Introduction

Methods

Orientation

The grid in this model is a rectangular lattice of square cells where the agents can move from one cell to another in 8 directions of Moore neighbourhood. The agents have four orientations: North represents the top of the grid and South the bottom, East is the right-hand side of the grid and West is the left-hand side. The orientations are perpendicular one to each other. Even though the movement is possible in 8 directions the agent has only 4 orientations because of the nature of paired agents. Two agents in a pair have the same orientation and are located in adjacent cells.

Directed agents

Agents can move to cells in Moore neighbourhood in 8 directions or stay in the same cell. Directed agents take into consideration the direction of the movement. The agents move in a discretized rectangular grid. The agent has 4 possible orientations - North, East, South, West - which are global and not relative to the agent itself. This means that agent which has orientation East is directed to the right-hand side of the grid.

Moore neighbourhood allows movement in 4 non-diagonal and 4 diagonal directions. In case of the non-diagonal directions, the agent orientation after the move is the same as the direction e.g. agent with orientation North moving to adjacent cell on the right will change orientation to East. In case of diagonal directions there are 2 types of movements: the first is movement to diagonal cells where the steering angle is +-45 degrees and the second is movement to diagonal cells where the steering angle is +-135 degrees. In the case of the first movement the agents keeps his orientation unchanged, e.g. agent facing West moves to diagonal upper left cell will have West orientation after the movement. In the case of the second movement the agent changes the orientation to the opposite orientation e.g. North to South and vice versa, East to West and vice versa.

Partner agents

Partner agents form a pair of two directed agents that are tightly bound so that they are in adjacent cells to each other. Two partner agents in the pair cannot be in cells diagonal one to each other. Their movement is synchronous, which means that if any of the agent is not able to move to the desired cell (conflict, bound agent did not move), the partner agent will abort its movement. The algorithm which checks whether both agents will successfully move is described below. In this model, the two agents in the pair are in a hierarchy of one agent being the leader while the second agent is not. The leader is responsible for calculating the probabilities of movement to cells for itself and it's partner according to maneuvers. The leader is the agent which has it's partner on the right. The partner agents are directed agents and both have the same orientation. Note that some maneuvers change the leadership in the pair.

Pair formation

The children in pre-school age are being taught to form a pair and hold hands when walking through corridor or crossing a road as these situations pose risks such as getting lost or encountering traffic. The model in this thesis simulates the coordinated evacuation with supervisor where a risk element is present.

The pupils are located in a classroom in a cluster where some pupils are close to each other and others are more isolated. The model does not consider any friendship preferences between children and assumes that pupils close to each other are more likely to form a pairs. It also assumes that pupils form pairs in group of even number of pupils so that there are no solitary children. If a pupil can't form pair immediately it will do so when other solitary pupil is nearby.

The Algorithm 1 below finds a way to form pairs of pupils which are not yet in pair.

Algorithm 1 Finding pairs

1: $G = (V, E)$
2: **while** $\exists v \in V : d(v) > 1$ **do**
3: Let $v^* \in V : d(v^*) = \max_{u \in V} d(u)$.
4: Let $w \in V : d(w) = \max_{(v^*,w) \in E} d(w)$.
5: Remove (v^*, w).
6: **end while**
7: **for** $(v, w) \in E$ **do**
8: `formPair(v,w)`
9: **end for**

Locations of pupils (directed agents) not in pair are transformed to a graph. The vertices of the graph correspond to the pupils' locations, and an edge is formed between two vertices in adjacent cells. The vertices in the graph may have different degrees, and cycles may be present. The algorithm iteratively selects the vertex with the highest degree and removes the edge connecting it to the vertex with the highest degree until all vertices have at most one edge. The vertices connected by an edge represent a pair.

Maneuvers

Each agent in a pair has the potential to move to 8 cells in the Moore neighbourhood, allowing for a vast number of possible maneuvers. However, most of these maneuvers do not maintain the structure of the pair and are therefore prohibited. Specifically, only 18 viable maneuvers are allowed for each orientation of the paired agents.

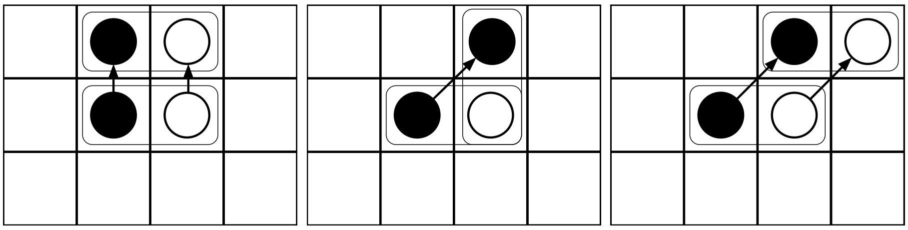

Figure 0.1: Simple maneuvers of paired agents.

Some maneuvers can alter the leadership of the pair or the orientation of the agents. Additionally, in some cases, the agents may have different movement speeds. During the synchronous atomic movement of a maneuver, both agents may move to diagonal cells, or one agent may move to a diagonal cell while the other remains in its current cell. Alternatively, one agent may even move outside its Moore neighbourhood in order to preserve the structure of the pair. In the Figure 0.2 can be seen the notable maneuvers.

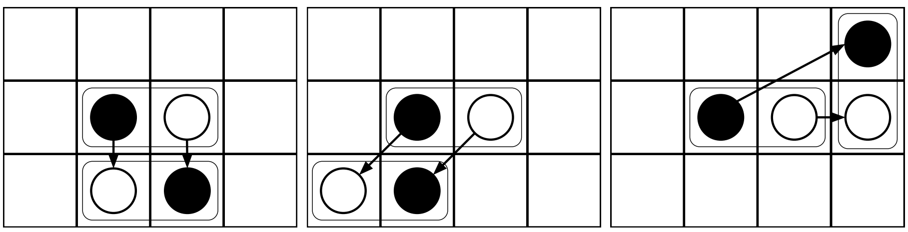

Figure 0.2: Complex maneuvers of paired agents.

Static floor field

Each cell in the grid in the static floor field *SFF* holds the value of shortest distance to the goal. The value is computed using BFS algorithm which allows diagonal movement. Alternatively the distance can be computed by novel approximate algorithm described in [1]. The leader has full information about the topology of the map and he moves towards the closest goal. The follower agents, solitary or in pairs, follow the leader and they do not attempt to find their own way to the exit. The leader has computes SFF for the follower agents where the goal is the leaders position.

Leader

The solitary agent responsible for navigation is the leader agent, which has complete information about the map topology and goals.

The leader agent is not directed and can move to cells in the Moore neighborhood based on the SFF of the current goal. The virtual leader is a leader agent, which does not occupy a cell and is used to set SFF for follower agents. The SFF for the leader agent differs from the SFF of follower agents. The SFF for follower agents is calculated in every step based on the virtual leader's position. The virtual agent navigates the followers when the leader moves to the end of the crowd.

Agent's proximity to the leader (not virtual leader) proportionally increases the static potential value:

$$d = \mathrm{distance}(\mathrm{leader}, \mathrm{follower})$$
$$S = S * (1 + \frac{1}{d})$$

With a higher static potential value, the agent is less likely to deviate from the optimal trajectory set by the SFF of the virtual leader. This method is described in more detail in Section **??**. The leader has simple rules for navigating towards the closest goal and checking if all follower agents reached the goal. One rule is the leader's ability to command the follower agents to continue to the goal while the leader moves to the most distant agent. Additionally, the (virtual) leader agent waits near the goal area and attracts the follower agents until they all reach it. The solitary agent responsible for navigation is the leader agent. In every step, the SFF for follower agents is calculated based on the leader's or the virtual leader's position. The leader has simple rules for navigating towards the closest goal and checking if all follower agents reached the goal. One rule is the leader's ability to move to the distant end of the cluster to speed up follower agents left behind. Additionally, the leader agent waits near the goal area and attracts the follower agents with its SFF until they all reach this area.

Attraction

The selection of the next cell by an agent is influenced by both the agent's own state, including sensitivity parameters, timestep, and partner agent, as well as the state of the agent's surroundings, such as the SFF of the cell, distance to the leader, occupancy of the cell, and obstacles in the corner. The agent computes the attraction of each cell in its surroundings using a mixed strategy based on the method proposed by Šutý [2]. The attraction of each cell is then normalized across all cells to compute the probability of selecting a particular cell from the set. The probability of an agent moving from cell x to cell y, denoted by $P(y \leftarrow x \mid N)$, is calculated based on two members, P_O and P_S.

$$P(y \leftarrow x \mid N) = k_O P_O(y) + (1 - k_O) P_S(y) \tag{1}$$

Specifically, $P_O(y)$ takes into account the occupancy of cell y and returns a normalized value in the range $[0, 1]$. On the other hand, $P_S(y)$ considers the static potential of cell y and guides the agent towards the exit, along with the diagonal motion indicator $D(y)$. Both P_O and P_S are normalized across neighboring cells from N.

$$P_O(y) = \frac{\exp(-k_S S(y))(1 - O(y))(1 - k_D D(y))}{\sum_{z \in N} \exp(-k_S S(z))(1 - O(z)(1 - k_D D(z))} \tag{2}$$

$$P_S(y) = \frac{\exp(-k_S S(y))(1 - k_D D(y))}{\sum_{z \in N} \exp(-k_S S(z))(1 - k_D D(z))} \tag{3}$$

The equation for computing cell attraction serves as the baseline for solitary agents moving to Moore neighborhood. When paired partner agents are involved, each agent computes the attraction of both cells in each maneuver. For instance, agent a_1 in cell c_1, paired with partner agent a_2 in cell c_2, computes the attraction of maneuver $m_1 \in M$ in the following manner:

$A_1 = A(a_1 \rightarrow c_{1*})$ and $A_2 = A(a_2 \rightarrow c_{2*})$, where c_{1*} and c_{2*} represent the cells after the maneuver. The attraction of maneuver m_1 is then determined as $A(m_1) = \min(A_1, A_2)$, and the probability of selecting maneuver m_1 by agents a_1 and a_2 is given by $P(m_1 | a_1, a_2) = \frac{A(m_1)}{\sum_{m_i \in M} A(m_i)}$.

Penalization and discipline

The behavior of children is known to be strongly influenced by the presence of authority figures, such as teachers, parents, or other responsible adults. In the proposed model, the leader assumes the role of authority and guides the children as they move through the environment. Specifically, children who are in close proximity to the leader exhibit higher levels of discipline and move in a more orderly manner, preserving the structure of the queue of pairs and following the optimal path as determined by the social force field (SFF). To promote this behavior, the static force value assigned to each agent is adjusted according to their proximity to the leader. In particular, as the distance d between an agent and the leader decreases, the static force value S assigned to that agent is multiplied by a factor of $(1 + \frac{1}{d})$. By moving back and forth within the queue, the leader can selectively increase the discipline of agents in their immediate vicinity and thereby exert greater control over the overall behavior of the group.

Discipline - distance to leader Change of orientation penalization Obstacle crossing penalization

Todo:

Incorrect change of orientation penalization adaptive time span crossing the corner numerical exponents with offset ability to pass tight space conflicts stochasticity strategy virtual leader topology

Maybe:

proximity to leader increases discipline of the agents (lower probability of error step) lonely agents try to pair?? ability to pass tight space - decouple pairs?