REPORT FOR

Public Transport Mobility Simulation's project

PREPARED BY
BOUHAOUALA Sana
BOUSSIF Melek
BOUGHZALA Ghailene

TABLE OF CONTENTS

ABSTRACT

In the dynamic landscape of transportation networks, where connectivity and real-time information wield significant influence, the role of modeling and simulation stands out as pivotal.

Simulation and modeling constitute integral components in the analysis of transportation networks. Given the widespread utilization of personalized, up-to-the-minute information sources, the efficacy of simulations hinges significantly on the individualized responses of travelers to the received information. Consequently, the simulation model must adopt an individual-centric approach, and within this context, an agent-based simulation emerges as the most promising paradigm. With the current personalization of information, simulations must account for the interplay among individually guided passengers.

This paper introduces a multiagent simulation model designed to observe and evaluate the repercussions of real-time information provision on passengers within transit networks. These consequences are gauged through the simulation of various scenarios, each contingent on the proportion of passengers connected to a real-time information system. Within the system, passengers and vehicles are represented as distinct agents. The analysis focuses on the simulated scenarios and their impact on passengers' travel times. The information disseminated to connected passengers is rooted in a space-time representation of the transportation networks.

The findings indicate that real-time personalized information can exert an increasingly positive influence on overall travel times with the ascending ratio of connected passengers. Nevertheless, a ratio threshold exists, beyond which the positive impact of real-time information diminishes.

INTRODUCTION

Effective formulation of mobility policies requires decision-makers to rely on support systems to facilitate their decision-making processes.
Simulation is a crucial tool in this context, enabling various strategies and scenarios to be tested without real-world traffic repercussions. However, the escalating complexity of transportation systems, characterized by the proliferation of intelligent and mobile entities, poses challenges.
Modern transportation systems incorporate the ubiquity of passengers' mobile devices and connected vehicles, providing real-time information that significantly influences their behavior. This allows for delivering optimal itineraries and real-time updates aligned with the dynamic network status, encompassing factors such as congestion, breakdowns, and accidents.
Despite the positive impact of providing passengers with real-time traffic information, our results underscore a critical observation: beyond a specific ratio of equipped travelers with smartphones, the benefits of personalized information diminish.

I. STATE OF THE ART

The design and implementation of a multi-agent system for simulating passenger movements and public transport in urban environments have gained prominence in transportation research. Numerous studies have explored the intricacies of simulating dynamic urban scenarios, aiming to capture the complexities inherent in passenger behaviors and public transport operations. Here, we present an overview of the state of the art in this domain, drawing insights from existing literature to inform the development of a comprehensive multi-agent simulation system.

- **System Parameters Definition:** Research by [Flötteröd et al., 2014] has delved into the effective definition of system parameters in multi-agent simulations, emphasizing the importance of incorporating realistic variables such as passenger density, vehicle capacities, and road network characteristics. These parameters play a crucial role in shaping the dynamics of urban transportation systems.
- **Representation of the City:** Studies like [Balmer et al., 2004] work on urban modeling using grids provide valuable insights into representing the cityscape within a multi-agent simulation. Using grid-based structures allows for the efficient modeling of roads, bus stops, and other critical urban elements, providing a foundation for accurate and scalable simulations.
- **Public Transport Vehicle Representation:** The representation of public transport vehicles within a multi-agent system has been a focal point in recent research. [Gao et al., 2012] explore the integration of predefined timetables and routes for buses, aligning with the project's objective. Understanding how vehicles adhere to schedules contributes significantly to the realism of the simulation.
- **Passenger and Vehicle Movements (Agent Behaviors):** The behavior of agents, passengers, and vehicles, is a key aspect of any multi-agent simulation. [Bazzan and Klügl, 2009]'s work on agent-based modeling of urban transportation highlights the importance of defining realistic movement patterns, considering factors such as mode choice, pedestrian dynamics, and public transport boarding/alighting behaviors.

I. STATE OF THE ART

Selecting an appropriate simulation platform or methodology is critical in developing a multi-agent system for simulating urban passenger movements and public transport. The choice of platform profoundly influences the system's capabilities, performance, and scalability. Several simulation platforms have been explored in the literature, each offering unique advantages based on the specific requirements of the simulation.

.a. Agent-Based Modeling (ABM) Platforms:

- NetLogo:
  - **Strengths**: NetLogo is a widely used platform for ABM, offering a user-friendly interface and a robust modeling environment. Its simplicity makes it accessible for beginners in agent-based modeling.
  - **Limitations**: While NetLogo excels in ease of use, it may face challenges in scalability and performance for complex simulations.
- AnyLogic:
  - Strengths: AnyLogic supports multiple modeling paradigms, including agent-based modeling. It provides a graphical interface, facilitating model creation through a drag-and-drop approach.
  - Limitations: AnyLogic, being a commercial tool, may pose financial constraints. Its learning curve can also be steep for those new to simulation modeling.

b. Programming Languages:

- Java:
  - Strengths: Java is a versatile language, known for its performance and scalability. It has been employed in various simulation frameworks, offering a balance between performance and ease of development.
  - Limitations: Java's verbosity and boilerplate code can be deterrents for rapid development, potentially hindering the implementation speed of complex

- **Python**:
  - **Strengths**: Python's readability, extensive libraries, and thriving community make it an ideal candidate for simulation projects. Libraries like Mesa and SimPy offer powerful tools for agent-based modeling.
  - **Limitations**: Python might face performance challenges compared to lower-level languages like C++, particularly in scenarios requiring high computational efficiency.

c. Visualization Libraries:

- Matplotlib (Python):
  - **Strengths**: Matplotlib is a powerful data visualization library in Python that provides flexibility for static visualizations. It integrates seamlessly with other Python libraries commonly used in simulations.
  - **Limitations**: Matplotlib may fall short compared to specialized game development libraries for dynamic and interactive visualizations.
- Pygame (Python):
  - Strengths: Pygame is specifically designed for game development and offers a straightforward approach to creating interactive and dynamic simulations. It excels in real-time visualizations and user interactions.
  - Limitations: Pygame might be considered overkill for static visualizations, but its strengths shine in scenarios requiring real-time updates and a visually engaging user interface.

While the choice of platform depends on specific project requirements, Python emerges as a formidable contender for urban mobility simulations. Its ease of use, extensive libraries (including Mesa and SimPy for agent-based modeling), and the availability of Pygame for graphical visualization make Python a versatile and accessible choice. The balance between performance and development speed positions Python as an optimal platform for projects where interpretability, rapid prototyping, and a strong community ecosystem are paramount. In the context of this project, Python's suitability is further validated by its adept handling of the complexities inherent in simulating urban mobility dynamics.

II.METHODOLOGY AND PLATFORMS:

Selecting an appropriate simulation platform is a critical decision in developing a multi-agent system for simulating urban passenger movements and public transport. The choice of platform profoundly influences the system's capabilities, performance, and scalability. Several simulation platforms have been explored in the literature, each offering unique advantages based on the specific requirements of the simulation.

The successful execution of the project relies on a meticulous selection of methodology and platforms, ensuring the simulation's accuracy, efficiency, and user-friendliness. This section elucidates the chosen methodology, programming language, and visualization tools, elucidating the rationale behind each decision.

a. Methodology:

The methodology adopted for this urban mobility simulation centers around constructing a sophisticated multi-agent system. This approach encapsulates the interactions between passengers and public transport vehicles in a dynamic city environment. The decision to employ a multi-agent system arises from its ability to model complex, decentralized systems, capturing the emergent behaviors of individuals and vehicles within the urban landscape. The simulation can emulate real-world dynamics by representing each passenger and public transport vehicle as autonomous agents, allowing for a nuanced exploration of urban mobility scenarios.

b. Programming Language:

Python stands as the language of choice for implementing the simulation. Several compelling factors underpin this decision:

1. **Readability and Expressiveness:**
    - Python's clear and concise syntax enhances code readability, which is crucial for collaborative development and maintenance.
2. **Extensive Libraries:**
    - Python boasts a wealth of libraries conducive to simulation and modeling tasks. Adopting libraries such as Mesa and SimPy provides robust tools for agent-based modeling, simplifying the implementation of complex systems.

II.METHODOLOGY AND PLATFORMS:

3 . **Community Support:**
- Python's large and active community ensures continuous support, facilitating problem-solving and knowledge sharing throughout the development lifecycle.

4. **Versatility:**
- Python's versatility aligns seamlessly with the diverse needs of the project, from simulation logic to data analysis and visualization.

c. Visualization Tools:

The choice of visualization tools is paramount in effectively conveying the simulation dynamics. In this project, the decision to use **Pygame** as the primary visualization library is rooted in its suitability for real-time, interactive simulations. Pygame provides a game development framework offering the following advantages:

1. **Real-Time Updates:**
   - Pygame's design facilitates real-time updates, which is crucial for visualizing the dynamic movements of passengers and public transport vehicles.
2. **User Interaction:**
   - The user-friendly nature of Pygame enables the incorporation of interactive elements, allowing users to engage with and comprehend the simulation intuitively.
3. **Rich Graphical Capabilities:**
   - Pygame's capabilities in rendering graphics, handling sprites, and managing animations contribute to the creation of visually compelling representations of bus paths, disturbances, and passenger interactions.
4. **Cross-Platform Compatibility:**
   - Pygame's cross-platform compatibility ensures a consistent user experience across different operating systems.

II.METHODOLOGY AND PLATFORMS:

d. Simulation Parameters:

The parameters defining the simulation are meticulously chosen to strike a balance between realism and computational efficiency. These parameters include the size of the city, the spatial distribution of bus stops, and the introduction of disturbances. The city size and bus stop distribution emulate real-world scenarios, while disturbances add a layer of complexity to evaluate the system's resilience in the face of challenges.

This comprehensive methodology, combined with the strategic selection of Python and Pygame, ensures that the simulation accurately models urban mobility and provides an engaging and informative user experience. The synergy of these elements sets the stage for a simulation that is both scientifically rigorous and accessible to a diverse audience, contributing meaningfully to the understanding of urban transportation dynamics.

e- How it works?

In this simulation, we have designed five essential classes, each playing a crucial role in creating a dynamic and interactive city environment.

- The "City" class is the foundation, defining the layout of roads and intersections.

- "BusStop" represents significant locations where passengers can board and alight buses. Passengers, embodied by the "Passenger" class, have the autonomy to choose their mode of transportation, either walking or waiting for a bus.

- The "PublicTransportVehicle" class encapsulates the behavior of buses, each with its unique route, following the shortest path to reach designated stops. Additionally, buses have a maximum passenger capacity, adding a realistic constraint to the simulation.

II.METHODOLOGY AND PLATFORMS:

The orchestration of these elements occurs within the "CitySimulation" class, serving as the central hub for initiating and managing the entire simulation process.

Furthermore, the simulation introduces two types of vulnerabilities – road issues or bus breakdowns. This reflects real-world scenarios where unexpected challenges can affect transportation systems. Passengers face the decision to either walk or patiently wait for a bus, emphasizing the simulation's dynamic nature. A key aspect is the calculation of the minimal waiting time, ensuring passengers experience the shortest possible delay in reaching their destinations.

Additionally, the simulation incorporates a time frame of three seconds to manage and resolve all vulnerabilities efficiently. This time constraint adds an element of urgency and realism to the simulation, mimicking the need for prompt resolution in real-world scenarios. Overall, this comprehensive set of classes and features aims to create a captivating and authentic simulation, offering users a rich experience in understanding and exploring urban dynamics.

III.EXPERIMENTS AND RESULTS

Our experimentation with the urban mobility simulation, grounded in the Python code provided, sought to unravel the dynamics between public transport vehicles, passengers, and the evolving city landscape. The simulation featured a city with roads, bus stops, and the introduction of disturbances, all woven together to mirror real-world urban transportation scenarios.

a. Simulation Dynamics:

- **Initialization and City Structure:**
    - The simulation initialized a virtual city with defined road networks, bus stops, and the potential for disturbances in inaccessible routes.
    - Bus stops were strategically placed, and a dynamic road network set the stage for vehicles and passengers to navigate.

- **Vehicles and Passengers:**
    - Public transport vehicles, each following a specific route, were introduced into the city. These vehicles moved from one bus stop to another, adapting their positions based on the defined routes.
    - Passengers, with randomly assigned origins and destinations, contributed a dynamic element to the simulation. They could walk or take public transport, navigating the city's landscape.

b. Disturbances and Inaccessible Routes:

- **Dynamic Disturbances:**
    - Disturbances were introduced dynamically by blocking some roads, creating inaccessible routes for vehicles and passengers.
    - This simulated real-world scenarios, such as road closures or disruptions, adding a layer of complexity to the interactions within the urban environment.

c. Graphical Representation:

- **Matplotlib Visualization:**
    - Matplotlib visualized the evolving city landscape, including roads, bus stops, disturbances, vehicles (public transport), and passengers.
    - Different markers and colors were used to distinguish between bus stops, inaccessible routes, roads, vehicles, and passengers.

III.EXPERIMENTS AND RESULTS

- **Real-Time Updates:**
  - The simulation was visually represented in real-time, allowing for a step-by-step observation of how the city dynamics unfolded with each iteration.
  - The graphical representation served as a dynamic and informative tool for tracking the movements and interactions of vehicles and passengers.

d. Simulation Output:

- **Travel Time Assessment:**
  - The simulation tracked the travel times of each passenger, measuring the number of steps taken to reach their destinations.
  - The output provided insights into how disturbances, modes of transportation (walking or public transport), and the evolving city structure influenced the travel times of individual passengers. f. Experimentation Results:
  - The experiment showcased the dynamic nature of urban mobility, with passengers adapting their routes based on accessible bus stops and the introduction of disturbances.
  - Travel times vary based on individual choices, disturbances, and the evolving city structure, offering a nuanced view of the interconnected dynamics in play.

- **g. Insights and Further Exploration:**
  - The simulation provided insights into the immediate impact of disturbances on travel times and laid the groundwork for further exploration.
  - Iterative experimentation could uncover patterns, optimize public transport routes, and explore the cascading effects of disturbances on the overall efficiency of urban transportation systems.

The provided Python code and its visual representation serve as a versatile platform for exploring and understanding the complexities of urban mobility dynamics. The experiment results illuminate the multifaceted interactions between public transport vehicles, passengers, and the evolving city environment, setting the stage for deeper investigations and potential optimizations in urban transportation planning.

IV CHALLENGES AND SOLUTIONS:

The evolution of any project is marked by challenges that demand adaptive strategies and inventive solutions. Our journey in developing the urban mobility simulation was no exception. Below, we delve into the challenges encountered, the strategies to address them, and the enhancements that fortified the project.

a. Graphical Display Issues:

- **Challenge:** The initial hurdle manifested as graphical display issues, hindering the accurate portrayal of bus and passenger movements. This affected the fidelity of the simulation and, consequently, the user's ability to interpret and engage with the model effectively.
- **Solution:** Swift adjustments were made to the Pygame implementation, guided by iterative testing and user feedback. This involved revisiting the rendering logic, optimizing sprite movements, and ensuring synchronization between simulation dynamics and graphical updates. The result significantly improved the accuracy and smoothness of the graphical representation, aligning more closely with the underlying simulation dynamics.

b. Disturbance Fine-Tuning:

- **Challenge:** The introduction of disturbances, though conceptually powerful, posed challenges in aligning them precisely with their intended impact on travel time. Achieving a realistic and nuanced representation of disturbances required fine-tuning to ensure they genuinely mirrored real-world scenarios.
- **Solution:** Meticulous problem-solving efforts were invested in refining the disturbance scenarios. Parameters such as the frequency and intensity of disturbances were adjusted based on simulated outcomes and feedback from trial runs. This iterative refinement process resulted in disturbances that more accurately simulated disruptions in the urban transportation network, providing a richer and more realistic simulation experience.

IV. CHALLENGES AND SOLUTIONS:

c. Iterative Problem-Solving:

- **Challenge**: The overarching challenge was the need for iterative problem-solving across various aspects of the project. This included addressing unforeseen complexities in the multi-agent system, refining simulation parameters, and enhancing the overall user experience.
- **Solution**: A meticulous and iterative problem-solving approach was adopted. This involved continuous testing, feedback collection, and subsequent adjustments to the simulation logic, graphical representation, and disturbance modeling. Each iteration brought the project closer to its objectives, with each challenge serving as an opportunity for enhancement. This iterative refinement not only resolved challenges but also contributed to the adaptability and resilience of the project.

d. Adaptability and Resilience:

- **Challenge**: Though inevitable, challenges often test a project's adaptability and resilience. This project faced diverse challenges, necessitating a dynamic and resilient response to ensure continued progress and improvement.
- **Solution**: The adaptability and resilience of the project were evidenced in the proactive response to challenges. Each challenge was met with a strategic and informed solution, ensuring that setbacks became stepping stones for improvement. The project's capacity to absorb feedback, iterate on solutions, and evolve in response to challenges underscored its adaptability, resulting in a more robust and visually captivating simulation.

In navigating these challenges, the project overcame obstacles and thrived on the learning opportunities they presented. The iterative nature of the solutions ensured that each development phase was an incremental enhancement, ultimately contributing to the project's adaptability and resilience. As a testament to the project's fortitude, the solutions implemented stand to its capacity to overcome obstacles and deliver a more refined and compelling urban mobility simulation.

V. DISCUSSION

The urban mobility simulation project, encapsulated in Python code and visually depicted through Matplotlib, emerges as a comprehensive exploration of the dynamics inherent in the interactions between passengers and public transport within a simulated city environment. This discussion serves as an in-depth assessment, encompassing code structure, challenges encountered, results obtained, and the overall evaluation of the project.

a. **Strengths and Realism:**

- The simulation excels in capturing the dynamic nature of urban mobility, presenting passengers with the choice to adapt travel modes and routes in response to disturbances and bus stop accessibility.
- Matplotlib's graphical representation serves as a robust tool, offering clear visual insights into the dynamic evolution of the city landscape.
-

b. **Code Modularity and Readability:**

- The code's modular structure, leveraging classes for City, PublicTransportVehicle, and Passenger, enhances readability and maintainability. Each component's behavior is encapsulated, contributing to code transparency.
- Adequate documentation provides clarity on class methods and key functionalities, facilitating understanding for collaborators and future developers.

c. **Challenges Encountered:**

- Initial challenges in graphical display were swiftly addressed through iterative testing and adjustments to the Pygame implementation. Disturbances, while conceptually potent, required fine-tuning to align more precisely with their intended impact on travel time.
- Meticulous problem-solving and iterative refinement stand as a testament to the project's adaptability and resilience.

V. DISCUSSION

d. **Results and Impact Assessment:**

- The simulation unfolds a dynamic narrative of urban mobility, unveiling nuanced insights into the intricate dance between passengers, public transport vehicles, and the evolving city landscape.
- Disturbances, strategically introduced, yield a realistic representation of disruptions, impacting travel times and showcasing the interconnected dynamics within an urban environment.

e. **Code Optimization and Scalability:**

- As the project evolves, there's potential for exploring optimization algorithms for public transport routes and assessing the scalability of the simulation to larger city sizes. These enhancements could lead to a more comprehensive analysis.
-

f. **Further Experimentation and Research Opportunities:**

- The simulation stands as a launchpad for ongoing research into urban transportation planning. Iterative experimentation could uncover patterns, optimize public transport routes, and explore the cascading effects of disturbances.
- Research opportunities abound, including the exploration of machine learning algorithms for predictive modeling and decision-making, ultimately elevating the sophistication of the simulation.
-

g. **Interdisciplinary Applications and Community Engagement:**

- Beyond its educational value, the simulation holds potential applications in urban planning, aiding policymakers in evaluating the effectiveness of public transport systems and the resilience of cities to disturbances.
- Encouraging community engagement could foster collaborative improvements, transforming the simulation into a shared resource for researchers, developers, and urban planners.

VI. CONCLUSION

The urban mobility simulation project, anchored in Python and Matplotlib, not only offers valuable insights into urban transportation dynamics but also presents a versatile platform for ongoing research and practical applications. Its foundations in modularity, realism, and adaptability make it a resourceful tool for diverse stakeholders in the realm of urban planning and mobility studies. The project stands poised for continuous evolution, promising deeper explorations into the challenges and opportunities of urban transportation.

Thank you
for you

ATTENTION