New Wayfinding Techniques in Pathfinder and Supporting Research

Charles Thornton, Richard O'Konski, Bryan Klein, Brian Hardeman, Daniel Swenson

Thunderhead Engineering, 403 Poyntz Ave STE B, Manhattan, KS, 66503, USA
{thornton, okonski, klein, swenson}@thunderheadeng.com

Abstract. This paper presents the wayfinding and door selection algorithm used by the Pathfinder egress simulator. The development of Pathfinder's Locally Quickest algorithm will be discussed along with unforseen consequences and one promising, but discarded, initial attempt. Validation of this new model is presented as well as characteristics of the third party research that make simulator validation possible.

Keywords: pathfinder, evacuation, simulation

1 Introduction

Pathfinder is a commercial agent-based emergency egress simulator. The software includes a user interface, simulator, and 3D visualization system. The wayfinding method discussed in this paper is an aspect of the simulator.

Simulation agents, or occupants, moving from their starting locations to an exit must choose a route to follow when walking toward their chosen exit. This route selection process (wayfinding) significantly affects the overall simulation results, as time spent waiting in queues and time spent walking control the time it takes all agents to reach their objectives.

Previously, Pathfinder used a simple procedure to perform this wayfinding task. Using the A Star (A*) search algorithm, Pathfinder would calculate, for each agent, the shortest possible path the agent could take to exit the model. Because this approach did not account for queue formation at doorways, bottlenecks along the selected route, and equivalent alternative routes, it was necessary to use a variety of workarounds to encourage agents to walk a few extra steps to avoid large queue times. An illustration of this problem is shown in **Fig. 1**.

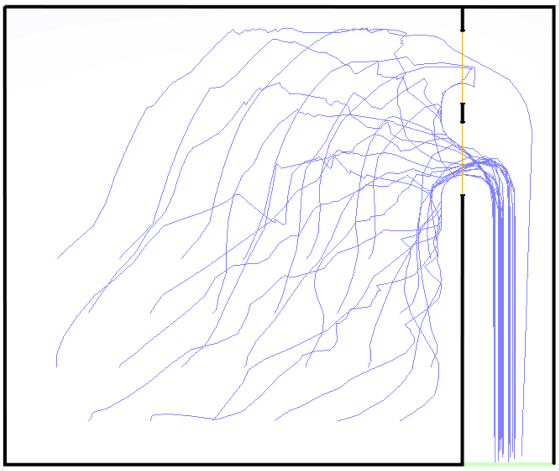

Fig. 1. Shortest-path-only agents fail to utilize the full width of divided doors.

Figure 1 shows a case where simulated agents attempted to exit a room using a divided door. Initially, the agents were positioned in the lower portion of the room. In this case, only one agent utilized the upper portion of the divided door. This occurred because the shortest path-based wayfinding algorithm relied only on the distance between the agents' initial starting position and the nearest intended exit. The calculation did not take into account the impact congestion would have on total travel time.

A similar problem occurred any time agents needed to pass through a bottleneck on the way to multiple exits (e.g. a hallway that led to an entry area that contained a bank of exit doors). In this situation, the path planning algorithm performed the same calculations for each agent after reaching a common point. This led to situations where all of the agents would pursue only the nearest of many available doors.

2 Locally Quickest

The Locally Quickest algorithm is designed to overcome the problems mentioned in the previous section. Instead of calculating a complete path to the exit at the beginning of the simulation, the Locally Quickest algorithm selects a door leading out of the current room. This process is repeated each time the agent enters a new room, until the agent has exited the simulation.

When evaluating routes leading out of the current room, agents calculate a time estimate for each route (i.e. door). This estimate takes into account the following parameters:

- an estimate for how long it will take to reach and pass through a local door, and
- an estimate for how long it will take to reach a valid exit after arriving at the local door.

When selecting a local door, agents compare the time to traverse the shortest path to the door with the time it would take to move through the queue that has formed at the door. The time required to enter the room beyond the door is assumed to be the larger of these two values.

To calculate the time to reach an exit beyond the local door, Pathfinder performs a distance calculation based on A* path search. Agents use the results of the distance calculation with their maximum velocities to estimate a time. This calculation assumes that agents have no knowledge of queues in other rooms, so the results represent the fastest possible time beyond the agent’s current room.

After estimating the local travel time and the time beyond the current room, agents select the door that provides the fastest overall time to their goal. In cases where multiple local doors appear to provide very similar travel times, agents will select a path through the nearest of the equivalent doors.

2.1 Local Movement Time

The local movement time calculation is important in ensuring that agents intelligently utilize all available egress doors within the current room. This element of the algorithm provides agents with a way to decide to use a longer route if the shortest available route is congested.

Calculating local movement time in an empty room is a straightforward process. The distance to the next doorway can be determined using A* search and the agent's velocity can be used to derive movement time from the distance. However, the presence of other agents makes this process much more challenging. It is necessary for the agent to estimate how long it will take to reach the door when waiting in a queue.

An early approach operated on the assumption that congestion would exist primarily around the doors (e.g. in areas where agents were waiting to pass through the door) and that a fine-grained density calculation[1] could indicate areas of congestion. An example of this approach is shown in Fig. 2 and Fig. 3.

Fig. 2. A Pathfinder simulation where a large group turns a corner, then exits through a series of turnstiles.

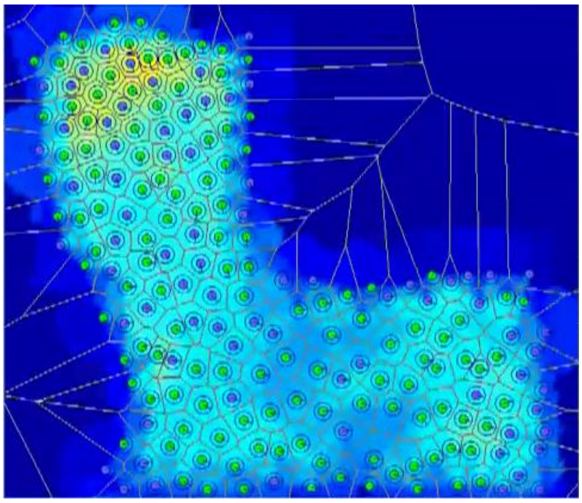

Fig. 3. A visualization of the fine-grained, Voronoi diagram-based density calculation used to determine density in the simulation shown in **Fig. 2**.

The model shown in Fig. 2 is an L-shaped room with twelve exit doors. The exit doors are structured like turnstiles with 0.5 meter deep divisions. All agents were initially packed into the upper-left portion of the model, making the top exit door the closest for most agents. Fig. 3 is a visualization of the Voronoi-diagram-based, fine-grained density calculation as the agents began to reach the exit doors.

Using the Locally Quickest approach implemented with the fine-grained density information produced better doorway utilization than the original shortest path algorithm, but it had serious shortcomings. Because the algorithm integrated density over a straight line path, estimates for a door on the far side of another door's queue would receive time estimates that combined multiple doors. Also, the density calculation was not computationally fast enough to scale up to stadium-sized models.

2.2 Door Queue Approximation

Because of difficulties with a density estimation-based approach, an approach to calculate local travel time based on door queue estimates was developed. To estimate the wait time to travel to a door, agents use the maximum of two values:

- a simple estimate based on minimum travel distance and agent velocity, and
- an estimate of time that will be spent waiting in that door's queue.

To estimate the time spent waiting in a door's queue, Pathfinder needed a way to estimate queue size. This is complicated by the different time at which door selections are made by each agent. It was necessary for agents to select a door before the queues were well-formed (i.e. when other agents were still moving closer to their selected doors).

The selected approach was to store a record of which agents were using each door within each room. Then, when an agent needs to measure queue size for a door, the value can be determined by counting the agents that are waiting on the door and are also closer to the door than the agent.

Once this count is determined, Pathfinder can then calculate the wait time at the door using maximum specific flow estimates from the SFPE Handbook[2]. The greater of this calculation and the pure movement time is then used as the local time estimate for reaching the door.

This calculation is repeated periodically for each agent to account for evolving door decisions made by the other agents.

3 Backtrack Prevention

Agents are only aware of queue sizes in their current room. When they enter a new room, knowledge of the last room is replaced by knowledge of the current room. Without any sort of backtrack prevention in place, large queues could lead to agents moving back and forth between two rooms until the queue situation changed. To prevent these oscillations, the Locally Quickest algorithm uses the following rules when selecting the list of doors agents are allowed to evaluate when considering an exit plan:

- Available local doors must have a shorter path to the agent's destination than the door through which the agent entered the current room.
- Local doors may not lead back into the room the agent previously occupied.

- If the previous criteria eliminate all available doors (e.g. the agent has been pushed into a closet) or the agent has been pushed through an unintended door, backtrack prevention is disabled. All local doors are available.

4 Downstream Queues

Another issue that occurs with certain floor plans is failure to apply downstream queue knowledge. Previously, the time from a local door to the destination has been discussed as an optimized value. However, because the algorithm places so much emphasis on the current room, situations where the current room is also a subsequent room require that knowledge of the current room be applied to time estimates to any segments of the route after the local door that pass through the current room.

Fig. 4 shows a room that contains a smaller room with two doors. The agent denoted by the letter "A" must choose between three available doors. Two doors belong to the small room contained in the upper left quadrant of the larger room. The third door is the goal door at the right.

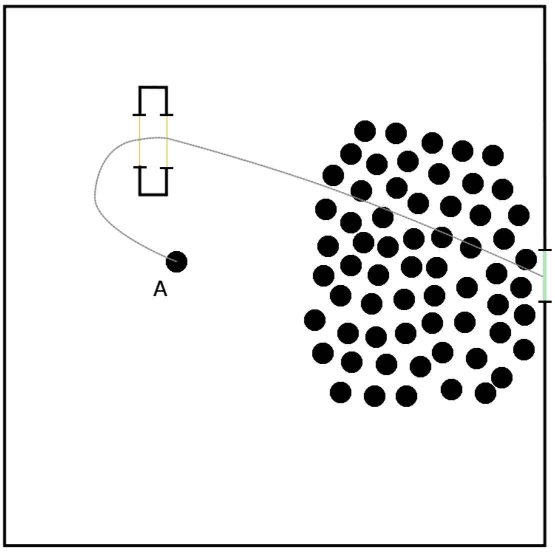

Fig. 4. Two floor plans that require downstream queue awareness to function properly.

The current description of the locally quickest algorithm suggests that, when evaluating the time beyond the doors of the small room a minimal, distance-based time calculation be used. However, in this case, that would cause the agent to walk through the tiny room, believing that upon emerging the large queue at the right will have dissipated. If you initialized the simulation in this configuration, many agents would peel off of the back of the queue and instead line up to go through the tiny room before emerging and lining up again behind the "real" door.

Pathfinder solves this problem by storing the optimal route calculated beyond each local door. If the route passes through the agent's current room, the agent replaces the time estimation for that segment of the route with a more accurate version based on the agent's knowledge of the current room. The improved estimate is made in exactly the same way as the initial door queue time approximation, by comparing the time it takes to travel the segment distance to the time spent waiting in queues and keeping the maximum value.

5 Door Selection Case Study

To test the impact of the Locally Quickest movement algorithm on agent door selection, a door selection case study from VTT Research Notes 2562[3] was used. In this study, university students were monitored as they moved from a staging area, through a hallway and into a lecture hall. The lecture hall had two entrance doors arranged consecutively along the same wall.

Pathfinder was compared against a general observation of the researchers who performed the test, "...people on the right (relative to the direction of movement) on the starting grid chose [the second door]. People on the left of the grid or in the center of the front rows chose the first door: door 1. In the back row, there was a tendency to use door 2." Fig. 5 shows the floor plan of the case study.

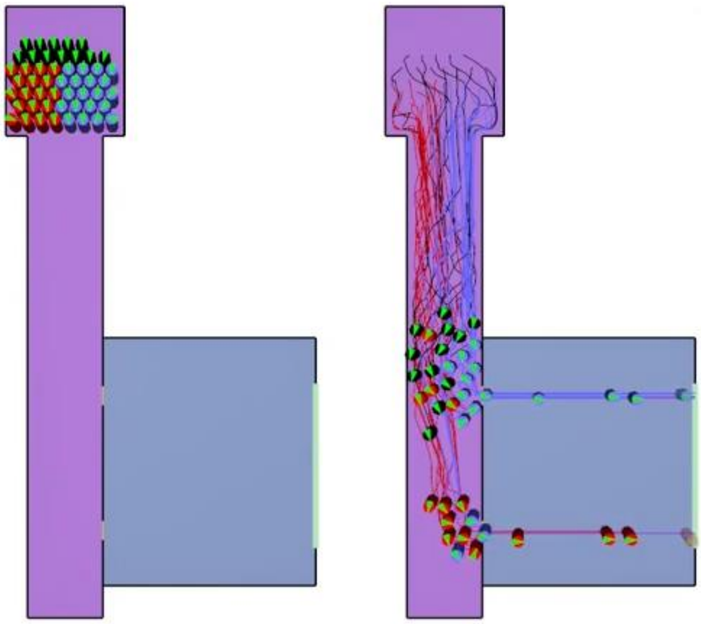

Fig. 5. Selected images of a pathfinder simulation of the case study.

The floor plan in Fig. 5 shows the starting locations and movement paths of a Pathfinder simulation model for this test case. The results of the simulation are shown in Fig. 6.

Pathfinder Exit Usage		
	Top	Bottom
Left (blue)	16	4
Right (red)	5	13
Back (black)	5	7

Fig. 6. Results of door selection case study.

In Fig. 6 the first door is indicated as "Top" and the second door is indicated as "Bottom". People to the right, left, and back are indicated as "Right (red)", "Left (blue)", and "Back (black)".

These results indicate that the results in Pathfinder agree with the general observations given by Rinne, Tillander, and Grönberg. The success of this case study represents a significant step forward for Pathfinder which previously would have given significant bias to the first door in all cases.

6 References

1. Steffen, B. and Seyfried, A. (2010) Methods for measuring pedestrian density, flow, speed and direction with minimal scatter. Physica A 389 (2010) 9, 1902-1910
2. Nelson, H. E. and Mowrer, F. W.: Emergency Movement. In: P. J. Denno & W. D. Walton (Eds.), The SFPE Handbook of Fire Protection Engineering (Third ed., pp. 3-367 - 3-380). Society of Fire Protection Engineers, Bethesda, MD (2002)
3. Rinne, T., Tillander, K., and Grönberg, P. 2010. Data collection and analysis of evacuation situations (pp A86-A92). VTT Research Notes 2562, Finland.