Summary

Evacuation algorithms are vital in modeling crisis situations and understanding building safety standards. Prior work on evacuation procedures has primarily been *prescriptive*, generating optimal protocols for evacuee movement. These methods, however, rely on strong behavioural assumptions about the evacuees. Other works take a more *descriptive* approach, simulating individual evacuee behaviour in a crisis, while offering limited *prescriptive* power.

We propose an online algorithm, based on a novel hybrid model which combines Linear Programming (LP) optimization and Cellular Automata (CA) simulations for evacuee behavior. This offers greater resolution than existing prescriptive methods, along with an actionable model that realistically simulates evacuee behavior. Our model, at any given stage of an evacuation, uses linear programming to generate an optimal prescription for evacuee movement and then identifies key areas where simulated evacuee behavior differs from the optimal prescription. This allows emergency responders to strategically deploy resources throughout the building, guiding civilians in the most critical areas.

We simulate this routing model on a finely discretized graph representation of the Louvre, showing an 80% increase in successfully evacuated civilians relative to our baseline CA model simulations. We further show how routing to hidden exits may lead to twice-as-fast evacuations. Finally, we identify potential bottlenecks with measurements of current flow centrality over graphs.

Crucially, our approach is modular. For example, we may combine the prescriptions of our LP model with other types of simulations, or we may run simulations independently of LP to predict evacuee behavior in advance of an emergency. As evidenced by our results, however, our unique combination of these two models is particularly strong.

Prescriptive Integration of Optimization and Simulation-based Evacuation Algorithms

ICM Contest Question D

SHIVAM NADIMPALLI TIMOTHY SUDOJINO NICHOLAS TOMLIN

January 28, 2019

Contents

1 Introduction

Evacuation scenarios require numerous split-second decisions to be made, both by civilians and by emergency response teams. Under these extenuating and often terrifying conditions, our choices are frequently non-optimal and run counter to the collective well-being. In light of recent terrorist attacks in France [2], we explore how algorithmic methods may be used to predict optimal behavior in adverse evacuation scenarios and generate real-time prescriptions for emergency response personnel in such situations. As a case study, we simulate our prescribed evacuation routes for the *Musée du Louvre* in Paris, France.

Unique to this work, we develop a real-time optimization-simulation approach to evacuation which allows us to *dynamically* place security personnel in optimal locations for routing civilians. In particular, we develop two models: (1) an optimization approach over coarse graph representations, and (2) a simulation over fine-grained graph representations. These models are run in tandem, in real time, to achieve optimal results, as shown in Figure 1 below:

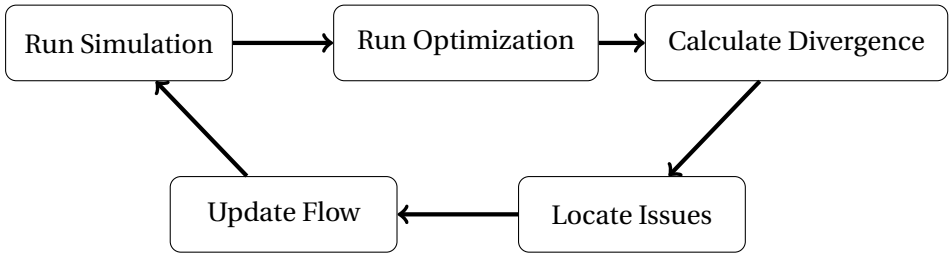

Figure 1: Model Workflow

We calculate optimal evacuation strategies with an Linear Programming (LP) problem whose solution may be approximated in polynomial time, modeling motion over a graph network with discretized time units. We further consider a number of extensions to the baseline LP model, such as strategies for limited-mobility visitors and deployment of security personnel to hidden exits. The full details of our optimization method are described in Section 4.

While the LP model offers a *prescription* should an evacuation take place, along

with actionable ways for emergency personnel to respond, the model lacks a description of how the evacuation might occur under realistic conditions. In section 5, we explore several factors that come into play in real evacuation situations, such as panic and other social forces. One consequence of these observations is that we cannot expect panicked civilians to naturally follow optimal evacuation protocols. Instead, we must deploy emergency personnel to optimally route civilians according to their best interests.

To determine optimal routing, we develop a fine-grained stochastic cellular automata (CA) simulation of crowd behavior during evacuations. We overlay this model with the LP optimization and evaluate where non-optimal movement is most likely to occur. Once these crucial "problem areas" are located, we prescribe emergency personnel to route civilians accordingly, resulting in safer and more efficient evacuations. This model is described in Section 6.

In a real evacuation procedure, these models would be used dynamically, in real time. In particular, in the course of a real evacuation procedure, we locate the positions of all evacuees and convert those positions onto a graph representing the building of interest. At regular fixed intervals, we compute the optimal evacuation prescription by running the LP model with the current state of the graph; we then run our CA simulation to predict how the evacuees will move according to crowd dynamics. We identify regions in our building where non-optimal movement is likely to occur. As evacuation planners, we would then place emergency personnel at these locations to guide evacuee flow in the optimal direction given by the LP. We test this workflow using the same CA model we describe below.

Before moving on to the details of our model, we briefly review previous literature on optimization and simulation-based approaches to evacuation in Section 2. We then describe the details of our problem formulation and assumptions underlying the proposed models in Section 3.

2 Prior Work on Evacuation Algorithms

The literature on evacuation algorithms largely encompasses two types of strategies in modeling: *prescriptive* algorithms which use optimization-based methods to determine the best routes for evacuation, and *descriptive* approaches which simulate crowd dynamics, psychology, and panic factors in evacuation situations. Such models give equations that model the dynamics of individual agents trying to evacuate an area using insights from psychology, physics, and

other areas. Hybrid optimization-simulation procedures also exist but are less common than the above methods. For example, Abdelghany et al. [3] searches for optimal plans, where the search is guided by simulation-based methods. We emphasize that this is different from our real-time approach, which appears to be unique.

2.1 Optimization-based Methods

Optimization-based methods are utilized to generate evacuation plans that determine where to allocate resources in a network representing the evacuation site, and the best way to direct evacuees through the network. Implicitly, such methods seek to optimize some objective function, such as the number of evacuees at some fixed time step, or perhaps the total time evacuees spend in dangerous areas [5, 9, 22]. Throughout this paper, we refer to such a plan as a *prescription*: a deterministic set of instructions for how to move evacuees optimally in order to maximize some objective function.

These optimization-based methods convert proposed dynamics for evacuee movement, or dynamics for the evolution of evacuee flows (groups of evacuees), into constraints that are inputted into an optimization formulation such as a LP or Linear Program (IP). For example, the widely-cited cell transmission model (CTM) of Daganzo [10] prescribes hydrodynamic equations for the movement of traffic, which Romanski [22] converts into discrete evolution equations that are then inputted as constraints into an IP solver. Such constraints are equations relating how flow evolves at each node in the network, or equations describing node and edge capacities reflecting the maximum number of a evacuees a physical location can support.

Often the resulting linear programs are huge in scale, and require advanced computational techniques to solve. In Romanski [22], methods to improve the efficiency of large IP or LP problems are used, particularly the Benders decomposition [23]. These computational considerations inform our use of the LP which we describe in 4.

2.2 Simulation-based Methods

In contrast to the prescriptive optimization-based models above, simulation-based methods offer a more descriptive insight into evacuation problems. The

review paper by Zheng et al. [26] surveys several main types of simulation methods, notably agent-based models, social force models, cellular automata models, and flow/fluid dynamics-based models.

On the most microscopic level, agent-based models (ABMs) represent the dynamics of individuals [20]. The goal of such models is to endow evacuees with idiosyncratic behaviors based on results from psychology, and to study the emergent behaviors in crowds of such individuals. These models allows for a very high level of flexibility in modeling the individual characteristics of an evacuee, such as their behavioral tendencies or their physical attributes. These can models can be used to model situations ranging from panicked escape from a fire to calmly leaving a classroom [26]. However, these models tend to be computationally intensive, since every evacuee must be represented in the model.

The modeling of psychological effects at the individual level is similar to the approach of so-called social force models, which describe the dynamics of individuals in terms of laws of motion, influenced by reasonable personal desires. For instance, Helbing and Molnar [14] models influences such as the desire to reach some destination, the desire for personal space, and the desire to stay close to friends and family.

At the coarsest level are simulations that model flows, in the sense of partial differential equations. The implicit assumption in these models is that crowds can be modeled by fluids and gases, and hence can be described by hydrodynamic equations such as Navier Stokes [6, 16].

In an intermediate position between ABMs and flow-based models are Cellular Automata (CA) models [8, 24]. CA models are more similar to ABMs and social force models in that they represent dynamics of the individual evacuee, but they usually do so in a discrete graphical representation of the space, simulating the evacuation in discrete time. These considerations make CA models ideal for computer simulations [26]. In modeling individual agents, CA models differ in how much “intelligence” they give to the typical agent. While some papers model agents as in a ABM or social force model, others prescribe very simple rules defining how the agents in a CA model behave [20]. For example, the primary mechanism of movement in Burstedde et al. [8] is a flow field that directs evacuees to a set of exits. At each timestep, the agents stochastically “flow" towards the exits, while also being affected by the movement of other individuals. It is on this model that we base our CA simulation.

3 Problem Formulation & Assumptions

We consider the problem of efficiently determining and prescribing optimal directions in evacuation scenarios. As a case study, we focus on the Louvre, for which we provide a discretized graph representation as shown in Figure 2. The full details of this discretization may be found in the appendix, but we obtain this discretization using the following simplifying assumptions:

- We assume that structurally the floor layouts for each floor of the Louvre are the same, except for the lower floor and the level above, both of which have landings inside the Louvre Pyramid.
- We assume that every part of the Louvre is accessible - including the areas greyed out in the interactive floor plans [1]. This is reasonable, given that in emergency situations previously unusable space might be utilized as refuge.
- We assume the Pyramid reception in the basement of the Louvre represents the main entrance and exit from the Louvre - this is an approximation based on the provided interactive floor plans [1], as well as further floor plans found in [18].
- Stairwells are roughly based on the layouts in [1] and [18], and elevators in particular have been omitted.

Using these assumptions, we obtain a graph representation of the Louvre with roughly 9500 nodes. We settled on this figure for computational reasons. We emphasize that this graph representation can be refined by planners to better approximate the nuances of the Louvre's floor plans, and in fact our model presented below applies to a general graph representation of any building. While we apply our stochastic CA model on the graph visualized below, running our LP would be too costly on a graph of this size, so we discretize the Louvre further to obtain a graph of size 670 nodes, and set up the LP on this coarse graph. In the fine-grained representation of the Louvre, we assume that each node in the graph can hold up to one person in our CA simulation.

In our investigation, we assume that it would be unreasonable and practically impossible to provide individualized directions for each museum visitor, so we instead aim to determine the optimal locations of emergency personnel and the directions in which they should route nearby civilians. We measure optimality in terms of the number of evacuated civilians over time. We also make the following simplifying assumptions:

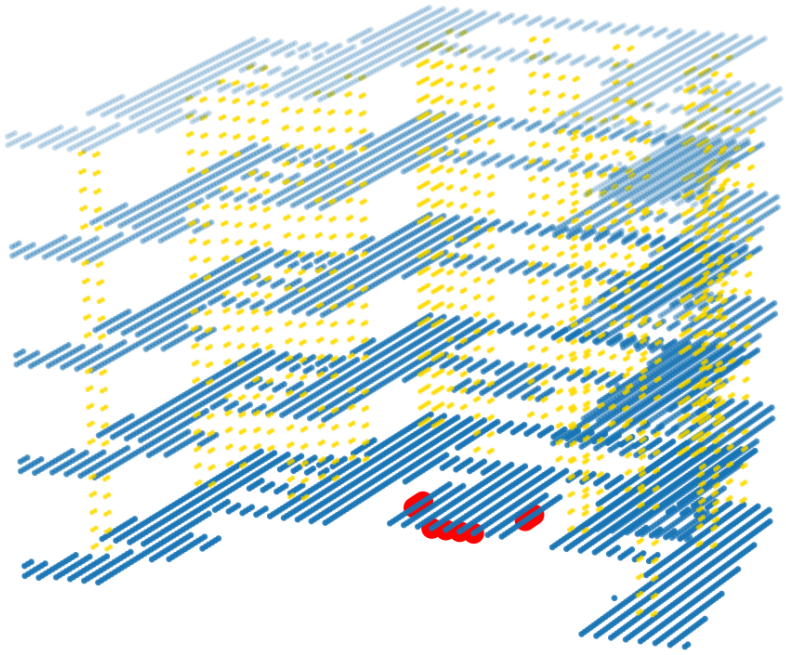

Figure 2: Fine graph representation of the Louvre, with stairwells and elevators in yellow and exits in red, for use in the simulation-based model. For our CA model, each node in the graph can hold at most one person.

- We model each evacuee homogeneously, assuming that the movement characteristics of each person are the same. We acknowledge that there is a large literature on this topic - there are agent-based models in particular model movement characteristics as a function of physical traits [26]. An interesting case is the modeling of handicapped individuals, which has been addressed in [11]. We do outline how to model handicapped individuals further in the paper.
- Threats and emergency personnel are regarded as static entities. However, in Section 4.4, we sketch an optimization method which accounts for emergency personnel movement.

- Threats reduce flow potential. In the simulation-based model, threats may also eliminate nearby civilians from the simulation.
- While threats may vary in number, they appear all at once.
- Approximate civilian and threat locations may be obtained through a variety of methods. For example, Bluetooth technology and the Internet of Things (IoT) may be useful in approximating civilian locations [25], and computer vision techniques may be used to approximate civilian populations and threat locations based on security footage. Given these capabilities, we assume that the rough locations of each evacuee are known in the CA model of Section 6.

Due to the wide range of potential disaster scenarios, these assumptions are clearly not representative of all evacuations. However, due to the modularity of our approach, additional factors may be easily added or removed at will.

4 Optimization with Linear Programming

We formulate evacuation flow on a graph representing building topology as a Linear Program (LP), following previous work in prescriptive analytics [5, 9]. The availability of efficient algorithms for solving LPs allows for evacuation flows to be updated in real-time, together with the flexibility for optimizing for an objective function (minimizing evacuation time, maximizing evacuated people, etc.) of choice. While all variables in our model are integers, we find that relaxing integrality constraints still gives an excellent approximation to the integral optima of the variables. Our model was implemented and solved using `Gurobi`, a commercial-grade LP solver [19].

We view solutions to our LP as an *optimal prescription* of an evacuation procedure. This, however, makes strong assumptions on the behaviour of each individual evacuee during the emergency. Crowds in emergency situations have been observed to make sub-optimal or irrational choices in states of panic, as we will discuss in Section 5. We incorporate these behavioral traits together with real-time of the optimal prescriptions provided by our LP in a cellular-automata simulation in Section 6.

4.1 Baseline Model

Let $G = (V, E)$ denote a graph with $V = \{1, \ldots, n\}$, and let $\{0, 1, \ldots, T\}$ denote discrete timesteps during the evacuation process. We modify G by adding a new node, node 0, depicting the outside of the building (or more generally any safe node), adding edges as appropriate.

In addition, we denote:

- the number of occupants of node i at time t by n_i^t
- the flow along edge (i, j) in the time-step $t \to (t+1)$, i.e. the number of people moving from node i to node j in $t \to (t+1)$, by $x_{i,j}^t$
- the node capacity, i.e. the maximum number of occupants of node i by c_i
- the flow capacity, i.e. the maximum number of people that can pass from node i to node j via edge (i, j) by $R_{i,j}$

We declare the variables n_i^t and $x_{i,j}^t$ to be integers subject to the following constraints:

$$n_i^{t+1} - \left(n_i^t + \sum_{(i,j)\in E} x_{i,j}^t - \sum_{(j,i)\in E} x_{j,i}^t \right) = 0 \qquad i \in V, t \in \{0, 1, \ldots, T\} \tag{1}$$

$$x_{i,j}^t + x_{j,i}^t \leq R_{i,j} \qquad (i, j) \in E, t \in \{0, 1, \ldots, T\} \tag{2}$$

$$n_i^t \leq c_i \qquad i \in V, t \in \{0, 1, \ldots, T\} \tag{3}$$

$$-x_{i,j}^t \leq 0 \qquad (i, j) \in E, t \in \{0, 1, \ldots, T\} \tag{4}$$

$$-n_i^t \leq 0 \qquad i \in V, t \in \{0, 1, \ldots, T\} \tag{5}$$

Constraint (1) enforces *conservation of flow*. Informally, it ensures that the number of people at node i at time $(t+1)$ is equal to the number of people at node i at time t minus the number of people leaving node i plus the number of people entering node i in the timestep $t \to (t+1)$.

Constraints (2) and (3) ensure that at no time-step during the evacuation process are the node and flow capacities violated. Finally, constraints (4) and (5) ensure non-negativity of node occupancy and edge flow.

4.2 Objective Function for Evacuation

Maximizing the number of people evacuated from the building in time T can be posed as the following optimization problem:

$$\max\ y_0^T$$

subject to the constraints described in Section 4.1. We can compute the minimum time required to evacuate all building occupants by varying T using binary search and solving the above LP for each T. This allows us to compute the Pareto boundary of the conflicting objectives max y_0^T and min T, as suggested in [5].

4.3 Modeling Adversaries

Adversaries are modeled as static obstacles to evacuee movement, decreasing the cell capacities of nodes they occupy. We note that the LP does not allow for movement of adversaries, nor does it allow for interaction between adversaries and evacuees. Both of these, however, can be incorporated into our model by means of a Stochastic Program (SP). In our simulations in Section 6, we set the cell capacity of a node occupied by an adversary (in our case an armed attacker) to be 0, reflecting the fact that all occupants of said cell will be eliminated by the adversary.

4.4 Optimization Extensions

Our LP formulation of an evacuation protocol allows for tremendous flexibility. For example, accessible evacuations for limited-mobility civilians can be modeled in the form of additional variables modeling movement, together with altered cell and edge capacities for the same. Specifically: we might create new capacity variables, with zero capacity on staircases. We would then ensure that the sum of all capacity variables remained under the original boundary.

Another possible extension involves modeling the movement of security personnel. One approach would involve treating security personnel as "opposite adversaries," increasing capacity in their vicinity, or increasing the capacity of hidden exits. These examples provide a brief insight into the diverse range of behaviors that optimization-based approaches may represent. Further, we suggest that this model may easily be extended to weather-related evacuations, traffic flow, and other scenarios.

5 Behavioral Patterns

While Section 4 demonstrated how evacuation may be modeled as an optimization problem, it is largely unreasonable to expect optimal evacuation behavior from frightened civilians. For example, Pelechano and Malkawi [20] mention that evacuees tend to proceed toward main entrances, often ignoring closer and potentially safer routes. In this section, we briefly review literature describing human behavioral patterns in these high-intensity scenarios. The observations made in this section will inform our simulation-based model in Section 6.

Although we primarily focus on large-building evacuation in this paper, we note that evacuation behavior may span many domains, including coastline evacuations, highway evacuations, and more. As discussed in Aguirre [4], patterns of evacuation are often similar across domains, thereby justifying our use of domain-general evacuation algorithms.

Helbing et al. [15] describes what is now a well-established "social force model" of evacuation behavior. Under this model, path clogging leads to dis-coordination, and fast movement may lead to overall slower evacuation times (due to increased clogging). This model also postulates that mass behavior is defined as some mixture of individual directions $\mathbf{e}_i$ and average directions $\langle \mathbf{e}_j^0(t) \rangle_i$ of its j nearest neighbors as follows:

$$\mathbf{e}_i^0(t) = \frac{1}{Z}\left[(1-p_i)\mathbf{e}_i + p_i \langle \mathbf{e}_j^0(t) \rangle_i\right] \tag{6}$$

where Z represents a normalization factor. The linear combination shown above captures the intuition that while humans will naturally move towards exits, they will be influenced by the recent behavior of those around them. This social force model motivates the widely-adopted cellular automata model of Burstedde et al. [8], which we implement in Section 6. We further extend this model to represent other behavioral factors, such as adversary avoidance.

6 Simulation with Cellular Automata

Our stochastic CA model closely follows the widely-cited model suggested by Burstedde et al. [8], with influences from social force models such as Helbing and Molnar [14].

6.1 Model Specification

The CA model represents the building of interest by a graph representation of the building—in the case of the Louvre, we utilize the fine-grained representation in Figure 2. Throughout this section, we represent our network representation as a graph G and the set of evacuees as a set of particles $\mathscr{E}$ on G. To distinguish between the fine-grained network of this CA model and the coarse representation utilized by the LP, we refer to both respectively as G_f, G_c,. When clear from context, we will write G for G_f. Both G_f and G_c are embedded in $\mathbb{R}^3$ using approximate coordinates obtained from Louvre floor plans in Figure 7, allowing us to directly compare evacuee behavior on both the representations.

At each timestep, each evacuee is assigned a probability distribution of moving to its neighbors, i.e. for every evacuee $e \in \mathscr{E}$ at node v at time t, we have a probability distribution $p_e^t \in \mathscr{P}(\mathscr{N}(v))$ where $\mathscr{N}(v)$ is the set of nodes adjacent to v. At each timestep, for every evacuee $e \in \mathscr{E}$, we move e according to its probability distribution.

The dynamics of the CA model are dictated by two objects, a *static floor field* S and a *dynamic floor field* D, which model how the probability distributions $\{p_e^t\}_{e\in\mathscr{E}}$ evolve to obtain $\{p_e^{t+1}\}_{e\in\mathscr{E}}$. Informally, the static floor field S models the tendency for the evacuees to route to the main/closest exit, whereas the dynamic floor field D models influence by the crowd and emergency personnel on evacuee movement.

We model S as a function on the nodes of G taking the value of the negative of shortest distance between the node and any exit in our graph. An evacuee, then, naturally moves in the direction in which S increases.

The dynamic floor field D evolves with time, and its value at node v at time $t+1$, written $D_{t+1}(v)$, is given by:

$$D_{t+1}(v) = d \times (D_t(v) + \text{flow}_t(v))$$

where d is a *decay* rate, and flow(v) is ± 1 representing whether or not the node v goes from being unoccupied to occupied in the timestep $t \to (t+1)$.

Following [8], we implement a temperature parameter controlling the influence of each floor field over evacuee movement. At lower temperatures, the static field exerts greater influence, whereas at higher temperatures the influence of the dynamic field is greater. Informally, low temperatures can be thought to represent smooth evacuee flow, whereas high temperatures capture crowd dynamics, or erratic, turbulent behavior.

Both floor fields S and D are dynamic in the presence of adversaries and obstacles, as well as emergency personnel. Emergency personnel are capable of influencing the dynamic floor fields of evacuees close to them in a given direction, and adversaries are modeled as static locations on the graph which endanger evacuees. In the case that the evacuee is within some chosen distance threshold of the adversary, the evacuee is eliminated.

6.2 Generating Prescriptions with LP Optimization

As mentioned previously, our model dynamically gives LP prescriptions for the optimal movement of evacuees, which can be relayed to emergency personnel, who then direct evacuees accordingly. At some later timestep, the LP prescriptions are generated again for the current state of the graph, and the updated directions are relayed to the personnel, who then redirect the evacuees according to the updated LP prescriptions. The rough workflow is given in Figure 1.

We now describe how this workflow is modeled in further detail. Since our LP cannot be run on a graph as large as G_f, we run the LP on G_c and convert between the two graphs using a nearest point mapping: nodes in G_f are mapped to the closest node in G_c when we convert information. This is achieved by overlaying the two graphs, visualized in Figure 7. Accounting for this, our model usage is as follows:

1. At a fixed time on our network G_f, we convert the information in G_c to obtain the numbers of evacuees at each node in G_c.
2. The LP model is run using the data from step 1 as an initial distribution. The LP model will then optimize the number of evacuees who have escaped by time τ_{LP}.
3. The solution of the LP contains information for the optimal locations and sizes of evacuee flow. We take the directions of the flow at the initial timesteps and convert this back to directions on G_f.
4. In parallel to solving the LP, we simulate the stochastic CA model above for τ_{cycle} timesteps. In a real evacuation procedure, this predicts the future evolution of evacuees. We then identify the nodes $v_c \in G_c$ whose flows in the simulation behave maximally differently from the direction given by the LP. We quantify this error and identify the top N_{EP} such nodes.
5. We then place emergency personnel at these nodes, who influence evacuees to move in the direction prescribed by the output of the LP.

6. Run this loop again, after waiting for τ_{cycle} timesteps.

This entire workflow is simulated using the same stochastic CA model to analyze how our model works in practice. We elaborate on the 3rd, 4th, and 5th points above.

For the third point, at a time t, for every node $v \in G_c$, we consider the vector given by the weighted sum of its neighbors. That is, let $v_i \in \mathbb{R}^3$ be the neighbors of node v. Once the LP is solved, we obtain the flow values at time 0, $x^0_{i,j}$, for all $j \in \mathcal{N}(v)$. Then we consider the vector:

$$V^t_v = \sum_{j \in \mathcal{N}(v)} x^0_{i,j} v_i,$$

which we denote as the coarse LP flow direction; we interpret this as the LP's prescription for how the flow of evacuees should evolve.

To identify problem areas where the flow in G_f runs counter to the prescription V^t_v for $v \in G_c$, we quantify the error in the following way. Let $Loc_{G_f}(v)$ be the locus of points in G_f that are closest to $v \in G_c$. We imagine that the coarse LP flow direction is being broadcast to every node in $Loc_{G_f}(v)$ by an emergency personnel; in this case, the static floor field would change, affecting the movement probability distributions of every evacuee e in $Loc_{G_f}(v)$. This update is given by

$$p_{new,e}(x) = \frac{1}{Z} p_{old,e}(x) \exp(V^t_v \cdot (x - v)),$$

where x is a neighbor of e in coordinates. This update has the effect of weighting the new probability distribution in the direction of the coarse LP flow direction. Using this, we can quantify the error of the node v: how badly are the evacuees in $Loc_{G_f}(v)$ deviating from the LP prescription? For each evacuee $e \in Loc_{G_f}(v)$, we calculate the difference of $p_{new,e}, p_{old,e}$ in the Euclidean norm. We then sum over these errors to obtain a measure $Err(v)$. After obtaining this error measure for every node in G_c, we identify the top N_{EP} nodes, corresponding to the number of personnel we have available.

Then, in the fifth step above, we place emergency personnel at each of these nodes. These will update the flows of every evacuee in $Loc_{G_f}(v)$ in the manner detailed above.

6.3 Results

While many questions could be asked about our model, we explore a few specific questions regarding how our model applies to buildings such as the Louvre:

- How does the number of hidden exits influence the evacuation?
- Where are the areas of the Louvre that impede evacuations?
- How does an evacuation improve with the number of emergency personnel guiding the evacuation?
- How does our model perform if we remove the prescriptions from the LP, and let the evacuees move on their own?
- How does evacuation time vary by the number of people in the Louvre?

We measure the progress of an evacuation using an *escape curve* as in [17], which plots the percentage of people who have evacuated against the running time of the simulation.

6.3.1 Evacuation Success vs. Number of Personnel

To explore how evacuations can improve given the number of emergency personnel guiding the evacuation, we generate the escape curves for the Louvre with 500 evacuees and two adversaries, with $N_{EP} = \{0, 100, 200\}$. Since G_c has roughly 700 nodes, the setting when $N_{EP} = 200$ roughly means that there are a significant number emergency personnel, spread throughout the Louvre. With $N_{EP} = 0$, we have a paucity of resources – no emergency personnel are stationed, which is equivalent to not prescribing directions according to the LP. For computational considerations, we run the workflow for $N = 20$ loops, with $\tau_{cycle} = 15$ timesteps. We do not add any hidden exits beyond the ones pictured in Figure 2. The results are shown below in Figure 4. Not only do we see an improvement in performance as the number of emergency personnel is increased, but we see a clear demarcation between the evacuation performance when we implement our method, that is, upon directing personnel to the most needed nodes, we do better than just letting the evacuee crowd move by itself. This shows success of our prescriptive solution over a non-prescriptive one.

Note that these results are calculated with $\beta = 2.0$, where β is a temperature parameter representing the relative effect of crowd dynamics to main exit flow. This parameter is analogous to the p_i linear combination term in (6) above. For low values of β, emergency prescriptions appear less useful; we may therefore interpret prescribed directions to encourage movement along the correct path, rather than in the direction of mass transit.

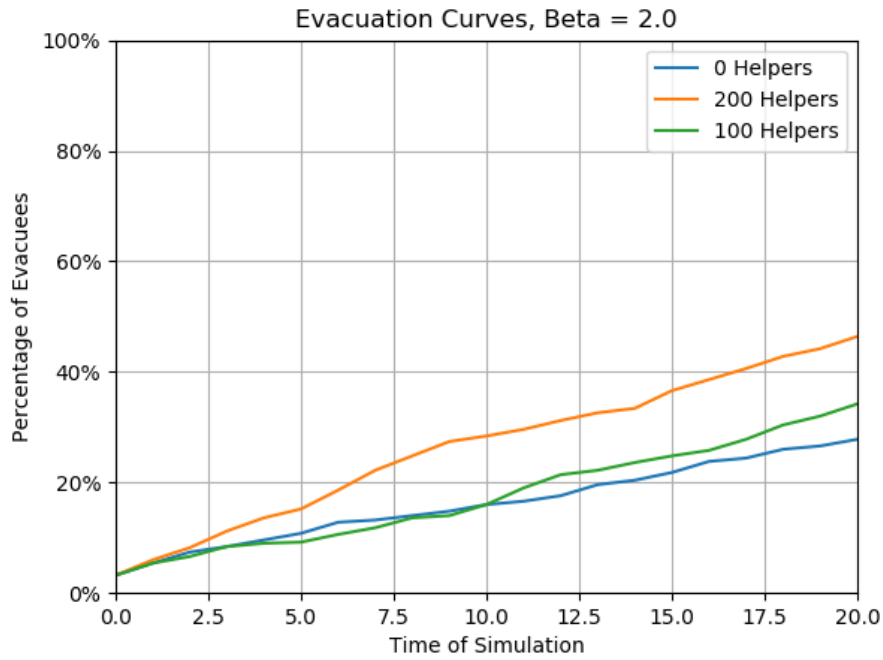

Figure 3: Evacuation curves, for 500 people, as the number of emergency personnel increases. We set values of N_{EP} at $0, 100, 200$.

6.3.2 Evacuation Success vs. Number of Hidden Exits

To answer this question, we simulate 3 situations with 0, 5, and 10 hidden exits each. In each of these situations, we pick a random set of vertices from G_f to serve as hidden exits; in practice these can be set to reflect the actual architecture of the building. We set our simulation parameters to have 500 evacuees, 2 adversaries, and $N_{EP} = 100$.

# Hidden Exits	0		5		10	
Time	Saved	Killed	Saved	Killed	Saved	Killed
0	0	0	0	0	0	0
3	60	11	229	11	272	11
6	133	12	348	11	407	11
9	219	12	387	11	468	11
12	299	12	424	11	475	11
15	358	12	453	11	478	11

Table 1: Evacuation Success vs. Number of Hidden Exits

Based on these results, we recommend that the Louvre open up hidden exits in the course of any evacuation procedure, as they offer an over 33% increase in the number of saved evacuees even with 10 extra exits.

6.4 Identifying Bottlenecks

We identify bottlenecks using established graph theoretic values, known as betweenness and closeness centralities [7]. These measures can be computed in Networkx [12]. These centrality measures represent how “in between" a node on the graph based on shortest paths between nodes, as well as shortest paths on electrical networkx. We use these measures to calculate regions on our graph that our the most central. The measures are visualized below, in Figure 4. Based on both measures, we can see that the wings of the Louvre are less central, whereas the parts of the Louvre in between the wings are the most central. In addition, we see in the betweenness measure that the nodes where staircases join with the graph are also declared central.

7 Conclusions & Future Work

We introduced an online dynamic evacuation method that combines the strengths of both optimization-based prescriptive methods and simulation-based procedures. After discretizing the Louvre into a fine-grained graph model, we use LP to compute optimal evacuation plans, and we use a stochastic CA model to predict crowd behavior. These two elements are combined in order to pinpoint locations at which evacuee crowds behave sub-optimally as compared to the LP’s

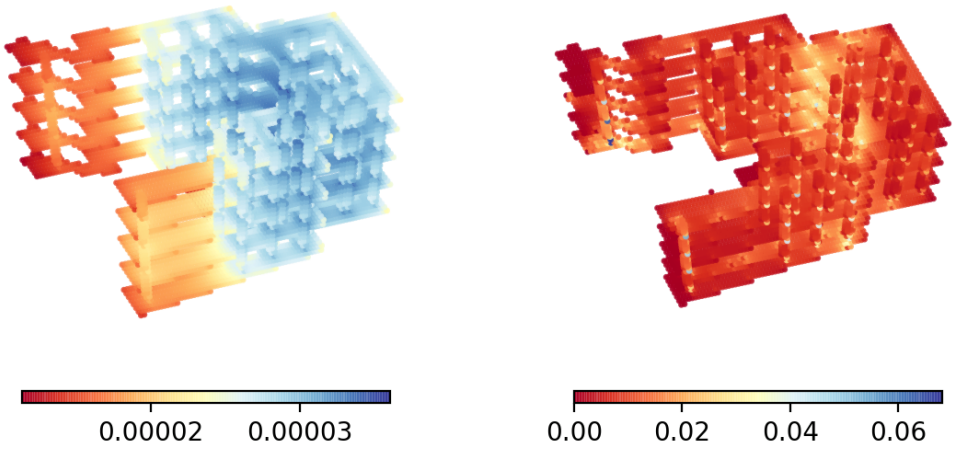

Figure 4: Current flow closeness (left) and betweenness (right) centrality measures on the fine-grained representation of the Louvre.

prescriptions. We then place emergency personnel to help guide the flow of the evacuee crowds at these locations.

We used our model to test several questions about the evacuation plans in the Louvre; in particular, how do the evacuation curves change when increasing the number of helpers, and how does performance change as a function of the number of hidden exits? We observed that an increased number of exits leads to significantly faster evacuations, and we visualized bottlenecks according to closeness and betweenness centrality measures. We also showed that increasing the number of emergency personnel - the parameter N_{EP} - improves evacuation curves with 500 people at high D field temperatures. This implies that

We believe that the relationship between evacuation success and the number and locations of adversaries is important. We imagine simulations fixing obstacles at key areas of the Louvre and seeing the impact on evacuation time and other metrics. We leave this to future work.

We modeled adversaries, terrorists, and obstacles as static; in practice, however, our model is dynamic, and the locations of threats can be updated whenever the models are rerun. This workflow allows for the extremely flexible modeling of potential threats and emergency personnel in real time. The parameters of our model can be updated to reflect the usage cases needed by building and evacuation planners – this may include placing the specific locations of known

exits in the Louvre, as well as inputting the given number of available emergency staff and even the severity of the threats in an evacuation.

For computational efficiency reasons, we made a number of simplifying assumptions regarding the topology of the Louvre as well as the scale of the evacuation. In a realistic implementation of our procedure, of course, one could model this network as finely as desired.

A Appendix: Modelling the Louvre

Scanned floor layouts from the official Louvre Travel Guide Lee [18] were masked and then converted to a graph using the `Image Graphs` scripts provided by Steve Eddins for MATLAB.

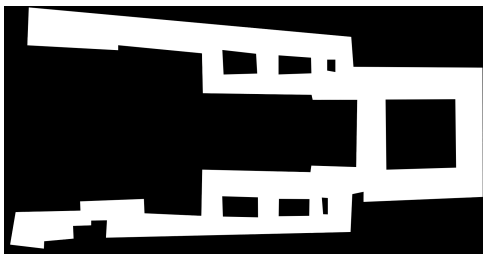

Figure 5: Masked Representation of Floor 2 of the Louvre

The coarse graph used for optimization in 4 consisted of approximately 669 nodes, whereas the fine graph used for the CA simulation in 6 consisted of 9574 nodes.

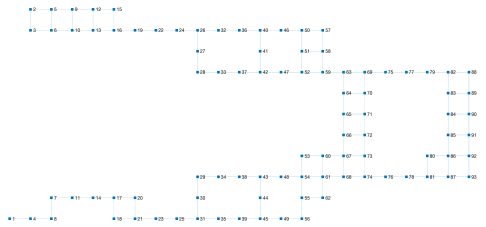

Figure 6: Coarse Representation of Floor 2 of the Louvre

Staircases between floors are added as edges between the corresponding graphs. Both the coarse and the fine representations of the Louvre were embedded in the same coordinates, with a hash map between the vertex sets of both representations. A simple graphical representation of the overlaid graphs is given below.

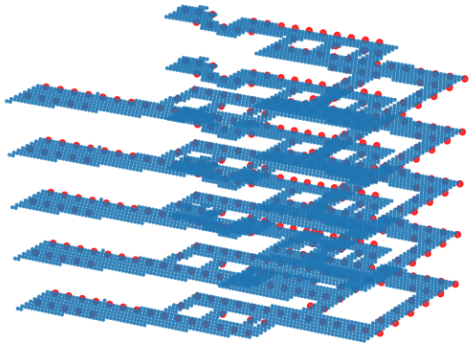

Figure 7: Overlay of the fine-grained Louvre graph, in blue, and the coarse-grained Louvre graph, in red. The nodes in the fine grained representation are mapped to the closest nodes in the coarse-grained graph.

B References

[1] Interactive floor plans. URL `https://www.louvre.fr/en/plan`. Accessed: 2019-01-27.

[2] Terror attacks in France: From Toulouse to the Louvre. Telegraph Media Group, Feb 2017. URL `https://www.telegraph.co.uk/news/0/terror-attacks-france-toulouse-louvre/`.

[3] Ahmed Abdelghany, Khaled Abdelghany, Hani Mahmassani, Hassan Al-Ahmadi, and Wael Halabi. Modeling the evacuation of large-scale crowded pedestrian facilities. *Transportation Research Record Journal of the Transportation Research Board*, 2198:152, 12 2010. doi: 10.3141/2198-17.

[4] Benigno E Aguirre. Emergency evacuations, panic, and social psychology. *Psychiatry: Interpersonal and Biological Processes*, 68(2):121–129, 2005.

[5] Claudio Arbib, Henry Muccini, and Mahyar T. Moghaddam. Applying a network flow model to quick and safe evacuation of people from a building: a real case. 09 2018.

[6] GE Bradley. A proposed mathematical model for computer prediction of crowd movements and their associated risks. *Engineering for crowd safety. Amsterdam: Elsevier*, pages 303–11, 1993.

[7] Ulrik Brandes and Daniel Fleischer. Centrality measures based on current flow. In *Annual symposium on theoretical aspects of computer science*, pages 533–544. Springer, 2005.

[8] Carsten Burstedde, Kai Klauck, Andreas Schadschneider, and Johannes Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3-4): 507–525, 2001.

[9] W Choi, H W. Hamacher, and Suleyman Tufekci. Modeling of building evacuation problems by network flows with side constraints. *European Journal of Operational Research*, 35:98–110, 02 1988. doi: 10.1016/0377-2217(88)90382-7.

[10] Carlos F Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994.

[11] Nirdosh Gaire. *A Study on Human Evacuation Behavior Involving Individuals with Disabilities in a Building*. PhD thesis, 07 2017.

[12] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[13] Dirk Helbing and Bernardo A Huberman. Coherent moving states in highway traffic. *Nature*, 396(6713):738, 1998.

[14] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[15] Dirk Helbing, Illés Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487, 2000.

[16] LF Henderson. The statistics of crowd fluids. *nature*, 229(5284):381, 1971.

[17] Gregor Lämmel, Hubert Klüpfel, and Kai Nagel. *The MATSim Network Flow Model for Traffic Simulation Adapted to Large-Scale Emergency Egress*

and an Application to the Evacuation of the Indonesian City of Padang in Case of a Tsunami Warning, pages 245–265. 11 2009. doi: 10.1108/9781848557512-011.

[18] Cecil Lee. Louvre museum guide download, 2010. URL `https://www.travelfeeder.com/travel_tips/louvre-museum-visitors-guide-download`.

[19] GuRoBi Optimization. *GuroBi Optimizer Quick Start Guide*, 2018.

[20] Nuria Pelechano and Ali Malkawi. Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. *Automation in construction*, 17(4):377–385, 2008.

[21] Nuria Pelechano, Jan M Allbeck, and Norman I Badler. Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108. Eurographics Association, 2007.

[22] Julia Romanski. *Algorithms for Large-Scale Prescriptive Evacuations*. PhD thesis, 2016.

[23] Julia Romanski and Pascal Van Hentenryck. Benders decomposition for large-scale prescriptive evacuations. In *AAAI*, pages 3894–3900, 2016.

[24] A Varas, MD Cornejo, D Mainemer, B Toledo, José Rogan, V Munoz, and JA Valdivia. Cellular automaton model for evacuation process with obstacles. *Physica A: Statistical Mechanics and its Applications*, 382(2):631–642, 2007.

[25] Yuji Yoshimura, Stanislav Sobolevsky, Carlo Ratti, Fabien Girardin, Juan Pablo Carrascal, Josep Blat, and Roberta Sinatra. An analysis of visitors' behavior in the louvre museum: A study using bluetooth data. *Environment and Planning B: Planning and Design*, 41(6):1113–1131, 2014.

[26] Xiaoping Zheng, Tingkuan Zhong, and Mengting Liu. Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment*, 44(3):437–445, 2009.