationSim, agents emulate passengers exiting a train and crossing a station platform to leave through one of several exits. The key entities in the model are the agents, representing individual pedestrians with attributes such as location, speed, and movement history, and the station layout with configurable dimensions, entrances, and exits. At model initialisation, N agents are instantiated and enter the rectangular platform environment at a steady rate via one of three randomly assigned entrance points. Following that, they are assigned desired exit gate location randomly at the entrance. Each agent is navigated across the platform given a maximum intended speed sampled from a Gaussian distribution, which is a common assumption in pedestrian modeling literature [39] and it simplifies the heterogeneity of real-world pedestrian populations. During navigation, faster agents encounter slower agents obstructing their paths, triggering avoidance behaviours. When such interactions occur, the impeded agent randomly selects either a left or right manoeuvre to circumvent the slower agent. This stochastic local collision response induces emergent macro-level congestion patterns that vary across model executions. The simulation concludes when all agents have entered the platform environment and successfully exited. Entrance and exit widths impose capacity constraints, restricting the volume of agents that can enter or leave in a given model time step.

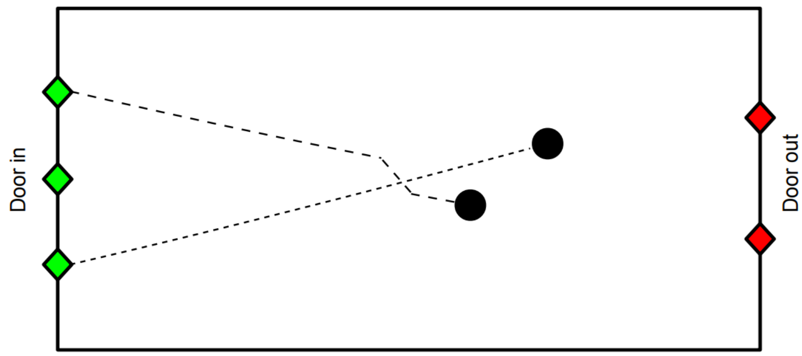

Fig. 1: The StationSim environment with 3 entrances and 2 exit doors.

The model's conceptual framework is based on established theories in social science field, such as the social force model [40] and the principle of least effort [41]. For example, the agent's navigation towards their desired exit while avoiding collisions is inspired by the social force model, which postulates that pedestrians are influenced by attractive forces towards their goals and repulsive forces from other individuals and barriers. Moreover, the model incorporates stochastic elements to represent the inherent variability in human behavior, such as random fluctuations in walking speed and direction. The model assumes a steady inflow of agents, homogeneous decision-making rules, and perfect knowledge of the station layout. While the model simplifies certain aspects of reality, such as group dynamics, queueing behavior, and multi-level structures, it captures the essential elements of pedestrian movement and interactions in a station setting. The three most important characteristics of this model are a) individual heterogeneity: each agent has a desired maximum speed, b) agent interactions: agents are not allowed to occupy the same space and try to overtake the slower agent, and c) emergence: crowding is an emergent property of the system that arises as a result of the choice of exit and their maximum speed [42, 43]. By explicitly stating these assumptions and limitations, the StationSim model provides a clear conceptual framework for understanding and interpreting the simulation results in the context of pedestrian behavioural Modelling.

As IGSS is an emmerging concept with limitted literature, this study discovers plausible behavioural rules using a simple ABM such as StationSim with psuedotruth data that incorporates stochastic variations in individual decision-making based on age. It tries to see whether IGSS can pick up the actual model rule with stochastic variations in the dataset to represent individual decision-making under uncertainty. In order to increase the complexity of decesion making we introduced some additional factors such as age, gender, distance to exits and crowd at exits that can create a more realistic representation of pedestrian behavior. Using genetic programming, an evolutionary computation technique, we evolve an exit selection rule that combines these factors to optimally capture the patterns observed in the stochastic pseudo-truth dataset, demonstrating the potential of IGSS for discovering plausible behavioral rules in ABM with limited prior knowledge.

3 Methodology

ABMs capture emergent social processes but get complex, evolutionary methods model transmission but require many design choices. Integrating them brings together individual and evolutionary dynamics for more plausible yet interpretable models [**?**]. [34, 33, 16, 8] introduced the Evolutionary Model Discovery (EMD) framework as the first novel methodology for IGSS research. EMD allows automated exploration, evaluation, and discovery of plausible micro-level agent rules that reproduce target macroscopic patterns. At its core, EMD utilises a genetic programming optimisation to evolve agent decision rules representing combinations of hypothesised behavioural factors affecting a process of interest. By representing decision rules as modular, reusable factor functions, the factor composition can be systematically mutated towards higher simulation fitness against macro-level data patterns. Each generation of alternate agent rules and resultant simulation fitness creates a dataset ameniable to causal inference via random forest feature importance analysis, quantifying the role of factors in achieving model fit. The EMD methodology was further developed by[37, 36, 17, 15]. The key advancements include formulating model discovery as a multi-objective optimisation to trade-off between data-fit and interpretability; assessing theoretical credibility of discovered structures with domain experts; developing modular software to enable testing of multiple theories; using grammar-based genetic programming for more structured search; and identifying solutions, like surrogate-assisted calibration, to address computational bottlenecks associated with parameter tuning for each candidate structure.

By extracting and manipulating the implicit ontology within an existing generative model, the authors open up the process of abductive reasoning to computational augmentation and support. Together these developments allow more systematic, theory-driven exploration for other researchers [38, 35] over wider model component spaces, while balancing empirical fit with interpretability and computation.

The IGSS approach involves the following steps as described by [30]. First, stipulating a macroscopic target pattern observed empirically or from theoretical models that one seeks to generate from the bottom-up (Step 1). For example, patterns of residential segregation, trends in alcohol consumption, flocking movements, etc. Rather than hand-design full agent rules, one then specifies more basic behavioural rule primitives and permissible ways of combining them (Steps 2 and 3). This includes factors like racial preferences, social conformity, collision avoidance and logical operators like if-then conditions, mathematical operations, nesting/recursion limits, etc. A fitness metric is chosen to compute model-target alignment, such as error distances (Step 4), and an evolutionary algorithm, such as genetic programming and grammatical evolution, is selected to search the space of rule combinations (Step 5). Evolution proceeds according to this metric until reaching preset limits (Step 6). The end result is one or more ABMs with evolved interaction rules that can successfully replicate the phenotype target pattern. In effect, IGSS inverts traditional agent-based modelling by shifting creative design from complete agent specifications to their underlying building blocks, using AI-based techniques for automated generative model discovery.

In lieu of real-world data, we utilise output from the StationSim simulation as target data for testing our approach. This allows us to validate the capability of our method to automatically reverse-engineer agent logic recapitulating provided reference data patterns. The evolved decision trees are automatically translated into modular Python code to initialise and run the evolved models for fitness evaluation.

3.1 Hypothesised alternate factors influencing exit selection decision

The agents in the original StationSim model select the exits randomly at the entrance gate. First, the StationSim model of the static exit selection rule was refined into a dynamic rule evaluated per time step. So, the IGSS is applied to the changed model rule where agent select the exit randomly at each time step. Then, the set of hypothesized factors that could influence an agent's exit choice, such as distance, crowd density were defined to be implemented as primitives in the genetic programming framework. To expand the realism of simulated human decision-making, we draw on the Agent_Zero framework proposed by [44], incorporating rational, social, and emotional dimensions into exit choice calculations. As shown in Table 1 the provided hypothesised factors are terminal nodes for the decision tree rules. We introduce each factor as a function(probability between [0,1]): compare_distance (F_d), compare_neighbouring_crowd (F_c), compare_age (F_a), and compare_gender (F_g). The categorisation of decision factors into categories provides a conceptual framework, though the mapping of specific variables can be subjective. The "rational" factor is represented by the comparison of distances to exits, reflecting the assumption that pedestrians tend to choose the nearest exitand the "emotional" and "social" factors are captured by the agents' tendency to follow the crowd or align with the behavior of similar individuals in terms of age and gender.

Table 1: Primitive set used for exit selection of StationSim model

Node	Syntax	Return Type
MinOf	min-one-of (exits)[comparator]	exit
MaxOf	max-one-of (exits)[comparator]	exit
RandomOf	random-one-of (exits)[comparator]	exit
+	comparator + comparator	comparator
-	comparator - comparator	comparator
Potentialexits	exit-x exit-y	exits
Rational factors		
CompareDistance	((distance x-of-exit y-of-exit) / totDistance)	comparator
Emotional / Social	factors	
CompareCrowd	((sum [num-agents] in-radius of 5) / totCrowd)	comparator
CompareAge	(((1-(Agent [age] - mean [age] of agents)) / maxAge) / totRelativeAge)	comparator
CompareGender	((sum [num-agents] with same (gender) of agent) / totAgents)	comparator

Next, a pseudo-truth dataset by running the StationSim model with a predefined exit selection rule was created to be used as the target dataset. The pseudo-truth StationSim model as mentioned in the Table 3 used to collect the target data utilised for the experiment. In this stochastic rule (later mentioned in the paper as "pseudo-truth rule"), each agent randomly selects an exit, but the probability distribution is weighted based on the similarity of the agent's age compared to the average age of agents near the exits. We then used the Deap library to set up the genetic programming experiment. This library parsed the StationSim model and generated a set of primitives and factors based on the provided annotations. To ensure the evolution of valid agent logic, the genetic programming components enforce strong typing between terminals nodes and the comparator nodes. As shown in Table 1, each proposed factor has an associated type definition. Comparator nodes embed code for assessing exit options on normalised sensor inputs, enabling fair mathematical operations across different factor types within the decision trees. Later they were compared by the MinOf, MaxOf, and randomOf nodes to exactly select what exit the agent selects based on exit options. By designing GP building blocks to align with programming syntax requirements, the system constrains exploration to the space of compilable agent behaviours. Legal code generation facilitates the translation of the evolved trees into executable agent-based model implementations.

For example, a combination of factors R(x) may be;

$$R(x) = F_d + F_c - F_a \tag{1}$$

Agents then decide the exit to select, $e^{'}$, using argmax, argmin, and random over the possible exits defined and considering the final probability based on R(x) value:

$$e^{'} = argminR(x) \tag{2}$$

The genetic programming algorithm evolved a population of candidate exit selection rules, represented as syntax trees, over multiple generations. Each candidate rule was evaluated by injecting it into the StationSim model, running simulations, and comparing the resulting exit distributions to the pseudo-truth data using a fitness function. The evolutionary process employed tournament selection, crossover, and mutation operators to create new generations of candidate rules. Finally, we analyzed the evolved rules and their fitness scores to identify the most promising exit selection strategies and assess the importance of different factors in reproducing the target patterns. Table 3 lists the most fitted "candidate rules" for the pseudo-truth rule.

For model simplicity, we assume agents have full information on all exit sites. The goal is to minimising divergence between simulated and target data on agent counts who selected each exit over time. Specifically, equation 3 defines model error as the sum over all time steps t of the squared differences between the simulated (Y) and actual agent ($Y^{'}$) counts exiting through each exit (e) at each time step (i). Individual GP tree fitness scores report aggregate error across a 1000-time step StationSim model evaluation. By explicitly fitting agent logic to match exit distributions, the system evolves behaviours governing exit choice to reconstruct emergent system-level patterns exhibited in the reference data.

$$Loss = \sum_{i=1}^{t} (Y_{i,e} - Y_{i,e}^{'})^2 \tag{3}$$

4 Results

We executed three independent genetic programming runs to evolve the StationSim agent exit selection logic. The configuration applied 20 generations with a population of 10 tree individuals, using crossover and mutation to explore the space of possible decision rules. Each candidate tree was translated into a complete StationSim implementation and simulated for 1000 steps. The aggregate deviation between the agent exit distributions and reference outputs determined error. We set key evolutionary parameters as 0.8 crossover rate, 0.2 mutation rate, minimum tree depth of 2, and maximum depth of 10. The minimum depth allows the first layer to be a random, minimum, or maximum function, with the second layer introducing a single factor influencing choice. Through iterative evaluation and refinement of agent behaviour trees, the system converges on rule sets replicating emergent circulation patterns from the original model.

Table 2 lists the top 10 evolved decision trees with corresponding error scores, ranked by similarity to the reference data. As expected, most appearing rule in the result is to select the exit gate randomly based on similarity by age. Figure 2 displays the mean and standard deviation of aggregate loss value of each generation decreasing over generations as agent behaviours improve. This close match confirms the capability to accurately reverse engineer the

Table 2: The 10 top candidate models

Run	Gen	Rule	Agg: Error
0	15	Exit = Randomly based on the similarity by age	4446.27
0	4	Exit = Randomly based on the similarity by age	4713.41
2	17	Exit = Minimum based on the combination of difference between (similarity by gender and crowdness) and combination of (distance and similarity by age)	4839.07
2	13	Exit = Minimum based on the distance	4926.21
0	3	Exit = Randomly based on the similarity by age	4960.11
0	10	Exit = Randomly based on the similarity by age	4967.73
0	19	Exit = Minimum based on the distance	5001.91
2	5	Exit = Randomly based on the similarity by age	5009.29
0	18	Exit = Randomly based on the similarity by age	5029.51

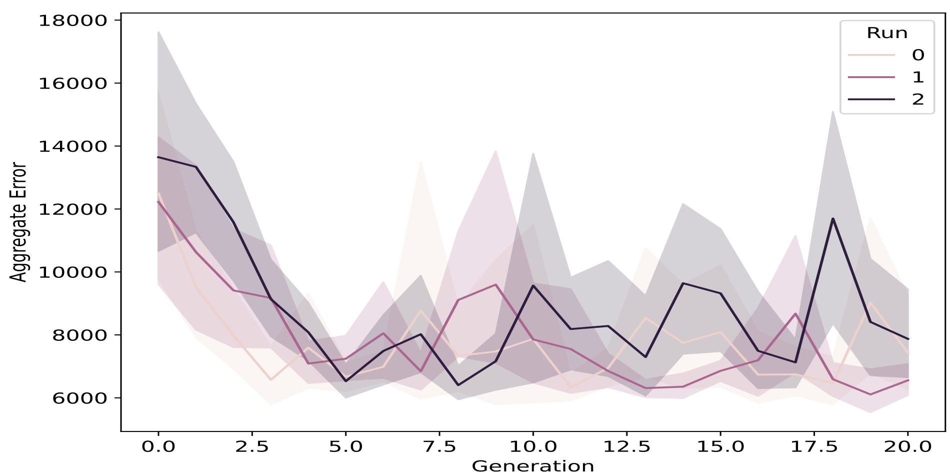

Fig. 2: Mean Aggregate Error over Generations

key exit choice logic governing circulation dynamics in the original StationSim model. The full genetic programming experiment required approximately 20 hours of computation time.

Figure 3 displays the Aggregate error of the best agent logic discovered in each genetic programming run over generations. Convergence to low error solutions appears gradual, likely due to the limited generations and pseudo-truth data set based on a stochastic rule. Figure 4 plots the size of the optimum trees found so far per generation in each run, indicating one run maintains compact decision rules while the others evolve between maximum allowed depth mostly in early generations. Between runs, variability in tree shape and speed of fitness improvement by lowering the error highlights the stochastic nature of genetic programming optimisation. Additional computational budget for longer runs and larger populations could improve solution quality and convergence rates.

For easy Visualisation in later plot, the unique candidate rules in the Table 2 are shortlisted in the Table 3 where pseudo-truth rule and the candidate Rule 1 are the same.

Figure 5 and Table 4 exhibit simulation results from three top evolved decision rules (rules 1, 3, and 4 in the Table 3) compared against each other across 100 simulation runs. These rules were selected based on their performance in reproducing the patterns observed in the pseudo-truth dataset, which was generated using Rule 1. In this analysis, Rule 1 serves as the benchmark, representing the ”ground truth” or the target behavior we aim to replicate. The other evolved rules (Rules 3 and 4) are evaluated based on how closely they match the performance of Rule 1.

When compared to Rule 1, Rule 2 has a similar median aggregate error score. Interestingly, the Rule 2 spread is fully contained within the Rule 1 distribution, indicating greater consistency. This shows Rule 2 can almost replicate Rule 1’s performance, serving as a plausible explanatory model even if falling short of ground truth decisions. Conversely, Rule 3’s mean is considerably lower at 7710.24, with a higher deviation of 1579.33. So it fails to approach Rule 1 aggregate outcomes while retaining a similar variability profile. Concretely, Rule 2 approximates truth better statistically, while Rule 3 diverges on overall accuracy but emulates variance characteristics. Given these

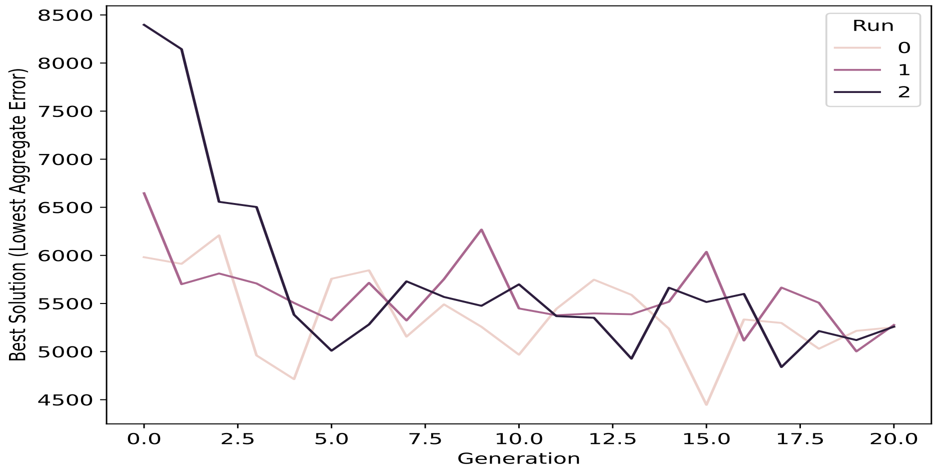

Fig. 3: The best model by generation in each genetic programming runs

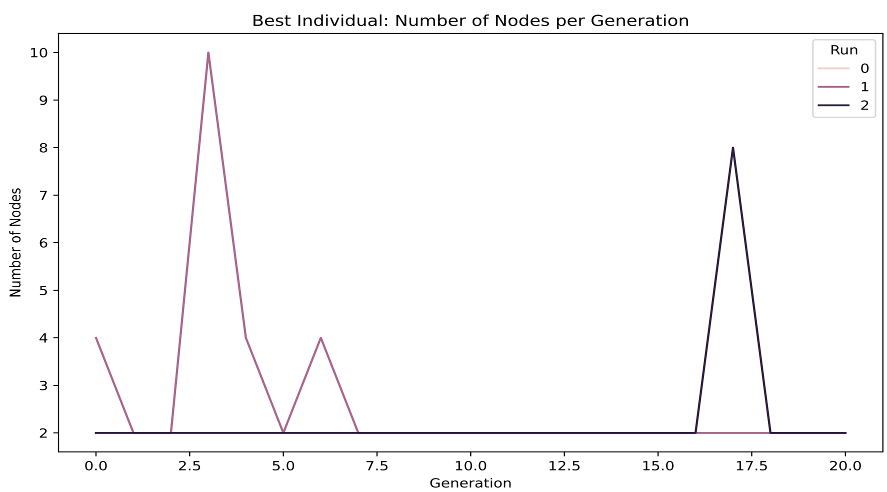

Fig. 4: Size of the best individuals

data details, Rule 2 provides a feasible decision theory not far from ground truth, whereas Rule 3 does not explain the core patterns well despite matching uncertainty levels.

Factor importance obtained through permutation accuracy importance techniques can be seen in Figure 6. It is calculated by randomly shuffling the presence values of each factor in isolation and quantifying the resulting decrease in error prediction performance of the trained random forest model. The result indicated compare_distance as the most influential contributor to model fitness (error reduction), with Similarity_by_gender also showing a stronger role in exit choice behaviour than the other hypothesised factors.

5 Discussion

Our study goes beyond a mere demonstration of a specific parameter setup and simulation execution. By employing the Inverse Generative Social Science (IGSS) approach with genetic programming, we aim to shed light on the fundamental mechanisms underlying pedestrian exit selection behavior in train stations. The StationSim model serves as a flexible platform to explore the space of plausible behavioral rules and test hypotheses about the factors influencing pedestrian decision-making. The evolutionary process of rule discovery allows us to uncover novel and emergent strategies that may not be readily apparent from traditional deductive approaches. The rules governing

Table 3: The unique top three candidate rules

Model	Exit Selection Rule
Pseudo-truth rule (for generating target data)	Exit = Randomly based on the similarity by age
Rule 1 (candidate rule)	Exit = Randomly based on the similarity by age
Rule 3 (candidate rule)	Exit = Minimum based on the combination of difference between (similarity by gender and crowdness) and combination of (distance and similarity by age)
Rule 4 (candidate rule)	Exit = Minimum based on the distance

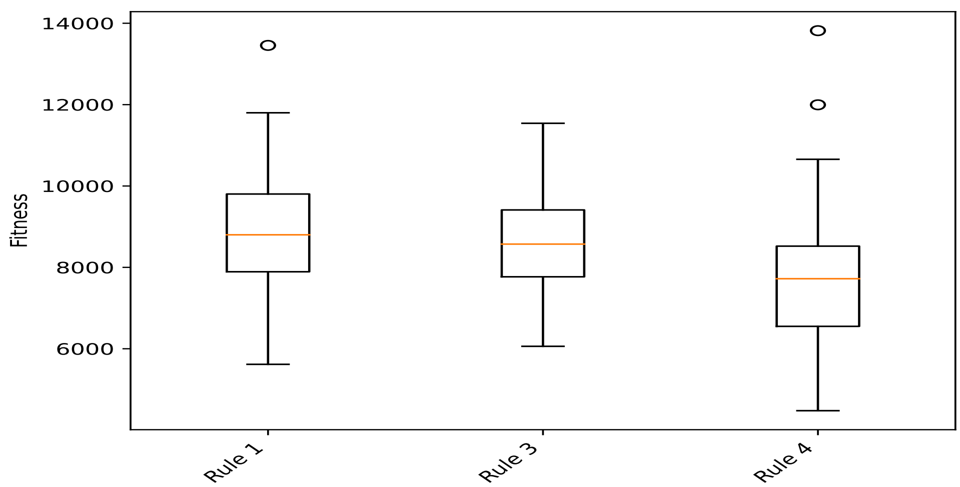

Fig. 5: Comparisons of 100 samples of three of the top-performing models

agents in current applications of IGSS thus far have been limited to deterministic specifications. Developing techniques to effectively search over stochastic mechanisms remains an open challenge for realising the full promise of generative modelling. Advancing stochastic IGSS methods suited to capturing the inherent randomness of human behaviour and social systems represents an important frontier.

In this study, we utilised genetic programming to evolve Python-based ABMs. We reanalysed and augmented a set of ABMs focusing on individual behaviour in exit selection as simulated via a pseudo-truth data set. We explored a broader set of potential decision rules while introducing the stochastic nodes to the decision trees than was previously done using deterministic rules. We found that IGSS can pick up the model with the same stochastic rule used for collecting pseudo-truth data as the most fitted model. Further, the resulting other models were able to display nearly identical macro behaviours to our reference models.

The concept of predictable human behaviours overlaid with random perturbations has appeared across behavioural modelling literature. [45] conceptualised this idea as agents following general decision tendencies, but imprecise reasoning and uncertainty inject stochastic elements. Later, [35] incorporated a stochastic term in evolving utility-based choice rules with IGSS. Their work simply adds a random factor (terminal node), where the selection primitives (comparator node) stayed deterministic to select between get_max or get_min. Additionally,

Table 4: Mean and Standard Deviation (SD) of best models

Rule	Mean	SD
gate_out=self.randomly_select_exit(model,(similarity_by_age))	8843.97	1422.62
gate_out=self.get_min_select_exit(model,(self.combine(self.subtract(similarity_by_gender neighbourhood_count_exits),self.combine(compare_distance,similarity_by_age))))	8595.48	1208.07
gate_out=self.get_min_select_exit(model,(compare_distance))	7710.23	1579.32

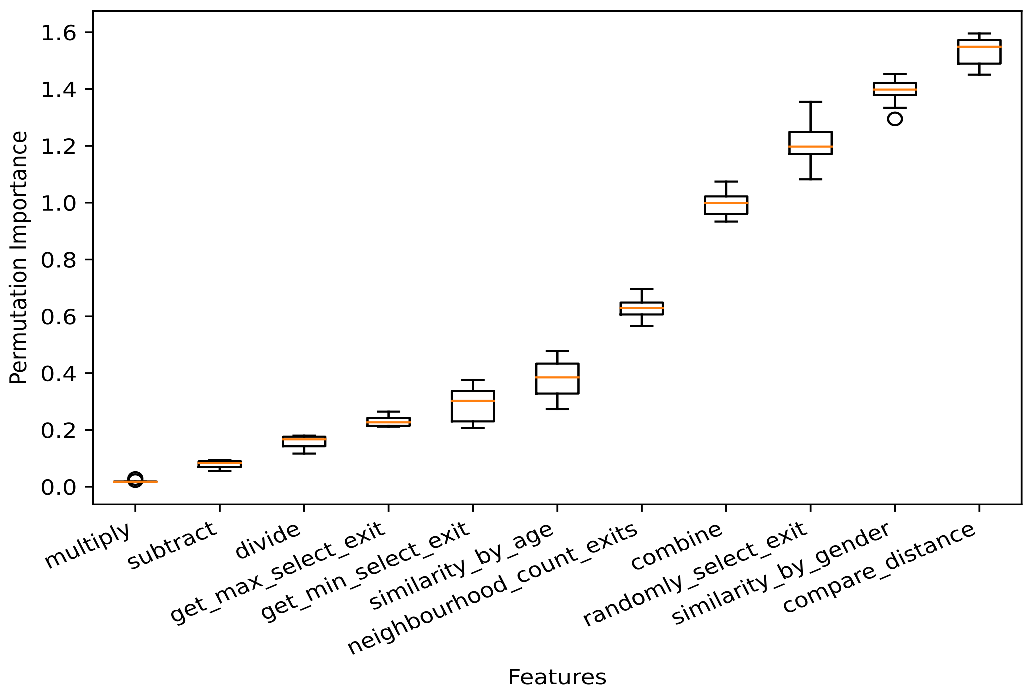

Fig. 6: Feature Importance Analysis

we introduced get_random, where the agent randomly selects exits based on the probability calculated by factor combinations.

There are additional insights to be had from the models generated and subsequent factor analysis. We note that rule 4, which is based on the distance factor in Table 2 represents an interesting outlier in the top performers as illustrated in Figure 5 and table 4 , as it shows a lower medium and mean error value when compared to rule 1, which is the benchmark to the specific pseudo-truth data set considered in rule evolution. On the other hand, contrary to the reference model exit selection behaviour of the StationSim model, where an agent would randomly select the exits in relation to age, the permutation accuracy importance results show that agents are most likely to random comparator over min and max, but distance as the most important factor over similarity by age.

In this genetic algorithm experiment seeking to evolve stochastic rules, the randomness in Rule 1 could allow it to achieve high error scores against the pseudo-truth by chance alignments, but the deterministic rules seem to do better when evaluated across many repeated trials. Further, reliance on a single static pseudo-truth data mapping for fitness evaluation might have led to overfitting and optimising the rules too closely to that specific pseudo-truth data. Though Rule 1 scored best on fitness against that target data, Rules 3 and 4 may be more robust rules in general. There may be gene interactions happening in the genetic algorithm evolution process that lead to unexpected emergent behaviours. The evolved rules may exploit certain combinations and nonlinearities.

To promote the evolution of broadly generalisable stochastic rules, multiple mechanisms must be employed to overcome this training overfitting effect. Specifically, k-fold cross-validation should be used so that rules are evaluated against multiple held-out data partitions rather than just a singular target output mapping. Additionally, explicit incentives such as bonus fitness rewards for error reduction should be incorporated for stochastic behaviours exhibited in promising rule candidates across evaluations. Other enhancements include expanding the rule search space complexity and forcing rules to generalise across iteratively updated pseudo-truth data. By emphasizing rule generalization beyond narrow accuracy gains on limited training distributions, the genetic algorithm can better identify robust and widely applicable stochastic rules for simulation modelling.

Currently, all the agent models being tested assume that everyone follows the same decision-making strategy. Allowing different agents to use different strategies could potentially improve heterogeneity in agent behaviour and might lead to models that fit the human data even better. Further, as [15] suggested exploring the whole model without just evolving one sub-model will uncover the true potential of IGSS. Future work could introduce agent heterogeneity and apply IGSS to learn personalised collision avoidance behaviours with the StationSim model.

This work adds to a growing body of literature [34, 37, 35, 38] demonstrating the promise of IGSS for agent-based social science research. By automating the discovery of the micro-level rules able to recreate emergence, these methods can strengthen model accuracy, plausibility, and generalisability compared to purely theoretical specifications. As IGSS matures to handle richer data and more complex collective dynamics, it may provide fundamental new insights into causal mechanisms in social systems. However, care must be taken to avoid overfitting and ensure model interpretability. Overall our results confirm the value of genetic programming for the next generation of generative social science focused on policy-relevant explanations over purely predictive functions.

Bibliography

[1] B Zhang, W K V Chan, and S V Ukkusuri. On the modelling of transportation evacuation: an agent-based discrete-event hybrid-space approach. *Journal of Simulation*, 8(4):259–270, November 2014. ISSN 1747-7778, 1747-7786. https://doi.org/10.1057/jos.2014.3. URL `https://www.tandfonline.com/doi/full/10.1057/jos.2014.3`.

[2] Jennifer Badham, Edmund Chattoe-Brown, Nigel Gilbert, Zaid Chalabi, Frank Kee, and Ruth F. Hunter. Developing agent-based models of complex health behaviour. *Health & Place*, 54:170–177, November 2018. ISSN 13538292. https://doi.org/10.1016/j.healthplace.2018.08.022. URL `https://linkinghub.elsevier.com/retrieve/pii/S1353829218301047`.

[3] Vashti Galpin, Natalia Zoń, Pia Wilsdorf, and Stephen Gilmore. Mesoscopic Modelling of Pedestrian Movement Using Carma and Its Tools. *ACM Transactions on Modeling and Computer Simulation*, 28(2):11:1–11:26, March 2018. ISSN 1049-3301. https://doi.org/10.1145/3155338. URL `https://dl.acm.org/doi/10.1145/3155338`.

[4] Xilei Zhao, Ruggiero Lovreglio, and Daniel Nilsson. Modelling and interpreting pre-evacuation decision-making using machine learning. *Automation in Construction*, 113:103140, May 2020. ISSN 09265805. https://doi.org/10.1016/j.autcon.2020.103140. URL `https://linkinghub.elsevier.com/retrieve/pii/S0926580519313184`.

[5] D. Li and J. Zhong. Dimensionally Aware Multi-Objective Genetic Programming for Automatic Crowd Behavior Modeling. *ACM Transactions on Modeling and Computer Simulation*, 30(3):19:1–19:24, July 2020. ISSN 1049-3301. https://doi.org/10.1145/3391407. URL `https://dl.acm.org/doi/10.1145/3391407`.

[6] Volker Grimm, Eloy Revilla, Uta Berger, Florian Jeltsch, Wolf M. Mooij, Steven F. Railsback, Hans-Hermann Thulke, Jacob Weiner, Thorsten Wiegand, and Donald L. DeAngelis. Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology. *Science*, 310(5750):987–991, November 2005. ISSN 0036-8075, 1095-9203. https://doi.org/10.1126/science.1116681. URL `https://www.science.org/doi/10.1126/science.1116681`.

[7] Uri Wilensky and William Rand. Making models match: Replicating an agent-based model. *Journal of Artificial Societies and Social Simulation*, 10(4):2, 2007.

[8] Chathika Gunaratne and Ivan Garibay. Alternate social theory discovery using genetic programming: towards better understanding the artificial anasazi. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 115–122, Berlin Germany, July 2017. ACM. ISBN 978-1-4503-4920-8. https://doi.org/10.1145/3071178.3071332. URL `https://dl.acm.org/doi/10.1145/3071178.3071332`.

[9] Ning Wu and Elisabete A. Silva. Artificial Intelligence Solutions for Urban Land Dynamics: A Review. *Journal of Planning Literature*, 24(3):246–265, February 2010. ISSN 0885-4122. https://doi.org/10.1177/0885412210361571. URL `https://doi.org/10.1177/0885412210361571`. Publisher: SAGE Publications Inc.

[10] Sharad Sharma, Kola Ogunlana, David Scribner, and Jock Grynovicki. Modeling human behavior during emergency evacuation using intelligent agents: A multi-agent simulation approach. *Information Systems Frontiers*, 20:741–757, 2018.

[11] William Rand. Theory-Interpretable, Data-Driven Agent-Based Modeling. In *Social-Behavioral Modeling for Complex Systems*, pages 337–357. John Wiley & Sons, Ltd, 2019. ISBN 978-1-119-48500-1. https://doi.org/10.1002/9781119485001.ch15. URL `http://onlinelibrary.wiley.com/doi/abs/10.1002/9781119485001.ch15`. Section: 15 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119485001.ch15.

[12] Runhe Zhu, Burcin Becerik-Gerber, Jing Lin, and Nan Li. Behavioral, data-driven, agent-based evacuation simulation for building safety design using machine learning and discrete choice models. *Advanced Engineering Informatics*, 55:101827, 2023.

[13] V. Anne Smith. Evolving an Agent-Based Model to Probe Behavioral Rules in Flocks of Cowbirds. In *ALIFE*, volume 2008, pages 561–568, 2008.

[14] Jinghui Zhong, Linbo Luo, Wentong Cai, and Michael Lees. Automatic Rule Identification for Agent-Based Crowd Models Through Gene Expression Programming. 2014.

[15] Tuong Manh Vu, Charlotte Probst, Joshua M. Epstein, Alan Brennan, Mark Strong, and Robin C. Purshouse. Toward inverse generative social science using multi-objective genetic programming. *Genetic and Evolutionary Computation Conference : [proceedings]. Genetic and Evolutionary Computation Conference*, 2019:1356–1363, July 2019. https://doi.org/10.1145/3321707.3321840. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7569505/`.

[16] C. Gunaratne and I. Garibay. Evolutionary model discovery of causal factors behind the socio-agricultural behavior of the Ancestral Pueblo. *PLoS ONE*, 15(12 December), 2020. ISSN 1932-6203. https://doi.org/10.1371/journal.pone.0239922.

[17] Tuong M. Vu, this link will open in a new window Link to external site, Charlotte Buckley, Hao Bai, Alexandra Nielsen, Charlotte Probst, Alan Brennan, Paul Shuper, Mark Strong, and Robin C. Purshouse. Multiobjective Genetic Programming Can Improve the Explanatory Capabilities of Mechanism-Based Models of Social Systems. *Complexity*, 2020, 2020. ISSN 10762787. https://doi.org/http://dx.doi.org.ezproxy.auckland.ac.nz/10.1155/2020/8923197. URL `http://www.proquest.com/docview/2412810415/abstract/D2D51F9C1D1D4EEFPQ/4`. Place: Hoboken, United Kingdom Publisher: Hindawi Limited.

[18] Yuhong Zhou. Agent-based modeling and simulation for pedestrian movement behaviors in space: a review of applications and gis issues. In *Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Geo-Simulation and Virtual GIS Environments*, volume 7143, pages 508–516. SPIE, 2008.

[19] Francesca Camillen, Salvatore Caprì, Cesare Garofalo, Matteo Ignaccolo, Giuseppe Inturri, Alessandro Pluchino, Andrea Rapisarda, and Salvatore Tudisco. Multi agent simulation of pedestrian behavior in closed spatial environments. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, pages 375–380. IEEE, 2009.

[20] Ioannis Karamouzas, Peter Heil, Pascal Van Beek, and Mark H Overmars. A predictive collision avoidance model for pedestrian simulation. In *Motion in Games: Second International Workshop, MIG 2009, Zeist, The Netherlands, November 21-24, 2009. Proceedings 2*, pages 41–52. Springer, 2009.

[21] Markus Schratter, Maxime Bouton, Mykel J Kochenderfer, and Daniel Watzenig. Pedestrian collision avoidance system for scenarios with occlusions. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1054–1060. IEEE, 2019.

[22] Caterina Ramos-Moreno, Ana Maria Ruiz-Teran, and Peter J Stafford. Impact of stochastic representations of pedestrian actions on serviceability response. In *Proceedings of the Institution of Civil Engineers-Bridge Engineering*, volume 174, pages 113–128. Thomas Telford Ltd, 2021.

[23] Jicai Dai, Xia Li, and Lin Liu. Simulation of pedestrian counter flow through bottlenecks by using an agent-based model. *Physica A: Statistical Mechanics and its Applications*, 392(9):2202–2211, 2013.

[24] Giuseppe Vizzari, Luca Crociani, and Stefania Bandini. An agent-based model for plausible wayfinding in pedestrian simulation. *Engineering Applications of Artificial Intelligence*, 87:103241, 2020.

[25] Michael Schultz. Stochastic transition model for pedestrian dynamics. In *Pedestrian and Evacuation Dynamics 2012*, pages 971–985. Springer, 2014.

[26] Jinghong Wang, Manman Chen, Wenyu Yan, Youran Zhi, and Zhirong Wang. A utility threshold model of herding–panic behavior in evacuation under emergencies based on complex network theory. *Simulation*, 93(2): 123–133, 2017.

[27] Eleonora Papadimitriou, George Yannis, and John Golias. A critical assessment of pedestrian behaviour models. *Transportation research part F: traffic psychology and behaviour*, 12(3):242–255, 2009.

[28] Raymond C Pitts. Reconsidering the concept of behavioral mechanisms of drug action. *Journal of the Experimental Analysis of Behavior*, 101(3):422–441, 2014.

[29] Joshua M. Epstein. Agent-based computational models and generative social science. *Complexity*, 4(5):41–60, 1999. ISSN 1099-0526. https://doi.org/10.1002/(SICI)1099-0526(199905/06)4:5¡41::AID-CPLX9¿3.0.CO;2-F. URL `http://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-0526%28199905/06%294%3A5%3C41%3A%3AAID-CPLX9%3E3.0.CO%3B2-F`. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291099-0526%28199905/06%294%3A5%3C41%3A%3AAID-CPLX9%3E3.0.CO%3B2-F.

[30] Joshua M. Epstein. Inverse Generative Social Science: Backward to the Future. *Journal of Artificial Societies and Social Simulation*, 26(2):9, 2023. ISSN 1460-7425. https://doi.org/10.18564/jasss.5083. URL `https://www.jasss.org/26/2/9.html`.

[31] Ron Wehrens and Lutgarde MC Buydens. Evolutionary optimisation: a tutorial. *TrAC Trends in Analytical Chemistry*, 17(4):193–203, 1998.