Evacuation Planning of Dante/Wayne Hall

Stephanie Allen, Heather Barcomb, Michael Ramsey

May 9, 2016

1 Introduction

A fair number of resources are invested into plans to evacuate people from buildings such as schools and offices, from transportation networks such as subway and bus lines, and even from locations such as cities or towns. These plans wish to maximize speed of evacuation and minimize panic among the evacuees. The problem can be viewed at the macro and micro levels with the former focused on the larger flows within the system and the later on the evacuees themselves. This paper will examine evacuation from a macro perspective, which can be examined via linear programming algorithms such as shortest path, maximum flow, and discrete time dynamical maximal flow. We coded these methods in Excel in the form of constraints and then used the Solver function to optimize the problem via the Simplex method. Our goal was to use these algorithms to learn more about the best way to evacuate the Dante/Wayne Residence Hall in SUNY Geneseo. In the case of the shortest path, we wanted to find the quickest way for each person to exit the residence hall and, with the maximum flow, we wished to understand the maximum number of people we could evacuate from the building given constraints on flow through the hallways and stairs. Finally, with the discrete time dynamic network flow model, we took time into consideration and wanted to find the maximum number of people we could evacuate given constraints on the hallways and given different amounts of time [3].

2 Literature Review

Before we begin the discussion of our methods and models, we should talk about some previous work with evacuation problems. Early on in evacuation problem literature, Francis developed some theories about evacuation problems, which included the uniformity principle [4, 6]. This principle states that if one can "minimize" the amount of time it takes to get everyone out of a building and if everyone can take any of the possible paths to get out of the building, then it will take the same amount of time to traverse each path because the paths which take longer can be made quicker by moving some people to a different path (This is assuming that the paths do not overlap at all.) [4]. Later, Chalmet, Francis, and Saunders constructed a dynamic model which integrated time into a "static model" by staging the complete set of nodes for each time period and linking all the nodes to the nodes they could traverse to according to the 'travel time' required between each node. The dynamic model

Figure 1: An image of Wayne/Dante Hall.

actually "triply optimizes" because it finds the smallest average time for a certain number of people to get out, finds the maximum number of people that could leave in a certain amount of time, and "minimizes the time period in which the last evacuee exits the building." The article also stated that the dual variables of this model can be used to find "bottlenecks" in the network [1].

Choi, Hamacher, and Tufekci employed a very similar dynamic model structure as the previous paper (and discussed the 'triple optimization' result), but they also created functions for the arc capacities that depended upon the number of people already traveling on the arc and attempted to solve the models with these functions via various methods [2]. Kostreva et. al incorporated time into their fire evacuation models, but they did also account for the case when a location (or 'node') cannot be traversed after a certain point in time (because of the fire) by including a larger 'cost' (similar to the Big M method) on arcs that go into the node after a certain point in time. Furthermore, the researchers dealt with 'multiple objective dynamic programming' where by each arc had multiple characteristics such as length and amount of smoke; the problem was solved by looking at the paths from each room to the exits that were not "dominated" by other paths - meaning the path chosen was not definitively worse than the other possible paths when looking at the multiple characteristics together [5].

3 Methods and Data Collection

We focus on the evacuation of the Wayne/Dante Residence Hall on SUNY Geneseos campus. This residence hall is home to approximately 100 freshmen each year who are ordinarily placed two to a room. We visited Dante to better understand the floor plans we found online and took measurements (in feet) of the hallways, stairs, and other distances to provide us with realistic data upon which to base our models [7].

We formulate the algorithms with some assumptions and simplifications. First, we assume that the evacuation situation modeled takes place at night such that everyone would be in his/her respective rooms meaning no one would be in the lounges, the kitchen, or the laundry room. We also disregard one set of stairs from our model in order to simplify the calculations. Furthermore, we use the measurements we obtained from the visit to Dante along with the floor plans (which we assume to be to scale) to estimate the other distances we need for the models. Finally, we do not include any extra constraints to the model for routes that overlapped each other.

We treat each room as a node, each corner of the rectangular hallway structure as a node (four per floor, four floors total), the entrance to the stairs on each floor as a node (one per floor), and the outside of each exit as a node (three total). Each room is connected to one or two hallway nodes based on the edge of the rectangule on which it is located, and the two hallway nodes located on the edge containing the stairway entrance are then connected to the stairway entrance node.

The arcs in the network are directed. The arcs from each room go to the hallway nodes in this one direction, but we place two arcs between each set of hallway nodes to make sure people can move in both directions. We also allow people to move up and down the stairs and, therefore, have two arcs between those nodes.

3.1 Shortest Path Model

We construct the shortest path problem using our initial distance calculations. We want to find the shortest path that a person leaving each of the rooms could take to leave the building. We identify each of the arcs and code them into MS Excel along with their distance measurements and add constraints such that each of the nodes will have a ‘net flow of zero. Then, for each individual room, we ‘constrain the net flow to 1, the net flow of the final Sink node (which was fed by the exit nodes) to -1, and run Solver to minimize the value of the decision variables given these constraints. We calculate the length of each path by multiplying each of the decision variables (arcs) that is equal to one by its corresponding distance; in Excel, we write this as the formula to be minimized. A representation of this problem in constraint form is written here, with the decision variables (Arc_{ij}) as the arcs and the subscripts representing the nodes (i is the node a person is coming from and the j is the node to which a person is going). As an example, we will show the set up when we are finding the shortest path for Room 205:

$$Min : Z = \sum \sum Arc_{i,j} * Distance_{i,j}$$

$$Subjectto : FlowOutofRoom205 - FlowIntoRoom205 = 1$$

$$FlowOutofAnyOtherNode_n - FlowIntoAnyOtherNode_n = 0$$

$$Arc_{i,j} = binary$$

The output of the Solver function gives us the total distance of the shortest path plus the route for each path based on if each of the decision variables are equal to one (which indicated that that arc was used).

3.2 Static Network Flow Model

Since the shortest path formulation does not account for multiple people being in the same hallway at once, we are also modeling this problem as a maximum flow problem. We use the same distances from the shortest path formulation and multiply them by $\frac{2}{5}$ in order to decide how many people could fit into the hallway at one time. If this capacity is a non-integer, we round it down to the closest integer value. The reason we decided on $\frac{2}{5}$ as our multiplier is based on the width of the hallways and other measurements. We determined that two people could fit side-by-side and each row of people would take up approximately five feet if we give them space on either side. For example, a 32.5 foot hallway could fit about 12 people at one time. From here, we can create the problem in Excel and use Excel Solver to maximize the flow, which represents how many people can get out of the building. We are also setting Solver to only find the maximum flow with integer values. Essentially, all that this maximum flow value tells us is how many people could evacuate the building without having to stop to wait for hallways to clear up. However this algorithm does not consider time, so once the capacity of the hallway has passed through it, no one else could use the hallway even though the people who had passed through previously would be gone. For example, consider the situation where there is a bomb or booby trap that goes off once a certain number of people have passed through a hallway, rendering that hallway useless to anyone else. That is what is happening if we formulate the evacuation as a maximum flow problem. The constraints we are using are as follows.

$$\text{Max}: \; Z = \sum \text{Flows Originating from the Source}$$

$$\text{Subject To}: \; \text{Flow}_{i,j} \leq \text{Max Flow}_{i,j}$$
$$\text{Flow In} - \text{Flow Out} = 0 \text{ for each of } n-2 \text{ transhipment nodes}$$

$$\text{and Flow}_{i,j} \geq 0, \text{ Flow}_{i,j} \in \mathbb{Z} \text{ for } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, n$$

We can modify our formulation in an attempt to account for time in a very simple way by assuming that the floors move in waves. We reason that by the time the fourth floor residents reach the third floor, the residents of the third floor would already be on the second floor, and so on. So to account for this, we multiply the maximum flow of relevant third floor arcs by two and multiply the maximum flow of relevant second floor and first floor arcs by three, since up to three "waves" of floors could pass through those hallways and staircases.

Obviously, this is a very naive way to implement time into our model as we are not controlling if the secondary waves of people are actually the evacuees taking advantage of the extra flow we inserted into the model. It also only deals with the number of residents who we were able to evacuate and not how quickly they were evacuated. For this reason, we also consider a dynamic model that takes time and hallway capacities under consideration simultaneously.

3.3 Discrete Time Dynamic Network Flow Model

In the previous section, we discussed the difficulty with incorporating time into the static network flow model. Time dependent conditions are difficult to express with the latter model, which is why we decided to implement the discrete time dynamic network flow model. With the dynamic model, we still use the same nodes as in the static problem, however multiple copies are created to account for the time dependency. An example of a static model and its corresponding dynamic model is presented below [3].

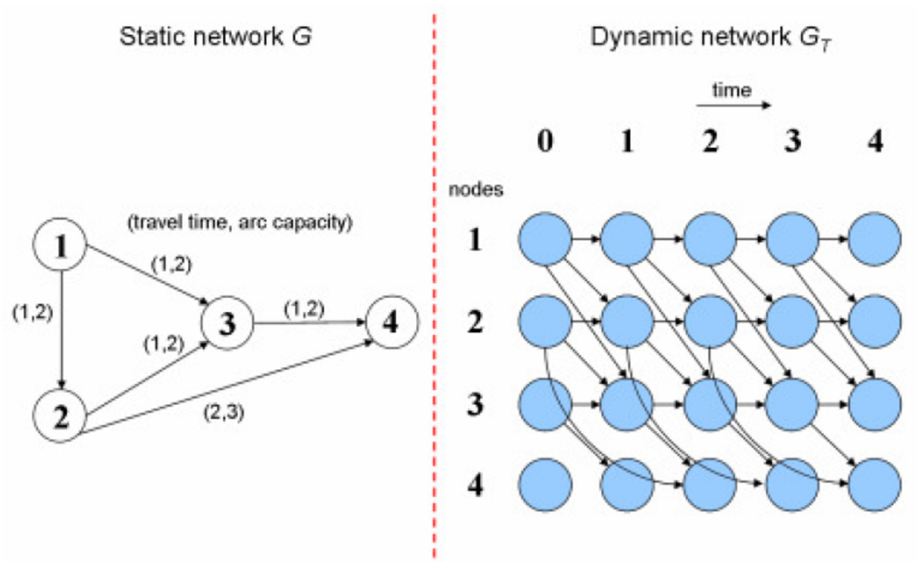

Figure 2: An example of a Static Network and its corresponding Dynamic Network.

As you can see, we can represent the dynamic model as a static one using discrete time steps of one second. By formulating the problem in this way, we can find the optimal flow by using the augmenting path algorithm or the simplex method. The new objective is to maximize the number of people that can evacuate the building in a given time horizon. For example we can pose the question, “How many people can evacuate Wayne/Dante Hall in 20 seconds?” The associated dual problem is to minimize the time needed to evacuate Wayne/Dante Hall [3].

We must set some base assumptions for this evacuation problem. First we assume that every individual is evacuating the building in a calm matter with a walking speed of four feet per second. We use this average speed and the distance information that we found in the shortest path problem to estimate the time it takes for the person to traverse between nodes. Once the allotted time has been satisfied, we format the model so that the people are allowed to traverse the arc to reach the next node. We also set constraints for the hallways. We assume that only two or three people can go through a hallway at a time. This number may seem small in practicality, however the small constraints make the problem more interesting by allowing temporary backups in flow. In the real world, this backup in flow could be caused from debris or other types of danger. In this problem, we must also allow the possibility for people to stay in place. In the model, this corresponds to moving to the same node at the next time steps. Since we do not want to prevent people from staying where they are, we set this constraint at a high integer number so that we do not restrict flow.

Since the incorporation of time steps causes the model to expand, we only formulate the evacuation of Wayne/Dante Hall's second floor. If we try to evacuate all four floors, the problem becomes too large for the Excel Solver to handle.

4 Results

4.1 Shortest Path Model

Using the Excel Solver to get results from our shortest path model, we find that the distance is generally decreasing for lower floors. For example, the average shortest route out of the building for the fourth floor is 122.26 feet. The third floor's average is 115.81 feet, the second floor's is 76.304 feet, and the first floor's is 45.95. This is what we were expecting since there are only three exits in the building: one on the second floor and two on the first.

Overall, the shortest minimum path is from Room 108 and was only 27.05 feet. The longest minimum path is from Room 406 and was 162.75 feet.The overall average path distance is 100.74 feet.

Below is the complete set of data we obtained by using Excel Solver to find the shortest path out of the building from each room:

Table 1: Shortest Path From Each Room

Room	Distance (ft)	Room	Distance (ft)	Room	Distance (ft)	Room	Distance (ft)
403	130.88	417	95.58	312	121.65	210	102
404	138.68	418	102.08	313	115.75	211	108
405	154.48	419	88.58	314	98.05	212	84.5
406	151.75	420	127.08	315	92.05	213	78.25
407	162.75	303	95.38	317	99.08	214	60.55
408	136.95	304	103.18	318	99.08	215	54.65
409	128.25	305	118.98	319	92.25	217	35.75
410	121.25	306	130.83	320	91.58	108	27.05
411	127.25	307	141.83	205	55.65	109	35.75
412	117.98	308	137.63	206	67.5	110	54.75
413	112.08	309	146.33	207	111	111	60.75
414	94.38	310	139.5	208	74.8	112	51.65
415	88.48	311	145.5	209	83	RD	45.75

Additionally, the following images are visual representations of the shortest path from each room to an exit found by Solver. We find that the only people who would exit through the two exits on the first floor are residents of the first floor, while everyone else exits through the second floor. Since this formulation of our problem only takes distance into account, this is what we expect.

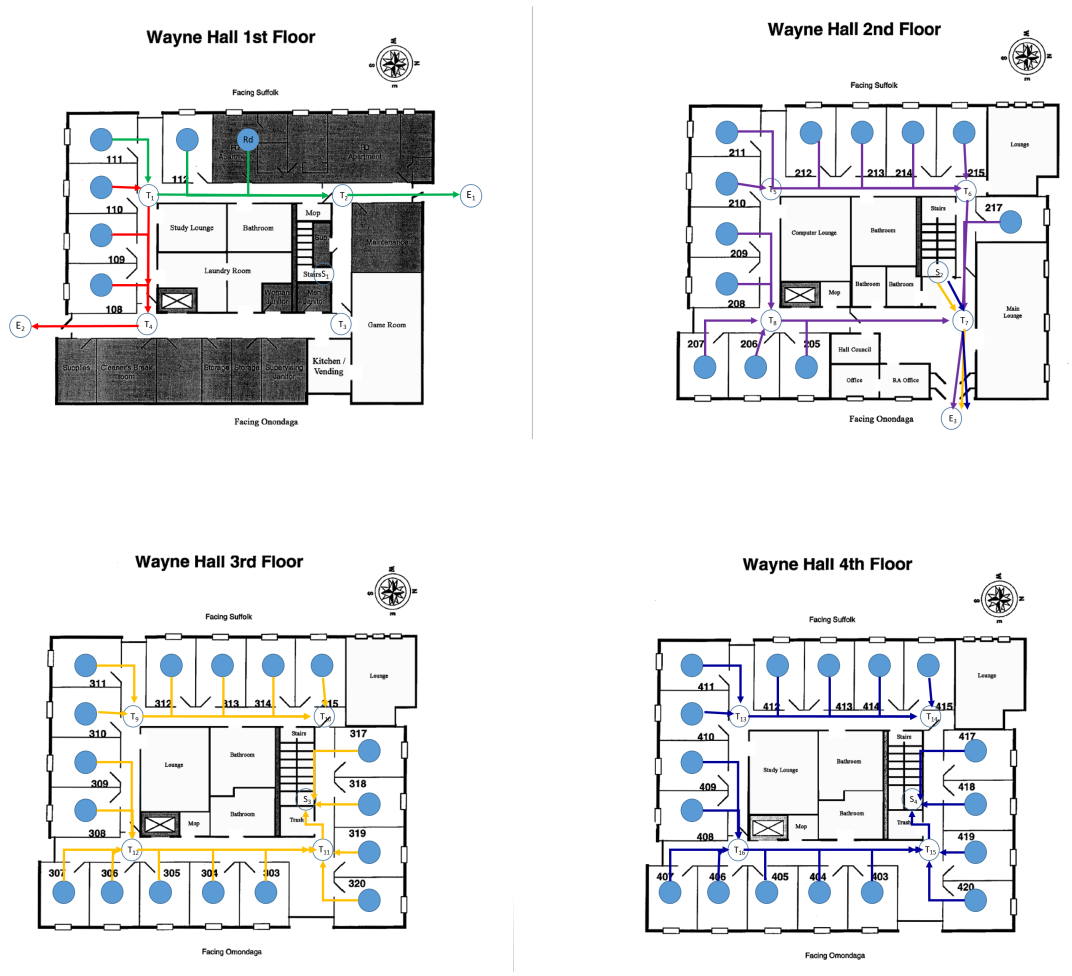

Figure 3: The optimal evacuation plan based on the shortest path.

4.2 Static Network Flow Model

Using Excel Solver, we initially find that 12 people were able to evacuate the building. This was out of a total of 104 residents. This is what we expect since, as we discussed above, formulating the evacuation process as a static maximum flow problem in this way is like assuming that once a certain number of people go through a hallway or section of a hallway, it collapses and makes it unusable to anyone else.

By modifying our model in an attempt to account for time, we were able to evacuate 32 people from the building. Although this is substantially better than the previous attempt, it is obviously much lower than the total number of people in the building, but this is also understandable considering that this number would b representative of the number of people who are getting out of the building without having to stop walking at some point due to backed-up crowds. This model does not have a way for people to stop walking and stay at the same location like the dynamic model does.

4.3 Discrete Time Dynamic Network Flow Model

We formulate the Discrete Time Dynamic Flow Model in Excel using time steps of 10, 15, and 20 seconds. We attempted to include even more time steps, however the model became too large for the Microsoft Excel Solver to handle. As you can see from figure 2, the incorporation of time steps causes the flow model to explode in both the number of arcs and nodes. Therefore to continue with more time steps, we need improved software to solve the problem.

We summarize the results of the Discrete Time Dynamic Network Flow Model below.

Table 2: Results of the Discrete Time Dynamic Network Flow Model

Number of Time Steps (s)	Number of People Evacuated	Rooms Evacuated
10	2	217
15	8	205, 214, 215, 217
20	16	205, 206, 207, 208 213, 214, 215, 217

As you can see, the second floor of Dante hall cannot be evacuated in 20 seconds. Only 16 people are able to leave the building while 8 are still inside. We suspect that if we were able to extend this model to 25 seconds, everyone would be able to exit the building from the second floor. Further information is available on the path that is taken to evacuate and the time a room is evacuated by taking a closer look at the results of the Excel Solver.

5 Further Research

The minimum path and the static flow models are too naive to be worthy of future research. However with the discrete time dynamic maximum flow model, there are many directions that we can investigate. First, we would extend the problem so that we could evacuate the entire building of Wayne/Dante Hall. Second, we can explore what happens if we prevent flow in certain paths. For example, suppose there is a fire and one of the hallways collapses after 10 seconds of burning. We can incorporate this change into our dynamic model by preventing flow through this path after 10 seconds. This can provide us with a “what-if” analysis of many possible evacuation situations. Third, as stated earlier, we can look into the dual problem to find out where bottlenecks occur in our flow system, and adjust our evacuation criteria as needed.

References

[1] Chalmet, L. G., R. L. Francis and P. B. Saunders. “Network Models for Building Evacuation.” *Management Science: INFORMS* 28, no. 1 (1982): 86-105.

[2] Choi, W., H. W. Hamacher, and S. Tufekci. “Modeling of building evacuation problems by network flows with side constraints.” *European Journal of Operations Research* 35 (1988): 98-110.

[3] ”Evacuation planning.” *24 Hours Operations Research*. 2016. `http://www.24hor.org/index.php?option=com_content&task=view&id=2&Itemid=47`

[4] Francis, Richard L. “A ‘Uniformity Principle’ for Evacuation Route Allocation.” *Journal of Research of the National Bureau of Standards* 86, no. 5 (1981): 509-513.

[5] Kostreva, Michael M., Malgorzata M. Wiecek, and Teodros Getachew. “Optimization Models in Fire Egress.” *Fire Safety Science* 3 (1991): 805-814.

[6] Pursals, Salvador C. and Federico G. Garzon. “Optimal building evacuation time considering evacuation routes.” *European Journal of Operational Research* 192, (2009): 692-699.

[7] “Wayne Hall (Dante House).” *Geneseo: The State University of New York*, April 28, 2016. `http://www.geneseo.edu/residence_life/dante-housewayne-hall%20`