Proposal of ABM Traffic Thesis

Joep Saris

February 2025

1 Personal Information Student

Name	Joep Saris
Student number	s1094356
E-mail address	joep.saris@ru.nl

2 Daily supervisor information

Name	Yuzhen Qin
Affiliation	Machine Learning & Neural Computing at Donders Institute
E-mail address	yuzhen.qin@donders.ru.nl

3 Second reader information

Name	Marcel van Gerven
Affiliation	Radboud Professor AI, Principal Investigator at Donders
E-mail address	marcel.vangerven@donders.ru.nl

4 Project Description

4.1 Title of project

4.2 Abstract

I am making a simple and scalable representation of a traffic problem in which given a cluster of junctions, cars are sampled that traverse from a random starting point to a destination, traversing intersections, and following traffic rules to get there. At the same time traffic lights are sampled resulting in a problem where cars minimize individual time taken, and lights minimize the collective time taken by the cars. This grid-world representation for traffic problems would expand ABMax, Mr. Chaturvedi's agent-based modeling framework using JAX, the state of the art when it comes to parallel operations.

4.3 Project Description

Traffic is an integral part of a thriving economy. Transportation access is listed as one of the five pillars of economic development [3]. If infrastructure is of such importance, it is a logical goal to optimize for traffic to maximize efficiency, which can be done through traffic modeling.
My thesis project aims to construct an environment where you can make a traffic problem of multiple

junctions fed into each other. This problem could then be used to play around with traffic flows, finding where the bottlenecks are, and allowing traffic lights to find lighting patterns that minimize waiting time. How much more effective are optimal traffic lights compared to traffic lights operating on static patterns, and what is the trade-off between waiting time and number of computations?

Traffic modeling is a complex system. For a modeling system to be complex, it must involve numerous interacting agents whose aggregate behaviors are non-linear; hence, it cannot simply be derived from summation of individual components' behavior [4]. In traffic models, traffic jams occur, a non-linear emerging phenomenon, making traffic models complex systems.

For tackling complex systems, agent-based modeling (ABM for short) is the workhorse. ABM is a computational modeling technique that represents individual agents in a system, along with the rules governing their behavior and interactions[2]. The beauty of ABM is that your agents can be as simple or as complex as necessary, and agents can represent anything from people and animals to companies or machines. ABM agents have autonomy: the ability to make their own decisions, given their perceptions and goals. Different agents can have different parameters, which lead to different behavior, facilitating diversity. Finally, due to agents' interactivity: affecting each other and the environment, complex behavior emerges. In our case, traffic jams.

In the past, not many agents could be simulated simultaneously. However, that changed. Making use of parallel operations this number could be vastly increased. JAX is a high-performance computing library for Python. It offers this scalability in the set of agents by leveraging Just-in-Time (JIT) compilation, enabling simulations to run multiple orders of magnitude faster. Traditional ABM implementations often rely on slow, sequential processing, but JAX transforms Python code into optimized machine instructions, allowing for parallel execution on CPUs, GPUs, or TPUs. Key features like vmap automatically vectorize operations across agents, eliminating costly loops, while pmap distributes computations across multiple processors for even greater efficiency. By integrating JAX into ABM workflows, researchers can simulate considerably larger sets of agents in real time, enabling deeper exploration of emergent behaviors and complex interactions. The approach removes computational bottlenecks, allowing for more extensive scenario testing and real-time experimentation. This thesis project will making use of Abmax, the exact approach described. Abmax is an integration of JAX into ABM, focused on dynamic population size [1]. Its capabilities:

- Adding and removing an arbitrary number of agents.
- Searching and sorting agents based on their attributes.
- Updating an arbitrary number of agents to a specific state.
- Stepping agents in a vectorized way.
- Running multiple such simulations in parallel

The ABM Traffic Thesis project will consist of two parts, the first one being environment construction and explanation, and the second one the simulations and conclusions that follow from them. The environment allows you to pick a traffic map from a list of pre-made ones. Alternatively, you could randomly generate a map or create one yourself. The traffic map is a grid-world of Cells. This grid-world representation means that a Car is always in exactly one Cell, moving through them one by one. The map consists of a number of junctions and roads with a set number of lanes, all of these being composites of Cells. Among the normal Cells, there's designated start and destination Cells. Once the simulation starts, Cars get added and placed in a random selection in the set of start cells, and their goal is to reach a random selection in the set of destination cells. They find the route shortest in time, and then navigate the lanes and junctions accordingly. During the simulation, junctions can be normal intersections or they could be managed by traffic lights, another type of agent. Traffic lights manipulate from which lanes cars are allowed to go onto the intersection, and what turns they may take. The traffic lights' goal is to minimize total within-intersection waiting time. When this all works as described, some additional features such as U-turns, driver chaos and emergency vehicles get added, and then we will move onto the simulations in which a comparison gets made between waiting time with traffic agents and with static traffic light patterns.

4.4 Schedule

Table 1: Project schedule

Task description	Time estimated	Deadline
Finalize proposal.	5 hours	14/03/2025
Construct robust Cell, Map representation	15 hours	16/03/2025
Implement shortest route algorithm	5 hours	18/03/2025
Implement parallel car movement	5 hours	20/03/2025
Create Car agent	3 hours	23/03/2025
Create Traffic Light agent	3 hours	30/03/2025
Add additional features and finalize Environment	15 hours	13/04/2025
Start thesis drafting process	N.A.	16/04/2025
Conduct research on simulations	5 hours	23/04/2025
Prepare draft thesis for feedback and send it to supervisor	20 hours	10/05/2025
Edit draft thesis following feedback	5 hours	24/05/2025
Hand in draft thesis and plan thesis defence session.	1 hour	25/05/2025
Submit finalized thesis.	10 hours	01/06/2025
Defend thesis	N.A.	xx/06/2025

5 Relevance

There have been no traffic ABM problem that makes use of parallel processing before, and none of the grid-world ABM problems have intersections without traffic lights, so this project would mean an advancement in science. While the intersections without traffic lights might not sound like a big step forwards, it allows an intersection of roads to be traversed from all four entrances to all four exits without requiring a single update in cell direction. Making use of parallel processing in the context of traffic would enable us to much faster and more efficiently test traffic maps to find bottlenecks so that we could potentially find the most optimal setup for a real-world scenario such that car emission pollution and travel time are reduced, limiting environmental damage and enabling a good economy.

6 References

References

[1] I-M-Iron-Man. *GitHub - i-m-iron-man/abmax: Abmax is an agent-based modelling framework in Jax, focused on dynamic population size.* URL: https://github.com/i-m-iron-man/abmax/tree/master?tab=readme-ov-file.

[2] Miller James. "Agent-Based Modeling: How to Simulate the Behavior of Large Systems". In: *Simultech* (Oct. 2024). URL: https://www.simultech.org/agent-based-modeling-how-to-simulate-the-behavior-of-large-systems/.

[3] Dr. O'Hara. *The Five Pillars of Economic Development.* May 2018. URL: https://docs.udc.edu/causes/Five-Pillars-DC-Final-05-2018.pdf.

[4] Luis Mateus Rocha. *Complex systems modeling.* URL: https://casci.binghamton.edu/publications/complex/csm.html.