SimRail: An Agent-Based Analysis of the Caltrain Rail Network

CS424M: Computational Modeling in Cognitive and Social Science

Alan Joyce
Stanford University

ABSTRACT

In this paper, we seek to develop an agent-based representation of the Caltrain rail network in the San Francisco Bay Area. By representing the transportation network as a system of interacting agents, we hope to produce a realistic model of the network's behavior from which we can draw insights about its properties and performance.

Furthermore, we establish a model for dynamic modification of train routes based on passenger activity and evaluate this model's performance against that of the real-world Caltrain route schedule. To do this, we define a process for representing a transportation network as a pair of graphs and we enumerate a set of graph analysis procedures that allow for relative comparisons to be made between networks.

Categories and Subject Descriptors

H.1.1 [**Models and Principles**]: Systems and Information Theory – *general systems theory, information theory.*

General Terms

Algorithms, Measurement, Performance, Design, Reliability, Experimentation, Verification.

Keywords

Transportation networks, agent-based modeling, rail networks, graph analysis, network representations, networked agents, system optimization.

1.RELEVANCE AND MOTIVATION

Public transportation networks represent a critical component of local infrastructure in cities across the world, providing mobility to millions of people every day and shaping the urban landscapes around them. At the same time, such transportation systems are the conglomeration of hundreds if not thousands of individual agents working to best serve the needs of a diverse set of independent passengers. The result is a set of interactions that produce complex and often unpredictable network effects that are difficult to model on a global level.

Regardless of scale or composition, all public transportation networks strive to maximize efficiency. The goal is to allow passengers to reach their destinations quickly and easily through the network while maintaining as few routes and vehicles as possible. Unfortunately, the complexity and scope of each system make this optimization difficult to perform at a theoretical level. However, by using agent-based modeling we can represent transportation networks in terms of the thousands of interacting agents that they are actually composed of, allowing for analysis and optimization at both a local and global level.

For the purposes of this paper, we turn our attention specifically to the Caltrain rail network, running from San Francisco, CA to San Jose, CA. This network provides a good resource for exploration, with complexities such as express trains and variable arrival frequencies throughout the day, but it also represents a simple geographical structure in the form of a single, largely straight line running from north to south. Beyond its structure, the Caltrain network presents a compelling cultural study, as the system faced a dire financial situation in 2011, only recently recovering to a more securely funded state [8]. Clearly, deep analysis and optimization have the potential to benefit such a financially unstable system, again making Caltrain an intriguing target for study in this paper.

Despite these motivating factors, the overall trajectory of this paper will not be to develop a specific method for the optimization of rail networks. Instead, we will undertake a process of experimentation with these networks' underlying behaviors, ultimately pursuing a deeper understanding of the mechanisms that contribute to or detract from the performance of a transportation system.

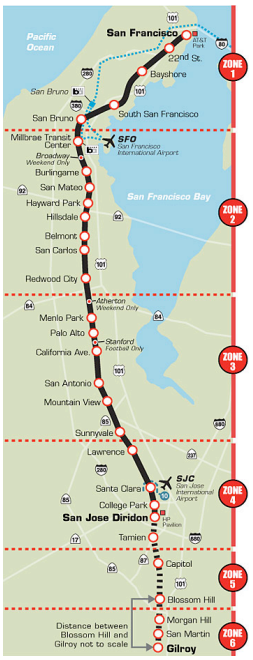


Figure 1. The Caltrain rail network [2]

2.LITERATURE REVIEW

This paper follows upon multiple works which have sought to apply computer-based modeling to the operation of public transportation networks, in addition to a few works that have developed frameworks for analyzing and comparing the properties of these networks.

Of particular interest is the work of Heidergott and De Vries [6], which demonstrates the use of discrete event systems to build a control theory framework for public transit. The authors focus on the specific problem of a train waiting on another train which is delayed in order to accommodate a connection, but in the process of approaching this problem they also define a model for a complete transportation network, even if it is only used to represent a simple example network.

In the work of Arentze and Timmermans [1], the authors develop a learning-based passenger travel model based on consumer choice heuristics. Crucially, the work focuses on representing passenger travel through a network as a bridge between two activities, one at the source of their trip and one at their destination. The combination of this approach with an analysis of the physical and social constraints that govern passenger behavior results in a conceptually sound, if somewhat complex, model of such behavior.

On another front, the work of Sienkiewicz and Holyst [10] concerns the examination of public transportation networks throughout 22 cities in Poland. In both this and work by Seaton and Hackett [9] on the Boston and Vienna subway systems, the authors use universal tools of complex network analysis to make

comparisons between the properties of different transportation networks, as well as predictions regarding those networks' structures and behavior.

Finally, in a previous work by the author of this paper [7], a set of network analysis procedures is defined and integrated with the commonly used General Transit Feed Specification (GTFS) [5] data format to create a semi-automatic procedure for transportation system analysis that is shown to be effective at identifying differences in network properties and overall system robustness.

3.PROPOSED RESEARCH QUESTIONS

By modeling the individual components of the Caltrain rail network as agents, we hope to investigate the following questions:

- Can an agent-based model of the Caltrain rail network accurately reflect the properties and behavior of the real-world system?
- If trains are allowed to use current passenger flow information to define and modify their own routes, will the resulting system behave realistically? If so, will it exhibit a gain in efficiency and/or robustness over the real-world Caltrain?
- What local and global behaviors can we identify as having a positive or negative correlation with the efficient operation of the rail network?

In attempting to answer these questions, we will look closely at specific behaviors of the transportation network as it operates throughout a typical weekday, but we will combine this observation with a study of the broad network properties and system-wide characteristics of the system. It is our hope that the combination of these two modes of analysis will result in a deeper understanding of the Caltrain network and the processes that govern its structure and performance.

4.METHODS AND EXPERIMENTS

The first goal of this project is to develop an agent-based model representative of the real-world Caltrain network. We use the SimPy framework for Python to define a set of agents and give each of those agents a list of operations to perform every time step (see section 5). Time steps are considered to each be a single minute of the day, with the simulation running for 1440 time steps to represent a single weekday.

4.1Data Sources

To both inform and verify this model, we use two data sources: the Caltrain annual ridership data for 2011 [3] and the complete Caltrain route schedule encoded in GTFS format [4]. The ridership data is used to generate a popularity distribution over all stations, with average weekday passenger activity "on" and "off" boarding counts dictating the popularity of all stations both as sources ("on") and destinations ("off"). Additionally, the ridership data informs a peak usage modifier to passenger flow as follows:

time period	**station popularity modifier**
midnight – 6 AM	0.5
6 AM – 8 AM	2.0
8 AM – 4 PM	1.0
4 PM – 6 PM	2.0
6 PM – 10 PM	1.0
10 PM - midnight	0.5

We use the GTFS route data to initialize the 31 stations of the Caltrain network, including their latitude and longitude coordinates, and we will continue to use this data to validate our model. Specifically, we use information about scheduled weekday trips to count the number of connections between stations, train stop events, and train movement events, so that we may later compare these with the behavior of the model Caltrain.

4.2Smart Trains

To address the next goal of this project, we added a degree of semi-autonomy to the trains themselves, giving them the ability to modify their routes in response to passenger activity. The central component of this feature is a payoff equation, I(train, origDest, dest) that evaluates the relative impact of train visiting dest instead of origDest. This equation takes the following form:

$$I(t, s_0, s) = a(P)\,(d_0 / d)^x\,(R)$$

P = # of passengers on the train and waiting at the current station who are traveling in the direction of s
d_0 = distance to s_0
d = distance to s
R = r if the train needs to change directions to travel to s
1 otherwise

In order for a train to modify its route such that it visits dest instead of origDest, the payoff value I must exceed a baseline inertial payoff I_0 for remaining on the regularly scheduled route and visiting origDest as planned. Thus:

$$s_{next} = \text{argmax}_s \{I_0 \mid s = s_0, I(t, s_0, s) \mid s \neq s_0\}$$

We have three constants in the payoff equation: a, x, and r, which balance the relative importances of helping as many passengers as possible, not traveling very far for a single stop, and not reversing directions. Through a process that will be described in section 4.3, we have set these constants as follows: a = 2, x = 3, r = 0.1. We have also set $I_0 = 190$.

4.3Verification and Adjustment

It is important to ensure that the semi-autonomous train model still exhibits broad network characteristics in line with those of the real-world Caltrain, because otherwise a direct comparison between the two would be difficult to make. To do this, we monitored four outputs while adjusting the constants of the payoff equation:

S the number of stations
C the number of direct connections between stations
(*such a connection exists between s_1 and s_2 if a train directly connects s_1 to s_2 at some point in the day*)
E_s the number of stop events
(*times when a train stops at a station*)
E_m the number of move events
(*times when a train moves between two stations*)

With the final constant values from section 4.2, we have the following comparison between real-world and model networks:

Table 1. Real-world and model Caltrain network structure

	real-world	**model**
S	31	31
C	122	119
E_s	2205	2112
E_m	4293	4217

4.4 Graphics

In addition to numerical validation, a graphical depiction of the system helps us both verify and analyze the network's behavior. We generate an overall network graph, as well as a detailed graphical representation that animates throughout the course of a simulated weekday.

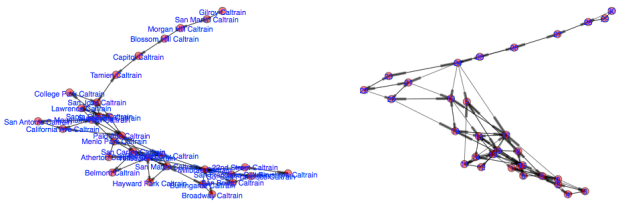


Figure 2. Network graph of the real-world Caltrain system (left) and the model Caltrain system (right)

The overall network graph (see Figure 2) depicts stations as nodes and connections between stations as edges. The correspondence between the real-world Caltrain network graph and the model network graph lends greater support to the validity of our model as a representation of the Caltrain system.

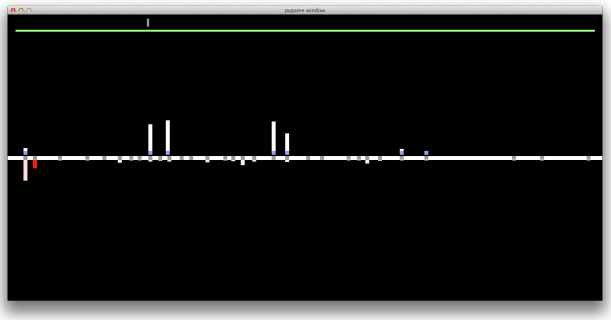


Figure 3. A graphical view of the model Caltrain system.

The animated representation of the system depicts individual trains, their passengers, and the passenger queues at each station throughout the day. The green timeline at the top of the screen is traversed by a gray marker as the day progresses. As this happens, trains (blue squares) move from station to station (gray squares) along the railroad tracks (white horizontal line). The northern terminus (San Francisco) is located on the left side of the screen, while the southern terminus (Gilroy) is on the right side.

Within the graphical display, passengers are represented by vertical bars. When riding a train, passengers are shown as a vertical white bar rising upward from that train. When waiting at a station, passengers are shown as a vertical bar extending below that station, with the bar turning from white to red as the average passenger wait time increases for that station's queue.

The entire window is updated at every time step (every minute of simulation time). This simple graphical representation allows us to monitor the model's progress throughout the simulation's duration, debug any odd behavior, and make specific observations about agent interactions.

5. DESCRIPTION OF MODEL

5.1 Agent Properties

The model we create utilizes three main agent types: trains, stations, and passengers. There is also a fourth agent type, the overseer, which handles graphical display and network analysis operations. These agents have the following properties:

TRAIN

id	a unique identifier
loc	current location (a station or "IN_TRANSIT")
last	the last location visited (a station or "NO_STATION")
speed	average speed
capacity	maximum passengers on board
passengers	a list of passengers on board
route	a list of stations scheduled to be visited
routePos	position of the next stop in the route
dest	current destination (a station or "NO_STATION")
distanceToDest	distance left until dest is reached

STATION

id	a unique identifier
lat	latitude coordinate of the station
lon	longitude coordinate of the station
sourcePop	popularity of the station as a source
destPop	popularity of the station as a destination
passengers	a list of passengers waiting at the station
trains	a list of trains stopped at the station

PASSENGER

id	a unique identifier
loc	current location (a station or "IN_TRANSIT")
train	current train (a train or "NO_TRAIN")
dest	destination for the day's journey (a station)
waitTime	amount of time spent waiting at a station

OVERSEER

time	the current time step in minutes since midnight
trips	a list of all trips (timed lists of train stop events)

5.2 Agent Behavior

When the simulation is running, each agent executes a sequence of operations every time step as follows:

TRAIN

- if the train is currently in transit
  - update distanceToDest
- if the train has arrived at a station
  - update loc and add this train to the station's trains
- if the train is currently at a station
  - check for the next station along the route
  - look through all other stations in the system
    - if a station would be a more productive destination, add it to the route
  - update the train's routePos and dest
  - wait at the current station for passengers to board
  - leave the current station
    - update loc and remove this train from the station's trains

STATION

– do nothing

PASSENGER

– if this is the first minute the passenger is active

– start at a station randomly weighted by sourcePop
– choose a destination randomly weighted by destPop

– if the passenger is currently at a station
– look for any train that will get the passenger closer to their destination
– if a useful train is found
– get on the train

– if the passenger is currently on a train
– if the train has arrived at the passenger's destination
– get off the train and leave the simulation
– if the train has arrived at a non-destination station
– if the train is about to take the passenger farther away from their destination
– get off the train

OVERSEER

– draw the system's current state in a graphical output window
– update the trips list to reflect any new stop events
– if we have reached the end of the day
– compile trips into a GTFS route schedule
– run network analysis routines
– on the real-world Caltrain GTFS data
– on the GTFS data generated from trips

6.DATA

A single run of our model rail network generates an alternate version of a complete weekday route schedule for Caltrain in GTFS format. We use this output, along with the original Caltrain GTFS data, to run a series of robustness evaluation routines (based on work from [7]) across both the real-world and model route schedules. The results of this procedure will aid our analysis of the relative merits of a semi-autonomous system compared to those of a set-schedule system.

At each step in this process, we will be modifying both transit systems in various ways and observing the results. Our observations take the form of two quantitative measurements: *average trip time* and *fraction of passengers stranded*. The former metric indicates how long it takes the average passenger to move through the system from their source to their destination. The latter metric represents the percentage of passengers who are not able to make it to their destinations before the end of the day.

6.1Random Stop Removal

In the first of these robustness tests, a random collection of stop events is removed from the system, with 10% of the total stop events being removed at each stage of the test.

Figure 4. random stop removal (trip time)

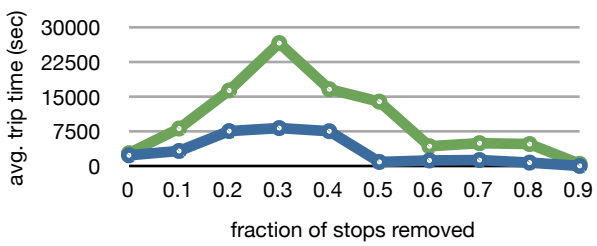
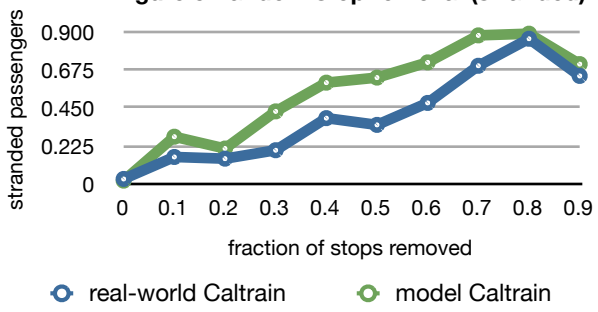


Figure 5. random stop removal (stranded)



6.2Degree-Based Stop Removal

The second robustness test also involves removal of stop events, but instead of the events being selected randomly, they are now selected based on degree, with higher-degree stops being removed first.

Figure 6. degree-based stop removal (trip time)

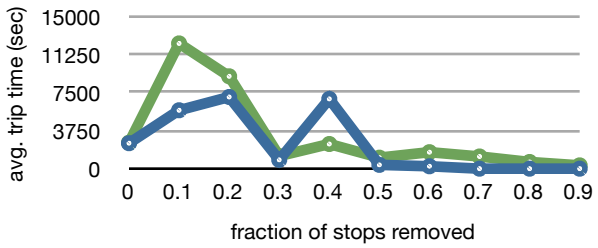
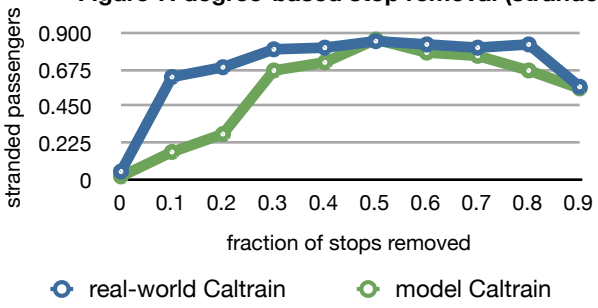


Figure 7. degree-based stop removal (stranded)



6.3Late Arrival Simulation

Finally, we measure each system's response to late arrivals. At each stage of this test, we choose a random 10% of all stop events and make them late by 60 seconds. We do this iteratively up to a total effective lateness of $600(E_s/10)$ seconds, where E_s is the number of stop events, as defined in section 4.3.

Figure 8. late arrival simulation (trip time)

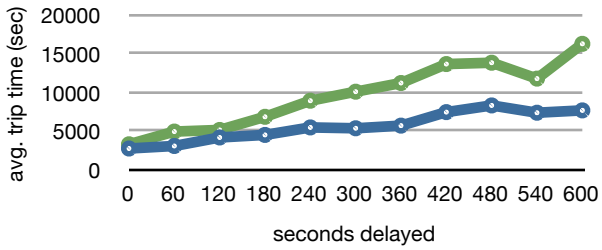
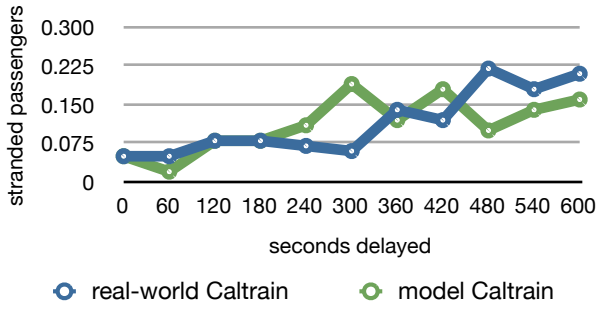


Figure 9. late arrival simulation (stranded)



6.4 Qualitative Observations

In addition to quantitative metrics, we can draw a number of qualitative observations from the behavior of the model throughout a day of simulation.

One observation made early on was that the removal of the direction change penalty R in the payoff equation results in a chaotic system that does not serve passengers well at all. Because passengers board a train with the hopes of riding it in the direction it is currently heading, they are quite ill-served by a set of trains that constantly reverse direction in response to passenger flows. Because the model does not account for train turnaround logistics, the benefit of changing directions is also greatly exaggerated if no penalty is present.

Another observation was the importance of at least some trains making local stops. There exist multiple stations along the Caltrain network, such as Bayshore and San Martin, which represent less than 100 passenger boardings over the course of a weekday. These stations are a frustration to an optimization-hungry train, as they represent a small payoff for the time required to visit them. However, if a train is moving past a local stop anyway, it is important that it can occasionally prioritize distance over raw quantity of passengers, letting it make that local stop in Bayshore on the way to San Francisco and picking up the small contingent of passengers who would otherwise be left stranded and frustrated.

Finally, we have the somewhat counterintuitive observation that two popular stops located very close to each other present a troubling challenge to the model. Indeed, as Figure 10 shows, the second-to-the-right stop (22nd Street) accumulates a queue of unhappy passengers, while the rightmost stop (San Francisco) is served adequately by the system. Both 22nd Street and San Francisco have relatively high popularity among passengers, but San Francisco is the more popular destination. This results in a state where trains heading toward San Francisco do not consider a stop at 22nd Street because San Francisco represents a massive payoff with only a few short minutes more time investment required. As a result, passengers looking to travel north to San Francisco from 22nd street end up waiting in an increasingly long and unhappy queue until the payoff becomes large enough to merit a stop by a northbound train.

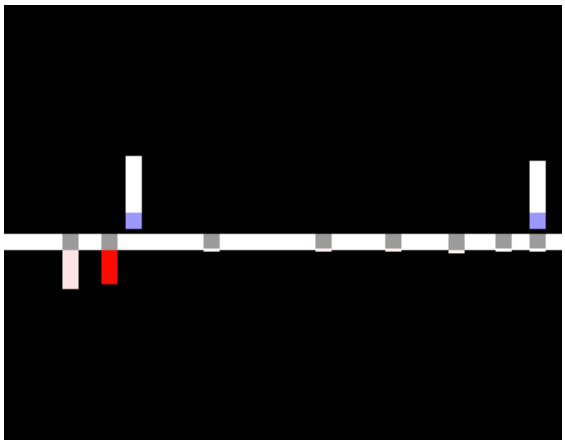


Figure 10. A long, unhappy queue at 22nd Street

7. DISCUSSION

7.1 Representational Accuracy

The first goal of this project is to effectively represent the Caltrain network with an agent-based model. With respect to this goal, we see some degree of success. As seen in Table 1, the network characteristics of the model system and those of the real-world Caltrain system share a great deal of resemblance. By initializing the model using parameters handed down from the real-world ridership data and adjusting the behavior of the payoff equation, we produce a model in which routing decisions are made using a different process than in the real-world, but result in a similarly structured transportation network.

Furthermore, as we consider the system robustness metrics detailed in sections 6.1-6.3, we see that while the model network's response curves often differ in magnitude from those of the real-world network, the two systems' curves often exhibit similar overall trends. For example, in Figure 7 (stranded passengers for degree-based stop removal) we see that both networks reach a peak of stranded passengers at 50% of stops removed, and similar correlations can be seen throughout the other metrics. To confirm that this correlation is indeed due to network similarity, we can see from [7] that the exact robustness metrics used here often produce distinct response curves on different real-world transit networks. Thus, the correlation here appears to represent a notable degree of similarity between the model and its real-world equivalent.

The result of these findings is that we can conclude that the sort of semi-autonomous agent-based representation of the Caltrain network which we have created here does in fact correspond to the real-world structure which it aims to represent.

7.2 Effectiveness of Semi-Autonomy

Beyond simply creating an alternate version of the Caltrain network, we seek to evaluate the performance of this version against that of the real-world network. For this, we turn again to the metrics from sections 6.1-6.3, but with an eye toward the relative successes and failures of each network under stress

simulation. Immediately apparent is an overall trend of the real-world Caltrain network performing more robustly than the constructed one, but we will examine the results more closely.

In random stop removal, the model Caltrain performs significantly more poorly at each point in the simulation than the real-world Caltrain, both in terms of trip time (Figure 4) and stranded passengers (Figure 5). However, in degree-based stop removal, the story is slightly different. The model Caltrain exhibits a similar rapid collapse in trip time as the real-world Caltrain does (Figure 6), with both networks falling into states where almost all trips become very short by the time 30% of stops are removed. On the other hand, the stranded passenger response curves (Figure 7) tell a slightly different story, in which the model Caltrain strands about 60% fewer passengers than the real-world Caltrain does up until the network collapse takes place at 30% stops removed.

It appears that while the model Caltrain is not generally more robust in response to stop removal, it does have the ability to lose its high-degree nodes and continue ferrying passengers to their destinations. This is likely due to the less stringent encoding of specific transfer stations or connections in the model network, lending it more flexibility in the face of losing its higher-degree stops.

Finally, in the last robustness metric, late arrival simulation, the broader trend continues, with the model Caltrain exhibiting a more rapidly increasing average trip time (Figure 8) than the real-world Caltrain, with the gap widening significantly after 240 seconds of delay. The stranded passenger response graph is a bit more muddled, with no certain winner, as both networks claim the advantage for various segments of the simulation.

Overall, we cannot assert that the semi-autonomous version of the Caltrain network produces meaningful gains toward system efficiency or robustness. Rather, this alternative appears to noticeably reduce robustness as measured by several metrics, even if it yields moderate advantages in a few cases. As such, the second goal of this project is not successful, but we believe that further exploration of route definition methods and additional work on the payoff equation could yield significant gains in this area.

8.LIMITATIONS AND CRITIQUE

While the model network presented here has been shown to share many properties with the real-world Caltrain network, and the model has behaved in accordance with expectations under scrutiny during simulations, there are still a number of limitations that must be acknowledged, along with several possible directions for further improvement, refinement, and inquiry.

First, while the model does encode some notion of geography (stations have latitudes and longitudes), there is no directly encoded notion of tracks, which are integral to the design and function of a rail network. Indeed, the simple action of having a train reverse directions or pass another train becomes significantly more complex when paired with the notion of one-way railroad tracks with limited junction points.

On another front, passenger flow remains rather rudimentary in the current model. Passengers behave greedily, taking any train that gets them closer to their destination, but this represents a rather bleak picture of the human travel experience. Borrowing some perspective from the work of Arentze and Timmermans, it would be an interesting exercise to consider a more nuanced passenger travel model, in which the system is simply a means to an end, rather than the entirety of a passenger's life cycle.

From a practical standpoint, larger and broader quantities of data could provide additional avenues for expansion and exploration. As it stands, Caltrain's ridership data records when passengers board and exit trains, but there is little information on when passengers arrive at stations, transfer trains, or make return trips. This information, along with further details on how actual day-to-day train and passenger movement patterns may differ from the scheduled patterns, would all contribute to a useful basis for further investigation.

Finally, this study is limited in scope in that it focuses specifically and exclusively on the example of the Caltrain system. While Caltrain makes for an interesting case study, it is not representative of all transit networks, or even all rail networks. The linear nature of Caltrain makes it an approachable system to visualize and explore, but it also differentiates it from the massive number of non-linear rail and transit systems across the world. As such, it is difficult to determine whether the results of this study have bearing on other transportation networks beyond Caltrain, making this an avenue ripe for future inquiry.

9.REFERENCES

[1] T. Arentze and H. Timmermans. "A learning-based transportation oriented simulation system". *Transportation Research Part B*, 2002.

[2] Caltrain. "System Map". 2012. http://www.caltrain.com/stations/systemmap.html

[3] Caltrain. "February 2011 Caltrain Annual Passenger Counts Key Findings". 2011. http://www.caltrain.com/about/statsandreports/Ridership.html

[4] Caltrain. "Mobile Device Schedules". 2012. http://www.caltrain.com/schedules/Mobile_Device_Schedules.html

[5] Google. "General Transit Feed Specification Reference". 2012. https://developers.google.com/transit/gtfs/reference

[6] B. Heidergott and R. De Vries. "Towards a (Max,+) Control Theory for Public Transportation Networks". *Discrete Event Dynamic Systems: Theory and Applications*, 2001.

[7] A. Joyce. "Experimental Evaluation of Network Properties in Public Transportation Systems". 2011. http://thisisalan.com/other/joyce_transportation.pdf

[8] W. Reisman. "Transportation agencies' funds put Caltrain budget on solid ground". *SF Examiner*, 2011. http://www.sfexaminer.com/local/bay-area/2011/10/transportation-agencies-funds-put-caltrain-budget-solid-ground

[9] K.A. Seaton and L.M. Hackett. "Stations, trains and small-world networks". *Physica A*, 2004.

[10] J. Sienkiewicz and J.A. Hołyst. "Statistical analysis of 22 public transport networks in Poland". *Physical Review E*, 2005.