

CX 4230 / CSE/ECE 6730 Project 2

Modeling and Simulation Life Cycle

Due Dates:

- Project teams / project description: due 11:59 PM, Tuesday, March 12, 2019
- Project Checkpoint: due 11:59 PM, Monday, April 1, 2019 [no fool'n, and no extensions!]
- Final Report: 11:59 PM, Tuesday April 23, 2019 [again, no extensions!]

The goal of this project is to exercise the steps of the modeling and simulation study life cycle for a specific problem. Parts of the problem statement and many aspects of the conceptual model are provided in this document. These are incomplete, and it is up to you to fill in the details. You will need to complete the other steps: conceptual model development, input data collection and analysis, simulation model development, verification and validation, design of experiments, output analysis, and documentation of your work and findings in a written report.

A second goal of this project is to compare different discrete event simulation approaches to simulate a common physical system. This project focuses on simulating a small road transportation network, but the goal of the project is to understand simulation methodology and software development, not to understand transportation systems.

You will work in a team of 2 or 3 students. Each member of your team will develop a separate simulation model of the same system using a different approach, as specified later in this document. You will need to compare the results produced by your simulators, e.g., as part of the validation process. Team members should jointly perform all the other steps in the study and jointly develop a *single* written report documenting all of your work on the project.

Your simulation software must be written in one of the following high-level programming languages: C, C++, Java, Python, or FORTRAN. The different simulators developed by your team need not be written in the same programming languages. The simulations must be executable on a Georgia Tech server so that the TAs can independently verify it works correctly. You may use software packages such as Matlab or Excel for standard data analysis tasks or generating graphs, but not for the simulator itself. You may not use simulation packages or code downloaded from the web for the project. Your simulators should be largely developed “from scratch,” though you may use built-in libraries and data structures provided by the language such as random number generators or priority queues to implement the future event list.

The main deliverables for this project are (1) a project report documenting your efforts on the project and (2) software for the simulations your team develops. The project report should include sections for each of the major steps in the modeling and simulation life cycle (problem statement, conceptual model, input analysis, simulation model, etc.), and document your team's efforts in each step.

1 Problem Statement

The goal of this simulation study is to assess the average travel time for vehicles to traverse a portion of Peachtree Street, the corridor from 10th to 14th street, in midtown Atlanta. You need only simulate traffic in one direction. The analysis will be focused on generating the distribution of vehicle travel times as they traverse this stretch of Peachtree Street for different levels of traffic (vehicle arrivals per minute). Once you get into building the actual simulation models you will discover that there are different scenarios or model parameter settings that will impact travel time. As such, you will need to revise this problem statement to match the specific scenarios your team will study.

2 Conceptual Model

The simulation models for this project will be based on queueing network models and cellular automata, as discussed below. To complete the description of the conceptual model you will need to think through details of the simulation models your team will develop, and document (1) assumptions you are making for this study, (2) details concerning the model not specified below, and (3) enumeration of simplifications you are making. While it is difficult to capture all of these, especially the assumptions and simplifications, your team should put some thought into this in order to better understand the simulation models you are developing and their limitations. These must be documented in your project report.

3 Input Analysis

You will need to collect information concerning the road network such as the length of road segments, location of signals, number of lanes of traffic, etc. Use online resources or other publicly available information. Some information such as details concerning the movement of vehicles, and specifically the routing of vehicles at each intersection (go straight or turn) may not be readily available. You will need to come up with an approach to model such aspects. Be sure to document your assumptions in the project report.

You will need to develop a model for traffic arrivals into the Peachtree corridor. Your model must be able to simulate different levels of traffic. Use the NGSIM (Next Generation Simulation) data set for this purpose. The NGSIM data set provides trace data for vehicles traveling through the area modeled in the simulation study as well as traffic signal timing data. Use a portion of this data set to create an *empirical distribution* of the interarrival time for traffic entering the region that will be used to generate traffic in your simulation model. The remaining data can be used to validate the simulation models.

4 Simulation Model

Each member of your team will develop a separate simulation model for the road network. For two-person teams, one person should develop a simulator based on a queueing network model using an event-oriented world view. The second should develop a simulation based on a cellular automata approach. For three-person teams, two individuals should do the same as the two-person team, and the third should develop a queueing network model, but using either the activity scanning or the process-oriented world view. The three simulators are briefly described as follows:

1. *Event-oriented queueing model.* The Peachtree corridor is modeled as a queueing network where each intersection is modeled as a server, and vehicles must queue while waiting to enter the intersection. An event-oriented discrete event simulation should be developed that includes separate software modules for the simulation engine and the queueing network model. The simulation engine includes a future event list to hold scheduled, but not yet processed events using a priority queue data structure. All information specific to queueing networks or vehicle traffic should be restricted to the simulation application software module, making the simulation engine independent of the application to which it is applied.
2. *Cellular automata model.* The corridor is modeled as a cellular automata where each section of each lane of the road is modeled as a cell that is either empty, or contains a single vehicle. Vehicles move from cell to cell in traveling through the road network using certain movement rules that are encoded into the simulation. See [1] for a description of these movement rules. The simulator uses a time-stepped approach to advance simulation time. Some teams developed a cellular automata traffic model as part of their first project. Such teams need to propose extension(s) to this model that will be developed for this project.

3. *Activity-scanning or process-oriented queueing model.* This model is similar to the event-oriented queueing model described above, and should produce identical or nearly identical results, but is implemented using either the activity scanning or process-oriented world view. The activity scanning approach is described in class; see also Chapter 4 of the Birta-Arbez textbook. You may use a time stepped or event-oriented implementation approach. The process-oriented approach was also described in class, but we only recommend this approach to more experienced programmers, or those already familiar with programming using threads. In either case, the software will be very different from the simulator developed using the event-oriented world view.

The simulation must be stochastic, i.e., use random numbers to model unpredictable or unknown elements of the system such as vehicle inter-arrival times. You will need to identify the places where randomness is appropriate to use in the model, and denote where you used them in your implementation in the final report.

The software must be developed in a modular fashion, with well-defined, documented application program interfaces (APIs). For example, the software must include a priority queue for the future event list and random number generator for the required probability distributions. You will likely also want to include a library of model objects, e.g., queues. Your software may also include test “driver” programs as needed to verify correct operation of library modules.

5 Verification and Validation (V&V)

You must verify and validate each simulator. With regard to verification, in your final report document how you verified that your simulator reproduced the behaviors defined by your conceptual model. This discussion need not be lengthy, but it does need to be sufficiently detailed to convince the reader (i.e., the graders) that you did a credible job in verifying your software.

Validate your simulation models by first comparing the results predicted by your simulation model with the portion of the NGSIM data set not used to construct your empirical distribution. Of course, this implies configuring your simulations with parameter settings to match the traffic included in the NGSIM study.

Second, compare the results produced by the two (or three) different simulators both for the NGSIM scenario and other scenario(s) of your own choosing. Do your simulators produce results that seem consistent with changes in the scenario? For example, one would expect longer travel times for higher levels of traffic. While one would not expect the queueing model and cellular automata simulations to produce identical results, you should compare the results that are produced, and discuss the extent to which they agree, or explain significant differences.

Your report should document your validation efforts, and (hopefully) convince the reader that the results produced by your simulations are credible, or at least explain unexpected results to demonstrate you understand the behavior of the simulator(s). Note that each of the simulators your team develops needs to be verified and validated, but it would be useful to use a common approach for V&V that is applied to all the simulators.

6 Design of Experiments and Output Analysis

Since these are stochastic simulations you will need to perform multiple runs and determine confidence intervals for your simulation results. You will also need to determine how long to execute the simulations. What about the “warm up” period for the simulations? What did you do here? In addition, you will need to construct sets of experiments to address the goals specified in the problem

description. Your experiment design needs to consider what simulation runs and parameter settings (e.g., levels of traffic intensity) you will complete.

7 Reporting Results

Your report should include sets of graphs documenting the results of the simulation study and suitable discussion of the results. Since your team developed more than one simulator, the results need to be compared, and explanations included in the report as needed. Your presentation of results should be designed so it is easy to compare the different simulators, e.g., by plotting results produced by the simulators together on the same graph rather than separately. The report should document all of your efforts on the project.

8 Teams

You will work in teams of two or three students. You are free to work with whomever you wish; you may use the same team members as were used in the first project noting teams must be either 2 or 3 students, or you may form an entirely different team. You will need to develop a plan to divide the work among team members, and figure out how to coordinate, e.g., to make sure the results that are produced are comparable you need to make sure the different simulators use the same input parameters. One person in your team should be designated as the lead who will have the responsibility for organizing and coordinating the team, scheduling meetings, etc.

9 Deliverables

Your grade will be determined largely by the quality of the work you completed in each of the steps of the M&S life cycle, in addition to the quality of the simulation software you developed. The following items that must be completed and turned in on canvas:

- *Project teams and project description.* To complete this step, simply state who is working on your project team. Also, state who will serve to role of team leader. If no member of the team worked on developing a cellular automata simulation for the first project, that is all that is required. If one or more members did work on such a model, a brief proposal (approximately half a page) should be submitted explaining how you will enhance this model for this project.
- *Project checkpoint.* By the time of the checkpoint you should have completed the problem description (a more detailed version of the described here), conceptual model, and simulation model steps of the simulation project life cycle. You should have some version of your simulation models up and running, though it may not be the final version, and may not have been fully verified or validated yet. These should be written up, and turned in. The project checkpoint represents a large portion of the final report. You will add upon this document to complete the final report, described below.
- *Software implementation.* Source code and documentation for all software in your simulation must be turned in with the project checkpoint and final report. Be sure to document and cite any reference you used, and any pre-existing software you used in your simulation model rather than develop from scratch, e.g., random number generators or data structures.
- *Final report.* The final report includes all documentation for the software and simulation study. It should describe the work you completed for all steps of the M&S life cycle in sufficient detail that the reader could reproduce your results. All work you completed for the project must be documented in this report. The report should include a separate section of each step in the M&S lifecycle, and document your efforts in each step. The final version of

the source code for your simulator must also be provided. The final report should be self-contained.

Your written reports should be concise, and to the point focusing the work you did on the project. Include more detailed information as appendices attached to the end of the report.

10 Grading

The project is relatively open-ended, so there is much latitude for you to vary your scope of effort in completing the project. In general terms, our grading standards are as follows. If you do an acceptable job completing all of the steps of the M&S life cycle based on material covered in class or provided to you, that will earn a “B” for the project. To get an “A”, we’ll be looking for you to complete some aspect(s) “above and beyond.” Each of the steps of the life cycle offers opportunities to excel and impress, e.g., by demonstrating strong knowledge of an area through your own investigations (e.g., well beyond assigned readings), or particularly careful and well-thought out execution of the life-cycle. Part of your work on the project is to identify areas where you feel you can excel, and document your efforts in the project deliverables. We suggest you and other members of your team identify aspects that are of particular interest to you, and devote special attention to that area. Impress us, and shows us what you can do!

Each person will be assigned a grade based on the project as a whole as well as your specific contribution. Thus it is in your best interest both to do whatever is necessary to ensure the overall effort is successful, as well as to make sure you have specific contributions that you can call your own. Team members will be given an opportunity to comment on the work completed by other team members, and these inputs will be factored into grading on a case-by-case basis.

11 Additional Materials

Support materials are available for you to use in this project (available on Canvas):

- *Airport simulation code*. Code for the event-oriented discrete event simulation of an airport discussed in class. This is written in the C programming language. You may use any or all of this software for your project, or translate the code to another programming language if you are not using C.
- *NGSim Data*. The NGSim data set with trace data for vehicles on Peachtree, as well as associated documentation.
- *Papers*. Reference [1] discussing the cellular automata approach to model traffic.

12 Final Comments

This simulation problem is, by design, open-ended and not all details have been specified. While you will be expected to do some data collection, you will not have all the information you need, and some assumptions and approximations will have to be made. Part of your task is to identify and document what assumptions you are making, and have reasonable justifications for the choices you made. Time constraints and missing important information are the reality in most real-world studies. Do the best you can, but be sure you are conscious of the limitations and assumptions you make in completing this study. Good luck!

References

- [1] Rickert, M., K. Nagel, M. Schreckenberg and A. Latour (1996). "Two lane traffic simulations using cellular automata." *Physica A: Statistical Mechanics and its Applications* **231**(4): 534-550.