RESEARCH ARTICLE

Constrained Linear Movement Model (CALM): Simulation of passenger movement in airplanes

Mehran Sadeghi Lahijani [1] *, **Tasvirul Islam**[2‡], **Ashok Srinivasan**[2‡], **Sirish Namilae**[3‡]

1 Department of Computer Science, Florida State University, Tallahassee, Florida, United States of America, **2** Department of Computer Science, University of West Florida, Pensacola, Florida, United States of America, **3** Aerospace Engineering Department, Embry-Riddle Aeronautical University, Daytona Beach, Florida, United States of America

☯ These authors contributed equally to this work.
‡ These authors also contributed equally to this work.
* sadeghil@cs.fsu.edu

Abstract

Pedestrian dynamics models the walking movement of individuals in a crowd. It has recently been used in the analysis of procedures to reduce the risk of disease spread in airplanes, relying on the SPED model. This is a social force model inspired by molecular dynamics; pedestrians are treated as point particles, and their trajectories are determined in a simulation. A parameter sweep is performed to address uncertainties in human behavior, which requires a large number of simulations. The SPED model's slow speed is a bottleneck to performing a large parameter sweep. This is a severe impediment to delivering real-time results, which are often required in the course of decision meetings, especially during emergencies. We propose a new model, called CALM, to remove this limitation. It is designed to simulate a crowd's movement in constrained linear passageways, such as inside an aircraft. We show that CALM yields realistic results while improving performance by two orders of magnitude over the SPED model.

Introducion

Pedestrian dynamics deals with modeling the movement of individuals, often as a part of a crowd. Its has been used in a wide variety of applications, from panic simulation and crowd behavior analysis [1–7] to diseases-spread modeling and robotics [6, 8–13].

The most popular models for the simulation of pedestrian dynamics are social force models [14]. Two sets of forces determine the movement of each person in these models. Propulsive forces tend to move a pedestrian toward that pedestrian's destination. Repulsive forces, induced by nearby pedestrians or physical obstacles in the environment, on the other hand, tend to impede this movement. Newton's law of motion is used with the aggregation of the above forces to compute the trajectories of the pedestrians. These trajectories are subsequently analyzed to extract the desired information for the target application.

The Self Propelled Entity Dynamics (SPED) model is one notable social dynamics model [8–10, 12]. It has been used to determine contact patterns while passengers board and disembark from planes. These contact patterns are subsequently used to identify policies that would hinder the spread of viral diseases in air-travel [10]. The results of this research, analyzing the risk of an Ebola outbreak, was widely covered in more than 75 news outlets around the world [15].

A critical challenge in the use of such models, especially during epidemics, lies in dealing with the intrinsic uncertainties in human behavior. The above application handles it by parameterizing the sources of uncertainty and then performing a sweep of the parameter space to generate all possible scenarios [8, 12]. Vulnerabilities of different policies can then be examined under this exhaustive set of possible scenarios.

The number of scenarios that need to be generated is large, leading to a high computational effort. Chunduri et al. [8] observed that it is crucial to have the results of decision-support systems in less than a couple of minutes during decision meetings. The average runtime of SPED code for a *single simulation* is over 350 seconds. This makes it infeasible to meet the time constraints needed for decision support. Consequently, there is a need for a model that can simulate the same application as SPED, while being faster.

The primary contribution of this paper lies in proposing a fast pedestrian dynamics model—*CALM*—for constrained linear movement of individuals in a crowd. Similar to SPED, this model is designed to simulate movement in narrow, linear, passageways, such as inside airplanes. Our results show that CALM performs almost 60 times faster than the SPED model. Apart from this performance gain, we have modeled additional behavioral features of pedestrians. Therefore, the CALM model can overcome the limitations of SPED in a decision support context where real time results are required.

Model description

Various types of methods have been used in pedestrian dynamics, such as cellular automaton [16, 17], social force [5, 14], and lattice-gas [3, 18] models. Social force models are among the most popular, and are particularly effective in generating fine-scale trajectories of each pedestrian. Both the existing SPED model and the proposed CALM model are social force models. We first summarize the SPED model and then explain how its critical qualitative features are carried forward to the CALM model, while enabling it to be computationally more efficient.

SPED model

Social force models treat each pedestrian as a point particle, analogous to an atom in molecular dynamics. Molecular dynamics uses models that capture the actual attractive and repulsive forces between atoms, which govern the movement of atoms. In social force models, conceptual forces are defined that perform a similar role, either increasing the speed or decreasing it. Each pedestrian wishes to reach a certain destination, which motivates a propulsive force f^{prop}. The pedestrian may be hindered by other pedestrians and fixed objects, whose net effect is considered a repulsion, f^{rep}. Newton's law of motion, shown in Eq (1), is then solved using a finite difference solver to determine the position x_i and velocity v_i of each pedestrian i with mass m_i as a function of time t.

$$m_i \frac{dv_i}{dt} = m_i \frac{d^2 x_i}{dt^2} = f_i^{prop}(t) + f_i^{rep}(t) \tag{1}$$

Both SPED and CALM use the model for propulsive force given in Eq (2), which is common in social dynamics [14]. Here, τ is the average reflex time of an individual and v_{0i} is the

desired velocity of pedestrian i (if there were no obstacles or other pedestrians). SPED assigns normally distributed random values for $|v_{0i}|$, with the direction of v_{0i} determined by the pedestrian's destination. The parameters for the distribution are based on empirically observed values for human movement [19].

$$f_i^{prop} = \frac{(v_{0i} - v_i)}{\tau} \times m_i \qquad (2)$$

Social dynamics models typically differ in their definition of the repulsive force. The basic idea is that the speed of a pedestrian should decrease on getting close to other persons or fixed objects on their path.

SPED computes repulsion for each pedestrian as follows. It considers the distance d_i of the closest pedestrian or fixed object in the direction of motion. It then uses one of the following criteria, depending on the value of d_i. (i) If d_i is very large, then it ignores the repulsive force in Eq (1). (ii) If d_i is very small, then it adapts the Lennard-Jones potential from molecular dynamics to compute f_i^{rep} in Eq (1) as explained below. (iii) If d_i is between these extremes, then it decreases the speed of the pedestrian by multiplying the speed of the pedestrian by a factor α to obtain the speed at the next time step of the finite difference scheme. That is, $v_i(t + \Delta t) \leftarrow \alpha v_i(t)$, where α is computed using Eq (3). In that equation, λ is a constant giving the desired stopping threshold of a pedestrian; that is, the pedestrian aims to stop when the distance to the nearest pedestrian reaches this value.

$$\alpha = 1 - \frac{\lambda}{d_i} \qquad (3)$$

A modified version of the Lennard-Jones potential from molecular dynamics is used to compute repulsion when d_i is very small. In that case, the distance r_{ij} to each of the other pedestrians j is determined and a "potential" computed for each as shown in Eq (4) [10]. The net repulsive force on pedestrian j is given by $f_i^{rep} = \sum_{j \neq i} f_{ij}$, where f_{ij} is the negative of the gradient of the potential V_{ij}^{LJ}. Repulsion is computed in an identical manner if the closest entity is a fixed obstacle rather than a pedestrian.

$$V_{ij}^{LJ} = \epsilon \frac{\sigma}{r_{ij}^{12}} \qquad (4)$$

Here, ϵ and σ are suitably defined constants obtained by fitting against empirically determined values. A few other behavioral characteristics are directly incorporated into the SPED code. (i) Passengers take a certain time to stow their luggage while boarding. (ii) Passengers take a certain time to retrieve their luggage while disembarking. (iii) Passengers let those in the row ahead of them leave first while disembarking.

We observed by profiling the code that the Lennard-Jones potential is the primary computational bottleneck. In addition, the methodology to reduce the speed in the intermediate range is numerically awkward for the following reason. In order to verify convergence, a typical test would be to repeat the simulation with a smaller time step size and check if the results are similar. However, the method adopted in SPED makes the velocity a function of the time step size, which would not yield similar results when the time step size is changed. We designed the CALM model to overcome both limitations, and also added an additional behavioral feature that ensures that the simulation always progresses.

CALM model

We note that actual human movement is not precisely defined by any particular potential. Rather, it varies so much that it is sufficient to capture its qualitative tendencies, and then use a parameter sweep to examine the range of movement patterns. We, therefore, define a simpler repulsive force by using a single curve that yields results qualitatively similar to SPED. It must satisfy the basic requirements that the speed should be the desired speed $|v_{0i}|$ when the nearest pedestrian is far away, and should decrease to 0 when the nearest passenger gets too close.

We accomplish this by using Eq (1) with propulsion defined by Eq (2) as with SPED, and repulsion by Eq (5) as defined below, for a value of β explained below.

$$f_i^{rep} = \frac{(\beta - 1)v_{0i}}{\tau} \times m_i \tag{5}$$

Fig 1 shows the values of this repulsion against d_i with three different values of v_{0i}. The force becomes negative when a pedestrian reaches close to another, with stronger repulsion for faster moving pedestrians.

Use of the above repulsive force is equivalent to solving Newton's law of motion with the net force defined by Eq (6) below, which is the actual equation that we solve.

$$m_i \frac{d^2 x_i}{dt^2} = \frac{(\beta v_{0i} - v_i)}{\tau} \times m_i \tag{6}$$

Comparing the above with the expression in Eq (2) shows that the above formulation aims to achieve a velocity of βv_{0i}. We choose β as a function of d_i such that βv_{0i} is a suitable velocity for the pedestrian whose nearest pedestrian is at a distance d_i.

We accomplish this goal as follows. We calculate the variation of v_i with d_i for SPED and fit a curve that will yield a qualitatively similar result for CALM. The specific details are as follows. We note that the typical desired speed v_{0i} varies around 1m/s. We solve SPED's equation when there are only two pedestrians, one stationary and another moving toward that pedestrian with an initial velocity of 1m/s while at a distance of 4m, with only the repulsion term
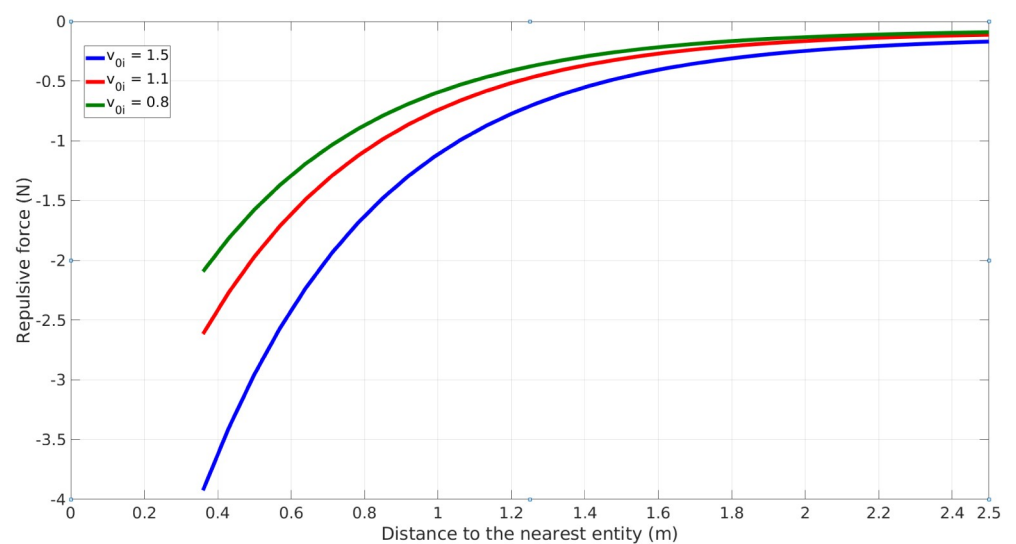
Fig 1. Repulsive force as a function of d_i.
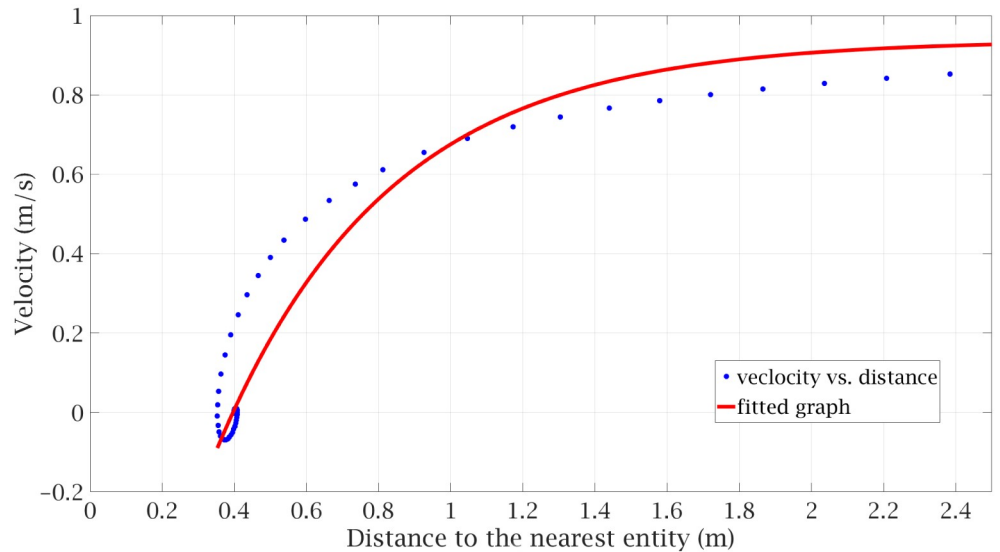
Fig 2. Fitting the repulsion term of CALM (solid line) to the SPED solution (dotted line).

considered. The intermediate zone is taken as a distance from 0.4m to 2.0m. We get the result shown in Fig 2.

In fitting the CALM model, we assume β of the form given in Eq (7) and determine parameters that give a best fit to the SPED results. We obtain $a = 2.11$, $b = 0.366$, and $c = 0.966$. We use the same value of β for any value of v_{0i}. The range of values of v_{0i} is small around 1ms/, and so the same value of β is considered a reasonable scaling factor.

$$\beta = c - e^{-a(d_i - b)} \tag{7}$$

Fig 3 demonstrates a comparison of the pedestrian dynamics results between the CALM and the SPED model in three different cases of $v_{0i} = 0.8(m/s)$, $v_{0i} = 1.0(m/s)$, $v_{0i} = 1.5(m/s)$. In
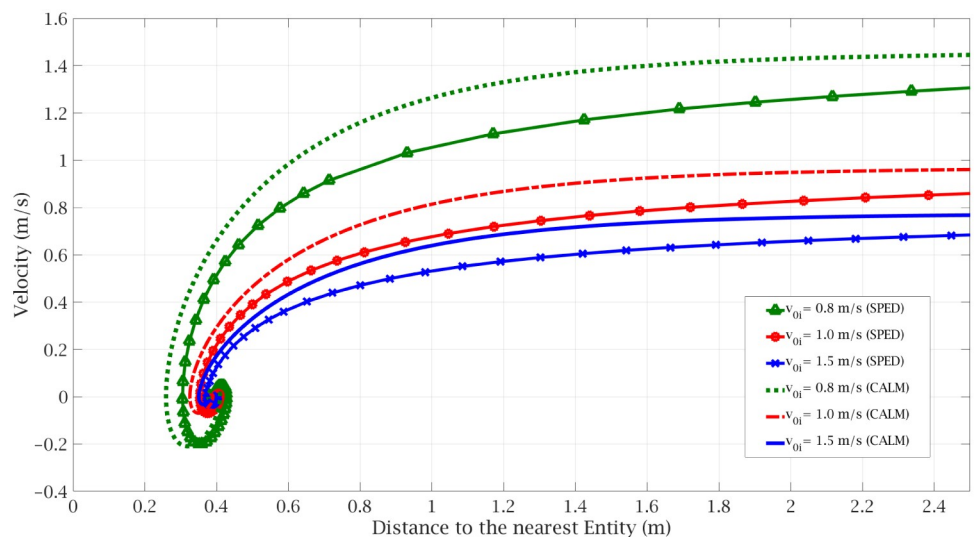
Fig 3. Comparing the pedestrian dynamics results of the SPED model vs the CALM model in three different cases.

each case, we assumed that a pedestrian is moving toward a stationary entity. These figures show a qualitative match between SPED and our model for this situation. *Given a range of values of v_{0i} for SPED, we can find a range of values for CALM that yield similar results.*

Deadlock. We incorporate the same behavioral features as SPED in our simulations. In addition, we include the following. We have designed the CALM model for simulation of movement in narrow passageways, such as airplane aisles. In these situations, people cannot walk side-by-side. As a consequence, sometimes, a deadlock situation can happen when two people try to get right of way but neither can move forward. The SPED model will not progress in these situations.

We have designed a mechanism to resolve this. One of the passengers will be declared the winner, with a random component to the decision, and get the right of way. This reflects human behavior in practice, where one person would yield to another.

Application to air travel

As we discussed in the introduction section, using pedestrian dynamics models for simulation of passengers movement in the airplanes is a critically important application for public health policy analysis. Namilae et al. [10] used the SPED model for suggesting policies that help to mitigate the risk of diseases spread during the flights. We use the CALM model in the same application to demonstrate that it yields realistic results. In this section, we explain the implementation details for airplane boarding and deplaning procedures.

Deplaning

We initialize the simulation by inputting the initial positions of passengers and physical obstacles from a file. We also assign each passenger a value of $|v_{0i}|$ drawn independently from a Gaussian distribution with mean $\overline{v_0}$ and standard deviation 0.2 m/s. The parameter $\overline{v_0}$ varies in different simulations, with $1.1 \leq \overline{v_0} \leq 1.3$ m/s. These values are derived from empirical data available for pedestrian speed [19].

After the initialization, each step of the simulation computes the position of all passengers at several different points in time. This is accomplished by using an explicit Euler scheme to solve Eq (1) with a time-step size Δt of 0.005s for the CALM model. We repeat these steps until all the passengers leave the plane. Algorithm 1 gives the pseudocode of the deplaning procedure.

Algorithm 1 Deplaning

```
while there are passengers in the plane do
  for each remaining passenger Pᵢ do
    Find the nearest passenger or physical obstacle to Pᵢ on its path
    Compute the repulsion
  end for
  for each remaining passenger Pᵢ do
    Compute the propulsion
    Update the velocity and position of Pᵢ
    Check for deadlock and update Pᵢ's state if necessary
  end for
end while
```

During the deplaning procedure, each passenger will go through a few different states. The initial state of each passenger is going *toward the overhead bin* to collect the carry-on baggage. We assume that each passenger has a bag that is in the nearest overhead bin and that they take 5-12s in a state where they are *collecting baggage*. Passengers will then attempt to *go toward the center of the aisle* and subsequently *move forward in the aisle* toward the exit. In the latter state, they will first wait for the passengers in the rows in front of them to go ahead before they move
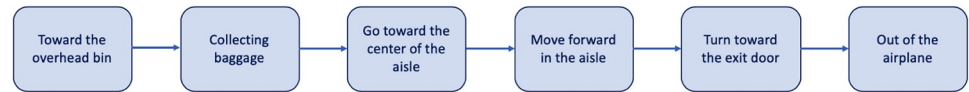
Fig 4. States of passengers while deplaning.

forward. When passengers reach the end of the aisle, they have to *turn toward the exit door* and leave the plane. Once a passenger is *out of the airplane*, we remove that passenger from the simulation. Fig 4 shows the state diagram for passengers during the deplaning, and Fig 5 demonstrates the progress of a disembarkation simulation.
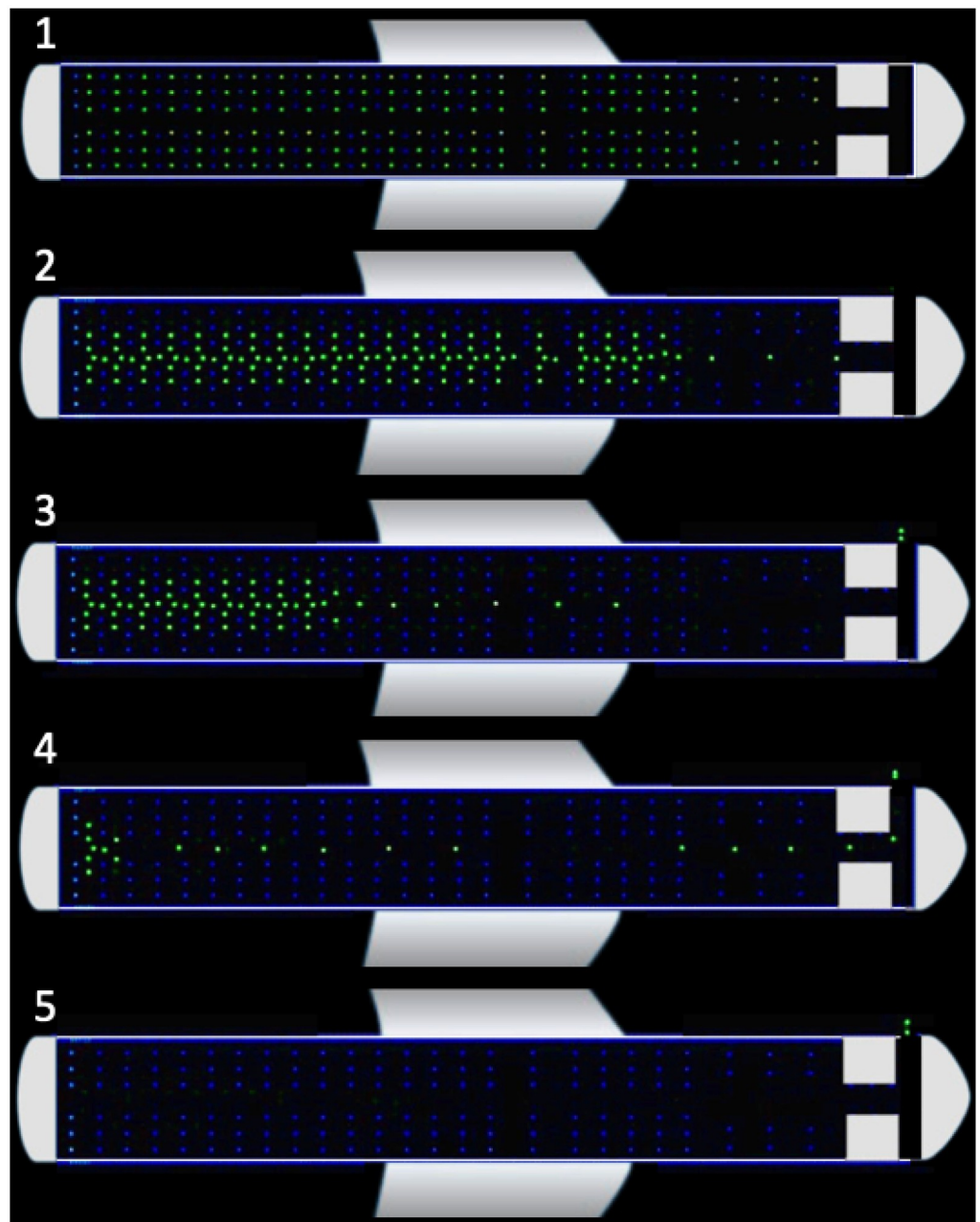
Fig 5. Progress of a deplaning simulation.

In addition to $\overline{v_0}$, we use five parameters to address the uncertainties in human behavior during the deplaning process. Passengers move from the seat to the overhead bin at a fraction of their maximum reachable speed. Therefore, in our model, we use a coefficient $0 < $ *toward_bag_speed_coefficient* < 1 that multiplies v_{0i} when passengers are in the *toward_the_overhead_bin* state. Similarly, passengers' alignment on the center of the aisle, after taking their bag, takes place with a fraction $0 < $ *aligning_speed_coefficient* < 1 of their maximum reachable speed. After this step, passengers should wait for the passengers in the front rows to move toward the end of the aisle first. Hence, the first passenger from row i can proceed only if the last passenger from row $i - 1$ has already moved in the aisle for *aisle_distance_threshold* meters. People often reduce their speed when getting close to an intersection at which they want to turn. In our model, passengers, reduce their speed by a factor *intersection_speed_coefficient* when their distance to the end of the aisle is less than *intersection_distance_threshold* meters.

According to [20], the average deplaning rate for the planes is 15-17 passengers per minute. In our application, the goal is to capture some extreme scenarios in addition to normal ones [8]. As a consequence, we tuned the parameter values for the CALM model to generate simulations with a slightly more extensive range of deplaning rates. We use 1000 parameter combinations for running simulations of the deplaning procedure for a full Airbus A320 with 144 seats and used the results of this parameter sweep for tuning the parameters of the CALM model. For this airplane, the deplaning rate of 15-17 passengers per minute yields to deplaning time of 8.47 to 9.6 minutes, and the CALM model generates simulations with the deplaning time of 6.52 to 11.10 minutes after parameter tuning. Table 1 shows the range of values for all six parameters of the CALM model after such adjustment.

Boarding

The CALM model can also be used for boarding of airplanes with implementation details being similar to deplaning to a large extent. The main difference lies in the state diagram of the passengers, which is roughly in the reverse order of deplaning, as shown in Fig 6. Passengers are initially outside of the airplane. As the simulation starts, passengers move toward the aisle from the exit door. Then, they go into the aisle until they reach the row of their seat. There, they will stow their bag in the overhead bin, and then go to their seat.

We have implemented boarding with three zones. Other boarding strategies can be implemented without changing the code, just by using suitable input files.

Boarding involves a few parameters, similar to deplaning. The parameter $\bar{v}_0$ was explained earlier. We assume passengers are in an aisle outside the airplane (behind the airplane door) and will move toward the airplane door when the passenger in front of them has moved for *line_distance_threshold* meters. Passengers reduce their speed by a factor *intersection_speed_coefficient* when their distance to the intersection of the aisle is less than

Table 1. CALM model parameter ranges for deplaning.

Parameter	Minimum Value	Maximum Value
$\overline{v_0}$	1.1 m/s	1.3 m/s
toward_bag_speed_coefficient	0.2	0.6
aligning_speed_coefficient	0.2	0.7
aisle_distance_threshold	0.5 m	1.6 m
intersection_speed_coefficient	0.2	0.8
intersection_distance_threshold	0.2 m	1.5 m
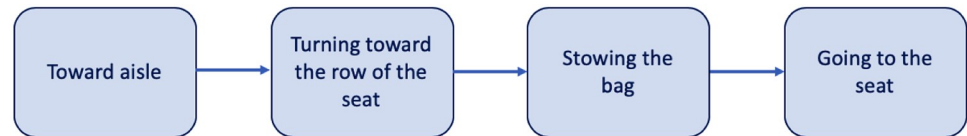
Fig 6. State diagram of passengers during the boarding procedure.

intersection_distance_threshold meters. After passengers place their baggage in the overhead bin, they will move toward their seat with a fraction of their maximum reachable speed, that we call *toward_seat_speed_coefficient*. These parameters are analogous to those for deplaning, and we use the same range of values for them, as shown in Table 2.

Results

We first show that the CALM models results are consistent with empirically observed metrics for disembarkation in airplanes. We then show that the CALM model yields substantial performance gains over the SPED model.

Experimental setup

We run all our experiments on the Frontera supercomputer at the Texas Advanced Computing Center. This system consists of 8008 compute nodes with 56 cores per node for a total of 448448 cores and ranks the 5th fastest supercomputer in the world. Each node contains two Xeon Platinum 8280 28C processors running at 2.7GHz with 128 GB memory. Nodes are connected through Mellanox Infiniband HDR-100 network connected in a fat tree topology.

We use a scrambled Halton low discrepancy sequence for performing efficient parameter sweep [8]. We employ dynamic load balancing for running this parameter sweep, using a master-worker algorithm where the master assigns a simulation to a core that has just completed its previous simulation. We use one node with 32 cores in all of our experiments. Each simulation of the parameter sweep is run on one core, with each core running multiple simulations.

Implementations of the CALM model for running parameter sweeps of deplaning and boarding simulations can be found at https://gitlab.com/Mehran_SL/calm.

Model validation

We validate our results by examining disembarkation times on three different types of airplanes. A single simulation does not capture the variety of human movement patterns, and so we perform a parameter sweep with 1000 different combinations of parameter values, covering the range mentioned in Table 1. We compare it against the empirically observed deplaning rate of 15-17 passengers per minute [20]. The airplanes considered are: Boeing B757-200 with

Table 2. CALM model parameter ranges for boarding.

Parameter	Minimum Value	Maximum Value
$\overline{v_0}$	1.1 m/s	1.3 m/s
line_distance_threshold	0.5 m	1.6 m
intersection_speed_coefficient	0.2	0.8
intersection_distance_threshold	0.2 m	1.5 m
toward_seat_speed_coefficient	0.2	0.6

Table 3. Comparison of predicted and empirically observed deplaning times.

Airplane	Deplaning time	Empirically-observed deplaning time
B757–200 (182 seats)	[8.21, 16.43] (min)	[10.71, 12.13] (min)
B757–200 (201 seats)	[9.32, 15.17] (min)	[11.82, 13.4] (min)
CRJ–200 (50 seats)	[1.46, 4.06] (min)	[2.94, 3.33] (min)

182 seats, B757-200 with 201 seats and CRJ-200 with 50 seats. In all of our experiments, all the planes are full.

Table 3 shows the ranges for deplaning time of each airplane based on empirical observations, along with the results from our simulations using the CALM model.

As the results of our experiments show, the empirically-observed deplaning time ranges are subsets of the ranges produced by the CALM model. We can draw two conclusions from these results. First, the results demonstrate that the CALM model generates results for all the expected scenarios. Second, the CALM model provides results that are a little outside the normal range. This was a deliberate design choice because our application goal is to generate rare scenarios that can capture extreme events [8].

In addition to validating the deplaning times, we selected several random simulations and checked the video output to examine if the behavior was realistic, as is commonly done in validation of pedestrian dynamics [21]. Simulation results in all these tests were consistent with the empirically observed behavior of passengers during deplaning and boarding, such as formations of clusters.

Moreover, it is common in this field use other features of crowds movements, such as the relationship between the density (number of passengers per unit area) and space-mean velocity, to validate models of pedestrian movements [22–24]. We assess the relationship between density and velocity in the CALM model and compare it against the empirically-observed trend [25, 26].
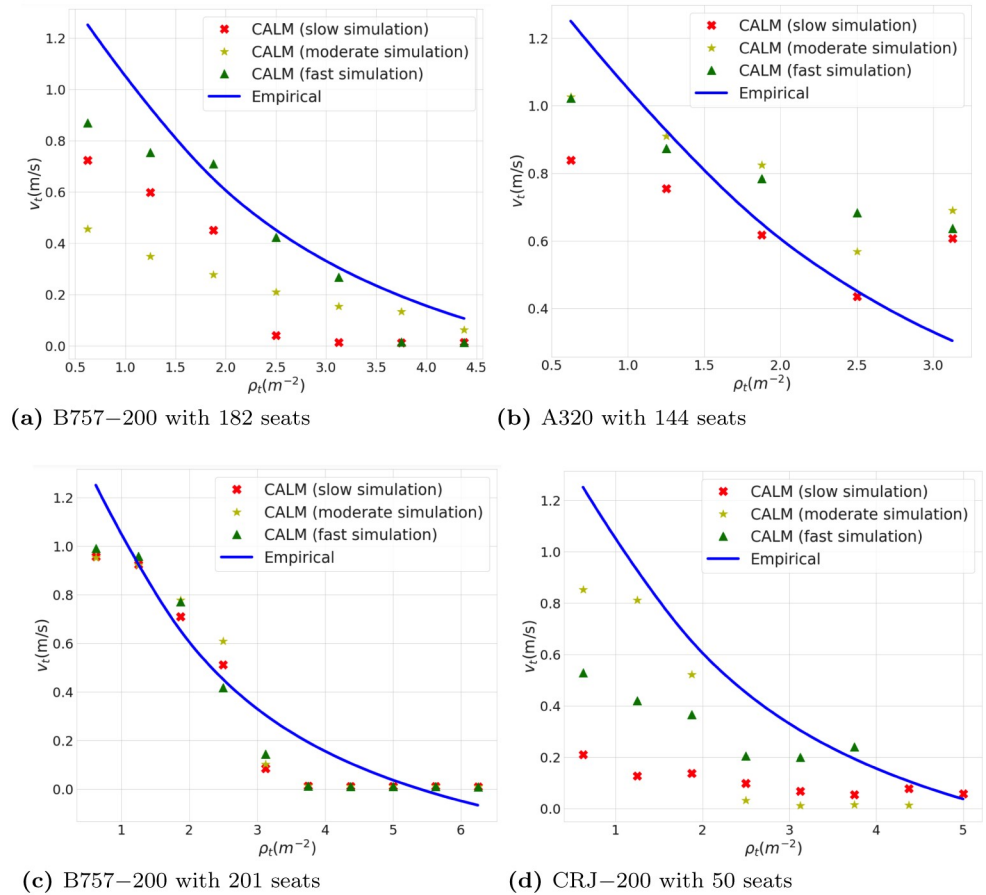
Analyzing these characteristics of the flow at bottlenecks in passengers' paths is critical as these bottlenecks usually have a significant role in crowd movements [22]. Therefore, we measure the density and space-mean velocity of passengers toward the end of the aisle, where passengers decrease their speed in order to turn toward the exit. We define this bottleneck region as a rectangle with a width of 0.4m (width of an average person) centered at the center of the aisle. The length of this rectangle is 4 meters for each airplane, except for the small CRJ-200 airplane, for which it is 3 meters.

We print the current position and velocity of all passengers every 100 iterations (which is equal to 0.5 seconds of real-world time) in deplaning simulations and then compute the density and space-mean velocity of passengers in the bottleneck for each of the sample iterations. Eqs (8) and (9) define these two factors.

$$\rho_t = \frac{N_t}{A} \tag{8}$$

$$\overline{v_t} = \frac{N_t}{\sum_{i=1}^{N_t} \frac{1}{N_t}} \tag{9}$$

In these equations, ρ_t, v_t and N_t represent density in m^{-2}, space-mean velocity in $\frac{m}{s}$ and number of passengers in the bottleneck at time t, respectively, and A is the area of the bottleneck region.

(a) B757−200 with 182 seats

(b) A320 with 144 seats

(c) B757−200 with 201 seats

(d) CRJ−200 with 50 seats

Fig 7. Fundamental diagram of passenger movement during deplaning.

https://doi.org/10.1371/journal.pone.0229690.g007

For each of the four airplanes that we consider in this paper, we select three deplaning simulations randomly from the fastest third, the slowest third and the moderate third of 1000 simulations in our parameters sweep. Then we compute the density and space-mean velocity for these simulations. For each value of density, different values of space-mean velocities may be observed. We compute the average of these space-mean velocities for each value of density in our plot of the fundamental diagrams.

Fig 7 presents the fundamental diagrams that show the relation between density and velocity for the four airplanes. We compare results from our model against empirical results. Empirical results have been determined through a relationship between velocity and density given by Eq (10), which fits empirical data [25, 26]. In this equation, $v_{F,hi}(D)$, $v_{F,hf}$, D and D_{max} are velocities at a particular density, velocity when density is zero, density, and the maximum possible density respectively. λ is a constant with the value of 1.913 [26]. We use the values of $v_{F,hf} = 1.34(m/s)$ and $D_{max} = 5.4(person/m^2)$ as specified in literature [26].

$$v_{F,hi}(D) = v_{F,hf} \times \left(1 - e^{-\lambda \times \left(\frac{1}{D} - \frac{1}{D_{max}}\right)}\right) \qquad (10)$$

Fig 7 shows that the results of the model reflect empirically observed trends, with decrease in speed at high densities. Velocities in the CALM model tend to be a little below the

Table 4. Comparing performance of the CALM and the SPED model.

	SPED model	CALM model	Speedup
Average runtime of a single simulation	277.59 (s)	4.7 (s)	59.06
Runtime of the parameter sweep	9209.4 (s)	156.86 (s)	58.70

empirical curve for the following reason. The empirical formula sets the maximum speed of passengers as 1.34(m/s). However, CALM uses a variety of values for this in the parameter sweep, with the largest possible value set as 1.3(m/s), and the typical value much lower. This makes a notable impact on the extremely small CRJ-200 airplane. Passengers move more slowly in such a small aircraft, and thus the low value reflects this reality. In summary, given the variability of human behavior, the model correctly captures the relationship between density and velocity.

Performance analysis

We compare the performance of CALM and SPED using a parameter sweep of size 1000 for disembarkation process on an Airbus A320 with 144 seats. We present the runtime of the parameter sweeps and the average runtime of a single simulation that is computing by getting the average runtime of 1000 simulations of the parameter sweep in Table 4. We have repeated these experiments five times and we report the minimum runtime among the five samples here. The results show that the average runtime for simulations of the SPED model is 277.52 seconds while it is only 4.7 seconds for simulations of the CALM model. Consequently, the CALM model performs approximately 59 times faster than the SPED model. The parameter sweep of the SPED model took 9209.04 seconds while the parameter sweep of the CALM model took 156.86 seconds to complete. Therefore, the CALM model runs approximately 59 times faster than the SPED model when used for the parameter sweep of size 1000.

There are two significant reasons for this considerable performance difference between these two models. First, we used a simpler force formulation to decrease its computational time in our model. In particular, much of the reduction in time was obtained by eliminating the Lennard-Jones potential and using a single simple formula to determine the impact of repulsion. This was the primary reason for decrease in simulation time, and led to around a factor 20 improvement in performance. Second, we removed passengers that had reached their destinations from the simulation. Both SPED and CALM models use algorithms of $O(N^2)$ time complexity where N is the number of passengers in the simulation. For the SPED model, this N is constant during the whole simulation. By removing passengers, N linearly decreased in the CALM model. This provided the secondary performance boost, yielding almost a factor 3 improvement.

Conclusions

Pedestrian dynamics is finding increased applications in diverse real-world problems. However, slow performance of existing models has been a bottleneck for policy analysis, especially in an emergency. In decision support meetings, a result is required in the order of a couple of minutes. The new CALM model delivers results in that time frame in contrast to the SPED model. Our validation simulations also show that the CALM model produces results that are consistent with empirical observations.

Acknowledgments

Author Contributions

Data curation: Mehran Sadeghi Lahijani, Tasvirul Islam.

Formal analysis: Mehran Sadeghi Lahijani, Ashok Srinivasan.

Investigation: Mehran Sadeghi Lahijani.

Methodology: Sirish Namilae.

Project administration: Mehran Sadeghi Lahijani, Ashok Srinivasan.

Supervision: Ashok Srinivasan, Sirish Namilae.

Validation: Mehran Sadeghi Lahijani, Tasvirul Islam, Ashok Srinivasan, Sirish Namilae.

Writing – original draft: Mehran Sadeghi Lahijani.

Writing – review & editing: Ashok Srinivasan.

References

1. Helbing D, Johansson A. In: Meyers RA, editor. Pedestrian, crowd and evacuation dynamics. New York, NY: Springer New York; 2009. p. 6476–6495. Available from: https://doi.org/10.1007/978-0-387-30440-3_382.

2. Garcimartín A, Pastor J, Martín-Gómez C, Parisi D, Zuriguel I. Pedestrian collective motion in competitive room evacuation. Scientific Reports. 2017; 7(1):10792. https://doi.org/10.1038/s41598-017-11197-x PMID: 28883459

3. Zou Y, Xie J, Wang B. Evacuation of pedestrians with two motion modes for panic system. PLOS ONE. 2016; 11(4):e0153388. https://doi.org/10.1371/journal.pone.0153388 PMID: 27055024

4. D H, Farkas I, Vicsek T. Simulating dynamical features of escape panic. Nature. 2000; 407(6803):487–90. https://doi.org/10.1038/35035023

5. Mehran R, Oyama A, Shah M. Abnormal crowd behavior detection using social force model. In: In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2009.

6. von Sivers I, Templeton A, Kunzner F, Koster G, Drury J, Philippides A, et al. Modelling social identification and helping in evacuation simulation. Safety Science. 2016; 89:288–300. https://doi.org/10.1016/j.ssci.2016.07.001

7. Moussaïd M, Perozo N, Garnier S, Helbing D, Theraulaz G. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. PLOS ONE. 2010; 5(4):e10047. https://doi.org/10.1371/journal.pone.0010047 PMID: 20383280

8. Chunduri S, Ghaffari M, Lahijani MS, Srinivasan A, Namilae S. Parallel low discrepancy parameter sweep for public health policy. In: Cluster, Cloud and Grid Computing (CCGrid), 2018 18th IEEE/ACM International Symposium. IEEE; 2018.

9. Namilae S, Derjany P, Mubayi A, Scotch M, Srinivasan A. Multiscale model for pedestrian and infection dynamics during air travel. Physical Review E. 2017; 95(5). https://doi.org/10.1103/PhysRevE.95.052320 PMID: 28618588

10. Namilae S, Srinivasan A, Mubayi A, Scotch M, Pahle R. Self-propelled pedestrian dynamics model: Application to passenger movement and infection propagation in airplanes. Physica A: Statistical Mechanics and its Applications. 2017; 465:248–260. https://doi.org/10.1016/j.physa.2016.08.028

11. Song W, Ma J, Fu L. Proceedings of the international conference on pedestrian and evacuation dynamics 2016. Collective Dynamics. 2016; A11:1–618.

12. Srinivasan A, Sudheer CD, Namilae S. Optimizing massively parallel simulations of infection spread through air-travel for policy analysis. In: In Cluster, Cloud and Grid Computing (CCGrid), 2016 16th IEEE/ACM International Symposium. IEEE; 2016. p. 136–145.

13. Knepper RA, Rus D. Pedestrian-inspired sampling-based multi-robot collision avoidance. In: 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication. IEEE; 2012. p. 94–100.

14. Helbing D, Molnar P. Social force model for pedestrian dynamics. Physical Review E. 1995; 51(5):4282. https://doi.org/10.1103/PhysRevE.51.4282

15. VIPRA news coverage; accessed July 31, 2019. Available from: https://www.cs.fsu.edu/vipra/?page_id=37.

16. Schadschneider A. Cellular automaton approach to pedestrian dynamics-theory. arXiv preprint cond-mat/0112117. 2001;.

17. Burstedde C, Klauck K, Schadschneider A, Zittartz J. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. Physica A: Statistical Mechanics and its Applications. 2001; 295(3-4):507–525. https://doi.org/10.1016/S0378-4371(01)00141-8

18. Muramatsu M, Irie T, Nagatani T. Jamming transition in pedestrian counter flow. Physica A: Statistical Mechanics and its Applications. 1999; 267(3-4):487–498. https://doi.org/10.1016/S0378-4371(99)00018-7

19. Zębala J, Ciępka P, RezA A. Pedestrian acceleration and speeds. Problems of Forensic Sciences. 2012; 91:227–234.

20. Wald A, Harmon M, Klabjan D. Structured deplaning via simulation and optimization. Journal of Air Transport Management. 2014; 36:101–109. https://doi.org/10.1016/j.jairtraman.2014.01.001

21. Namilae S, Derjany P, Mubayi A, Scotch M, Srinivasan A. Multiscale model for pedestrian and infection dynamics during air travel. Physical Review E. 2017; 95(5):052320. https://doi.org/10.1103/PhysRevE.95.052320 PMID: 28618588

22. Liao W, Zhang J, Zheng X, Zhao Y. A generalized validation procedure for pedestrian models. Simulation Modelling Practice and Theory. 2017; 77:20–31. https://doi.org/10.1016/j.simpat.2017.05.002

23. Steffen B, Seyfried A. Methods for measuring pedestrian density, flow, speed and direction with minimal scatter. Physica A: Statistical mechanics and its applications. 2010; 389(9):1902–1910. https://doi.org/10.1016/j.physa.2009.12.015

24. Seyfried A, Schadschneider A. Fundamental diagram and validation of crowd models. In: International Conference on Cellular Automata. Springer; 2008. p. 563–566.

25. Weidmann U. Transporttechnik der fussgänger. Strasse und Verkehr. 1992; 78(3):161–169.

26. Lämmel G, Rieser M, Nagel K. Bottlenecks and congestion in evacuation scenarios: A microscopic evacuation simulation for large-scale disasters. In: Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal; 2008.