THE UNIVERSITY OF ARIZONA

Chaos Control in Shuttle Bus Schedules

Matthew Avetian
Lynette Guzman
Ran Kumada
Robert Soimaru
Salim Zerkoune

Math 485

5/11/2010

Introduction

Chaos, it is a term that defines randomness and unpredictability. This feature appears often in everyday activities, crowd movements, inner city driving, even brushing your teeth is a process of random strokes of the toothbrush. There has become an increasing interest in determining what portions of these chaotic systems is predictable or even controllable, in order to make certain processes more efficient. In this report we attempt to reproduce Takashi Nagatani's results on shuttle bus schedules.

The system of the shuttle buses is a rather simple one. There are only two shuttle stops, and two buses. This system may represent an amusement park that has a rather large parking lot. To assist the visitors there is a shuttle system in place to move them from their cars to the park, or vice versa. The program we have developed is meant to simulate this process and watch the headway of the buses with time.

Background

The article "Chaos Control and Schedule of Shuttle Buses" by Takashi Nagatani analyzes two shuttle buses as they pass each other and pick up and drop off passengers. The buses will make up for any time delay caused by dropping off passengers by speeding up, indicated by speedup parameter. There are four different cases of speedup parameter to be considered. We will translate the equations in the article to a MATLAB code which will show the buses have periodic behavior and thus predictability in the shuttle bus system.

In our goal to measure chaos of shuttle buses we must first define what chaos means to better understand the issue at hand. We will use the definition by Strogatz from *Nonlinear Dynamics and Chaos* that states, "Chaos is aperiodic long-term behavior in a deterministic system that exhibits sensitive dependence on initial conditions." (Strogatz, p. 323) A deterministic system is one that has no random or noisy parameters and sensitive dependence allows the nearby trajectories to separate exponentially. Mathematician and physicist Jules Henri Poincaré was the first to discover a chaotic deterministic system which paved the way for chaos theory. However, the invention of the modern computer in the late 1950's led to far more advanced work that Poincaré could not have imagined. Mathematician and meteorologist Edward Lorenz took advantage of the advances in technology to study weather patterns and noticed the dependence on initial conditions. He noticed the solutions to his equations had structure within the chaos and were patterned in the shape of a butterfly, coining the term "butterfly effect." In another famous example, Mathematician Kevin Cuomo used synchronized chaos to encode hidden messages, namely pop music. Cuomo set up an experiment using resistors, capacitors, and amplifiers to mask singer Mariah Carey's song "Emotions" so that outsiders could only hear the chaos or static. However, when the song was sent to the receiver, the output was synchronized almost exactly to the original chaos. After some electronic subtraction, the static disappeared and the song emerged albeit fuzzy (Strogatz, 337). As chaos pertains to our problem, the chaotic motion for shuttle buses depends on both the loading and speedup parameter. Furthermore, chaos is closely related to nonlinear dynamics and our shuttle bus system can be modeled similarly to traffic flow.

Map model

The arrival time, $t_i(m+1)$ $t_i(m+1)$ of bus i at the origin for trip (m+1) is given by,

$$t_i(m+1) = t_i(m) + (\gamma + \eta)B_i(m) + \frac{2L}{V_i(m)} \qquad \text{for i=1,2…,M.}$$

New passengers arrive at rate μ and the new passengers that have arrived since the bus ahead i□ leaves the origin is $\mu(t_i(m) - t_{i'}(m'))$. This gives an equation for the motion of the bus,

$$t_i(m+1) = t_i(m) + \mu(\gamma + \eta)(t_i(m) - t_{i'}(m')) + \frac{2L}{V_0 + s_i\mu(\gamma + \eta)(t_i(m) - t_{i'}(m'))}.$$

Given this equation we want to obtain an equation for dimensionless arrival time. This can be done by dividing the equation by $2L/V_0$ for bus i at the origin. Completing this step we are left with

$$T_i(m+1) = T_i(m) + \Gamma(T_i(m) - T_{i'}(m')) + \frac{1}{1 + S_i(T_i(m) - T_{i'}(m'))},$$

where $T_i(m) \equiv t_i(m)V_0/2L$, $\Gamma \equiv \mu(\gamma + \eta)$, and $S_i \equiv s_i\mu(\gamma + \eta)2L/V_0^{\ 2}$.

Thus, we have created a system controlled by two parameters, Γ the loading parameter and S_i the speedup parameter. This equation states that as the number of perspective passengers increases, the value of the loading parameter becomes high.

The article by Nagatani compares four different cases of speedup parameters. First, both Bus 1 and Bus 2 have a speedup parameter of 0, thus no speedup by either bus. The second case looks at both buses having the same parameter of 0.2 and the last case takes into account two different parameters. In the first case Bus 1 is 0.3 and in the second case Bus 1 is 0.5 while Bus 2 maintains a speedup parameter of 0.2. The goal is to show the chaotic motion of buses is suppressed by the speedup parameter and thus we are able to control the chaos. The first step is to translate the arrival time equation given in the article to work in MATLAB.

MATLAB code

We have two codes to translate the given equation into MATLAB. The first code is the main programming that calculates the headway of the two busses as a function of the loading parameter. Another code is to plot the result and create graphs with four different cases of speedup parameters.

The main code, busMap, calculates each arrival time of bus i at the origin, using the equation

$$T_i(m+1) = T_i(m) + \Gamma(T_i(m) - T_{i'}(m')) + \frac{1}{1 + S_i(T_i(m) - T_{i'}(m'))}.$$

The article we refer to does not give a specific initial condition, and it only plots the arrival time of 901-1000th trip on the graph. Therefore, we choose the equation

$$t(i) = i + \frac{i-1}{i},$$

where i is the bus number, to calculate the initial conditions. One reason is to let bus 1 has a simple initial value, but to let bus 2 has non-integer initial condition with greater value than bus 1. Thus, t =1 and t = 2.5 are initial values for bus 1 and bus 2, respectively. However, we have to be aware that the different initial conditions will results in a different graph shape since the system is chaotic.

The structure of busMap is as follows. Suppose that bus 1 arrives at the origin. Then, the arrival time and trip number is stored in t[] and arrivalTimes[]. Here, t[] stores the bus number and the time at which trip is completed. The arrivalTimes[] stores four different information; 1) arrival time, 2) bus number, 3) trip number for either bus 1 or bus 2, and 4) flag to denote if the trip is already considered. The t[] and arrivalTimes[] are useful when we want to know which bus, i', arrived at the origin just before bus i' arrives at the origin. Next, we calculate the next arrival for this bus, and store the arrival time in tempTimes[]. Then, the simulation advances to the next bus arrival time, which is calculated by looking at the information stored in tempTimes[]. The flag in arrivalTimes is used to determine which previous values to use to calculate the arrival time of the current trip. If a trip is not flagged yet, then that trip is treated as the previous trip with respect to the current trip. Once the information of that trip is used, then it is flagged and thus will not be used again.

This procedure of storing and comparing arrival time is important since the equation involves the arrival time of the previous bus when calculating the arrival time of the current bus. Once the simulation finishes 1000 trips for each bus, it calculates the headway between bus 1 and bus 2. The headway is not defined in the article, but the graphs are plotted as [headway vs. loading parameter]. Therefore, the difference of arrival time of two busses for each trip is used to determine the headway (in absolute value). Thus, the equation for the headway is given by

$$headway = \left| T_1(m) - T_2(m) \right|.$$

The second code, which we call gammaLoop, plots 100 points (from m = 901 – 1000) for each value of the loading parameter Γ. The code is made so that we can choose speedup parameters for each bus. When we run the program, we type the parameters as an array to compare with different speedup parameter (e.g. gammaLoop ([0.2, 0.2])).

Results

We were able to compare our simulation results to the results found in the article by Nagatani (2006). By calculating time headways for various trips of two buses, Nagatani analyzes how both the loading and speedup parameters vary with the time headway. Below are examples of plots with different speedup values of time headway of bus 1 against loading parameter.

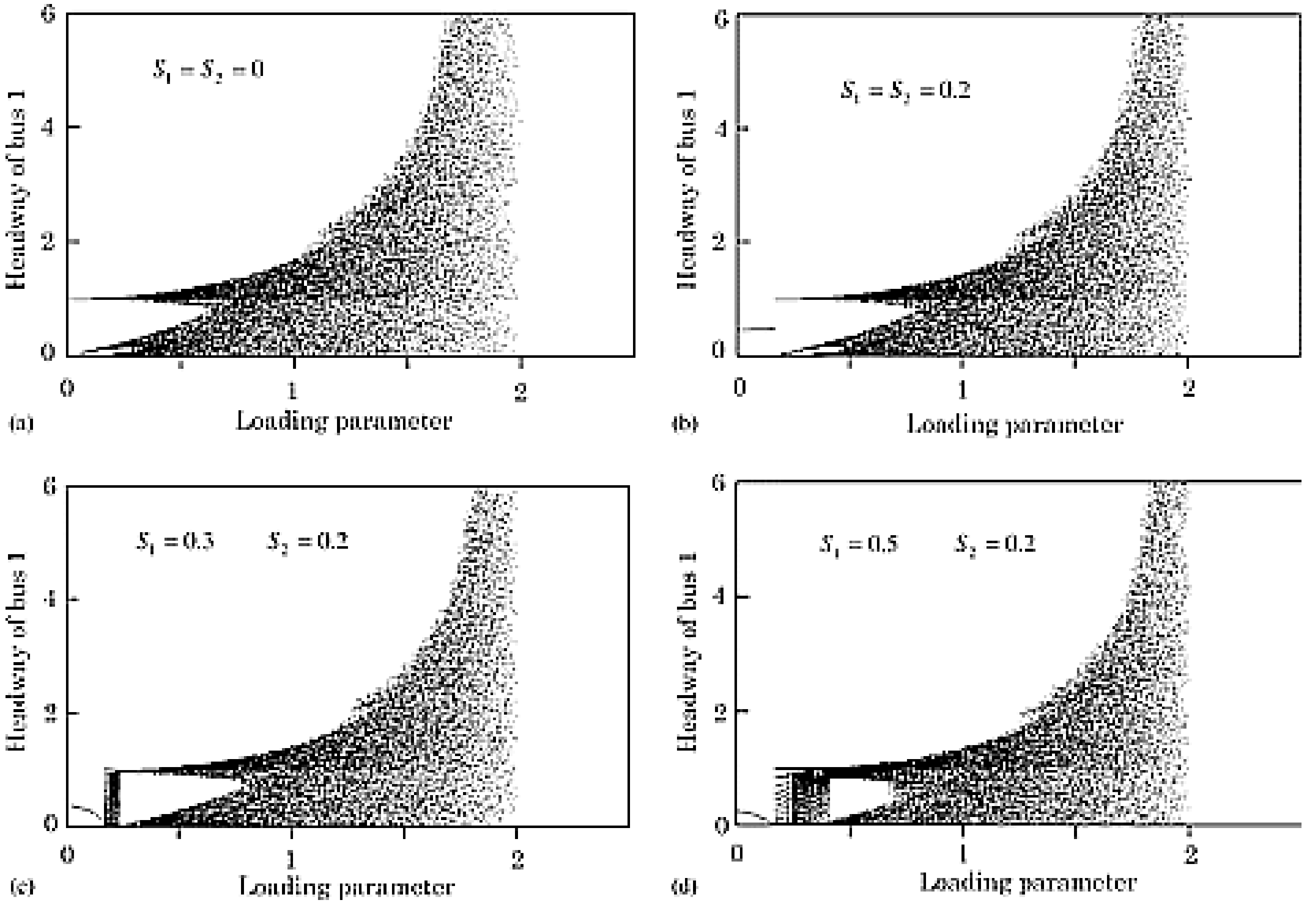
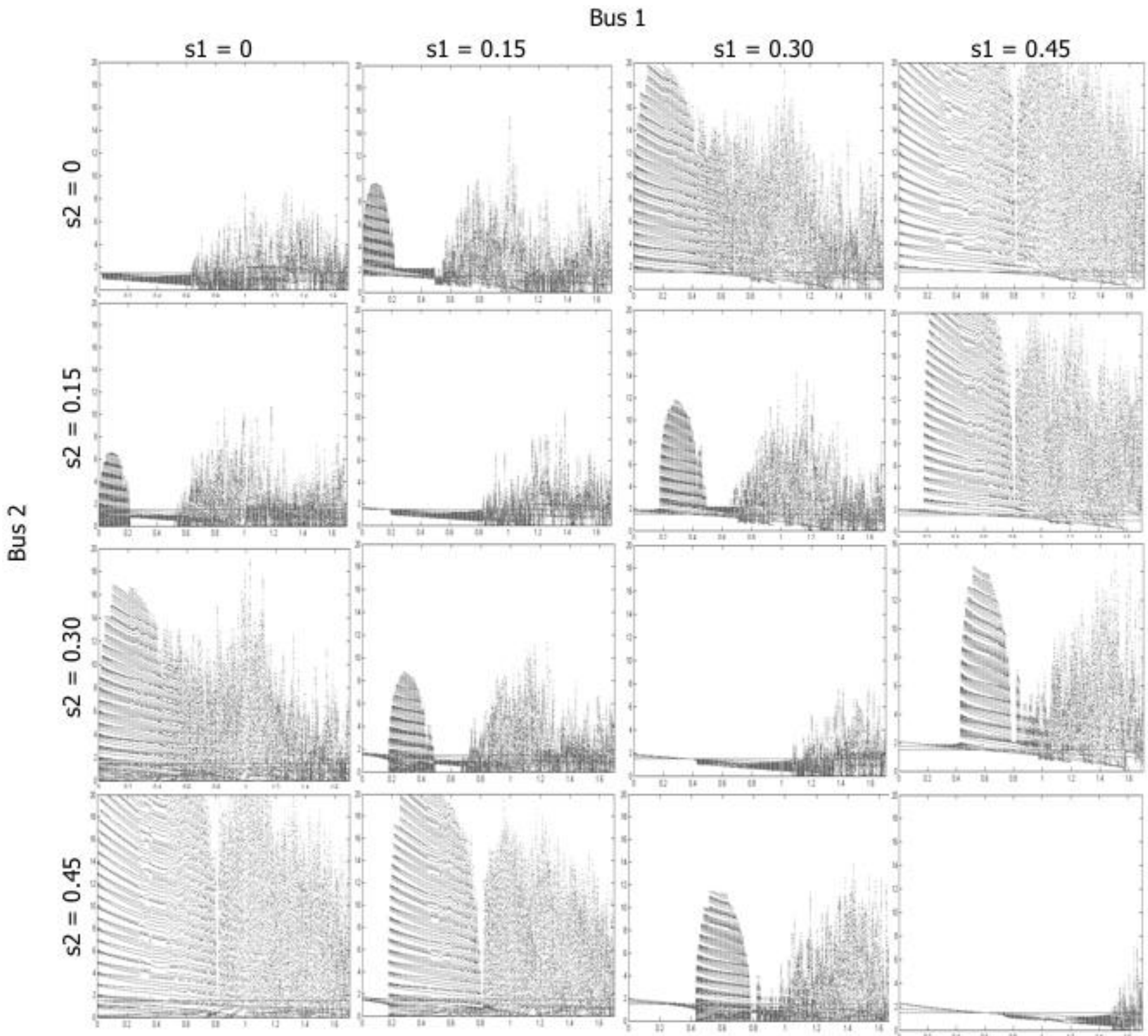


Diagram (a) reflects the distribution of time headway of bus 1 when there is no speedup. Diagram (b) shows time headway distribution for the case when both bus 1 and bus 2 speedup with the same value. Diagrams (c) and (d) both represent a time headway distribution for when bus 1 and bus 2 speedup with different values. In diagram (c), the speedup value for bus 1 is 0.3 while the speedup value for bus 2 is 0.2, and in diagram (d), the speedup value for bus 1 is 0.5 while the speedup value for bus 2 is 0.2.

Nagatani found that when the loading parameter is low, such as in case (a), then the time headway fluctuates around three values and the distribution displays three peaks near these values. As the loading parameter increases, localized distributions extend around the peaks and become two extended distributions. With a high loading parameter, these two extended distributions turn into a single extended distribution. Also, if the loading parameter is greater than 2, then the delay of buses increases and the time headway diverges, creating chaotic motion for loading parameter values between 0 and 2.

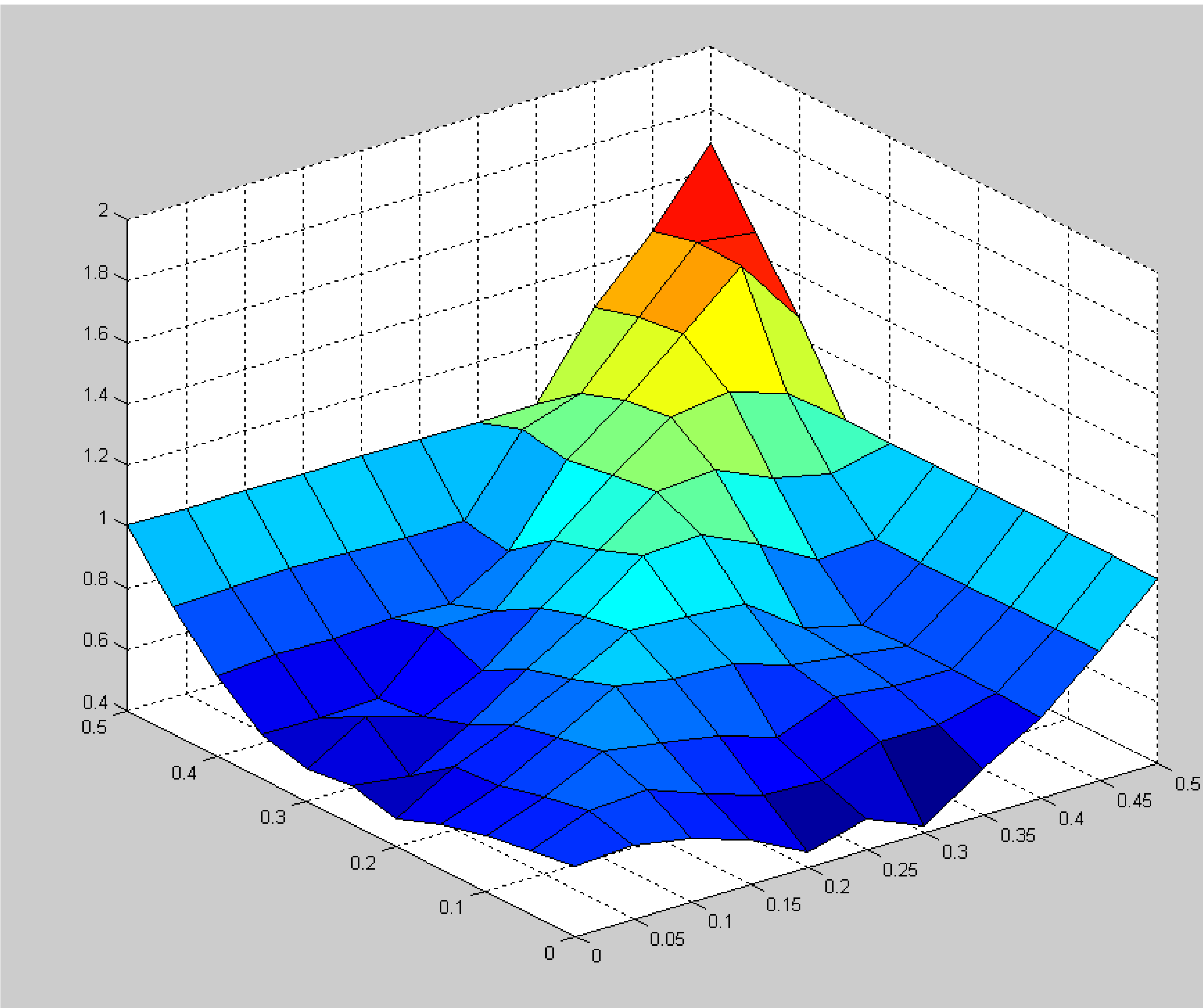
For cases with speedup parameters, the two buses move at a constant speed until the loading parameter reaches the value 0.167, which is when the bus motions start to fluctuate. This motion is then suppressed by the speedup parameter. As seen from figures (c) and (d), the behavior of time headway is not the same for these two graphs. This indicates that the difference between speedup parameters induces the complex motions of buses.

Our simulation result graphs also maintained similar behavior and the same shape as graphs in Nagatani's paper. Generally, the results show a start of periodicity, which is defined as the recurrence of an event at regular intervals, and progressively leads to chaotic behavior.



Above are diagrams from our simulation. The graphs in this grid represent headway of bus 1 vs. the loading parameter. Each row and column is labeled with speedup parameters, S, indicated by bus 1 as the columns and bus 2 as the rows. The chosen speedup parameters are set to 0, 0.15, 0.30, and 0.45, respectively.

Comparing results, our simulation graphs demonstrate periodic behavior and reflect similar shapes as the Nagatani (2006) graphs. This supports the conclusion that speedup regulates chaos. As the loading parameter value (known in our equations as gamma) increases, periodic behavior is then implemented by changing the speedup value. Thus, by using a speedup parameter, this will lead to more orderly traffic by diminishing chaos.



Another interesting component of our findings shows the optimal value of gamma needed for efficiency. Above is a 3-D plot of gamma values that demonstrates the relationship between speedup and the starting point of chaos. The x and y-axis represent speedup parameters of bus 1 and bus 2, respectively, and the z-axis represents the loading parameter value of the point where chaos begins.

These results indicate that by using a speedup parameter, chaos diminishes, thus leading to more orderly traffic. Therefore, the shuttle bus system has some predictability, which may be controlled for better efficiency. Optimal efficiency is achieved when both speedup parameters are equal for bus 1 and bus 2, which is supported in our 3-D model of critical values for the chaotic behavior.

Conclusion

Our results show similar periodic behavior in the same regions as the report showed, even though the graphs themselves were very different. This periodicity means that there is some predictability to the shuttle bus system, and that the buses can be controlled by allowing them the ability to "speed-up" if they fall behind.

References

Nagatani, Takashi. Chaos control and schedule of shuttle buses. Physica A 371 (2006), pg. 683-691.

Nagatani, Takashi. Rep. Prog. Phys. 65 (2002) 1331.

Strogatz, Steven H. Nonlinear Dynamics and Chaos. Cambridge, MA: Perseus Books Publishing, LLC, 2000.

Acknowledgement

We would like to thank our professor, Dr. Ildar Gabitov, for explaining the structure of presentation and guiding us to emphasize setting up the model in an intuitive manner rather than focusing on duplicating the original result. We also appreciate our mentor David Love for his help as we programmed MATLAB code and his advice regarding our presentation and report.