

Lecture with Computer Exercises: Modelling and Simulating Social Systems with MATLAB

Project Report

Train Boarding Platform Simulation

Daniel Graf & Matthias Krebs

Zurich December 2010

Eigenständigkeitserklärung

Hiermit erklären wir, dass wir diese Gruppenarbeit selbständig verfasst haben, keine anderen als die angegebenen Quellen-Hilfsmittel verwendet haben und alle Stellen, die wörtlich oder sinngemäss aus veröffentlichen Schriften entnommen wurden, als solche kenntlich gemacht haben. Darüber hinaus erklären wir, dass diese Gruppenarbeit nicht, auch nicht auszugsweise, bereits für andere Prüfung ausgefertigt wurde.

Daniel Graf

Matthias Krebs

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Daniel Graf

Matthias Krebs

Contents

1	Individual contributions 6						
2	Introduction and Motivations						
3	Des	Description of the Model					
	3.1	Model Overview	8				
	3.2	Trains	9				
	3.3	Agents	9				
		3.3.1 Agent States	9				
		3.3.2 Door Selection	9				
		3.3.3 Groups	11				
	3.4	Doors	11				
	3.5	Obstacles	11				
	3.6	Dynamics	11				
	3.7	Forces	12				
		3.7.1 Doors	12				
		3.7.2 Obstacles	13				
		3.7.3 Agents	14				
	3.8	Simulated Situations	15				
		3.8.1 Two Trains	15				
		3.8.2 One Train	16				
Δ	4 Implementation						
т	4 1	Model Add-ons	17				
	т.1	4.1.1 Forces Correction	17				
	$4\ 2$	Initialization	10				
	4.3	Simulation	10				
	4.4	Undate state	20				
	1.1	4.4.1 Look for a seat	$\frac{20}{20}$				
	45	Undate strategy	$\frac{20}{20}$				
	4.6	Calculate forces	$\frac{20}{20}$				
	4.7	Move agent	20				
	4.1 1.8	Plot	21				
	4.9	Save Data	$\frac{21}{23}$				
۲	C:	ulation Deputs and Discussion	<u>م</u>				
Э	5 1	Conoral Results	24 94				
	0.1	5.1.1 Observations on the map	24 94				
		$0.1.1 0.0501 \text{ values on the map} \dots \dots$	24				

		5.1.2	Insights out of the graphs	26
	5.2	Simula	tion Variables and Result Indicators	33
	5.3	Test S	eries	35
		5.3.1	Number of Agents	35
		5.3.2	Door Decision Modes	36
		5.3.3	Laziness Coefficient Optimisation	38
		5.3.4	Door Decision Frequency	38
		5.3.5	Door Decision Limit	41
		5.3.6	Patience	41
		5.3.7	Velocity	42
		5.3.8	Groups	42
		5.3.9	Simulation Start relative to the Door Opening	43
		5.3.10	Waiting Area	44
6	Sun	mary	and Outlook	45
6	Sun	nmary	and Outlook	45
6 7	Sun Refe	nmary erences	and Outlook	45 47
6 7 8	Sun Refe App	nmary erence: pendix	and Outlook	45 47 48
6 7 8	Sum Refe App 8.1	nmary erence: oendix Link t	and Outlook	45 47 48 48
6 7 8	Sum Refe App 8.1 8.2	nmary erences pendix Link t Source	and Outlook	15 17 18 48 48
6 7 8	Sum Refe App 8.1 8.2	erences erences Dendix Link t Source 8.2.1	and Outlook	15 17 18 48 48 48 48
6 7 8	Sum Refe App 8.1 8.2	erences bendix Link t Source 8.2.1 8.2.2	and Outlook	45 47 48 48 48 48 52
6 7 8	Sum Refe Apr 8.1 8.2	erences bendix Link t Source 8.2.1 8.2.2 8.2.3	and Outlook and Outlook s and Outlook o further material and outlook bcode and outlook s and outlook <td> 45 47 48 48 48 48 52 74 </td>	 45 47 48 48 48 48 52 74
6 7 8	Sum Refe 8.1 8.2	erences bendix Link t Source 8.2.1 8.2.2 8.2.3 8.2.4	and Outlook 4 s 4 o further material 4 ecode 5 Starting points 5 Initializations 5 Simulation 5 Statistical evaluation 5	 45 47 48 48 48 52 74 84
6 7 8	Sum Refe 8.1 8.2	mary erences bendix Link t Source 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5	and Outlook 4 s 4 o further material 4 ecode 5 Starting points 5 Initializations 5 Simulation 5 Statistical evaluation 5	 45 47 48 48 48 52 74 84 85
6 7 8	Sum Refe 8.1 8.2	hmary erences bendix Link t Source 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5 8.2.6	and Outlook a s a o further material a bcode a bcode a Starting points a Initializations a Simulation a Statistical evaluation a Plotting a Saving and loading simulation data a	 45 47 48 48 48 52 74 84 85 91

1 Individual contributions

As we met and exchanged ideas several times per week, both of us have contributed to most parts of this project. We contributed equally to the model description, development, testing and final documentation. A specific separation can be made here:

Matthias Krebs was in charge of finding and implementing an appropriate forces model. He invested a lot of time to get the agents moving around smoothly and keeping always an appropriate distance to all obstacles and trains.

Daniel Graf designed the different test cases and executed them during several days on the remote workstations of D-ITET. He also did the statistical analysis of the collected data.

2 Introduction and Motivations

As our ways to the ETH include about two hours of travelling by train every day, we decided to simulate a specific situation, that we come across twice a day. When trying to board the train on a crowded platform, one can observe a special kind of bottleneck problem.

We often discussed different techniques to find and enter a free door as quickly as possible, so we wanted to simulate and analyze them using Mathworks *MATLAB* as our term project of the course *Lecture with Computer Exercises: Modelling and Simulating Social Systems with MATLAB*.

At first it was essential to take a look at the different factors influencing the behavior of a passenger:

- A single passenger normally just wants to get in as fast as possible. But if the next free door is too far away he might reconsider his choice. If possible, he decides for a door that takes him a little longer to get in, but shorter to walk to. This problem of the optimal doordecision was our main interest and is covered in full detail in the results section. It is of special interest to separate one's personal optimum from the global one. For the train crew it is important that *every* last passenger gets in as soon as possible, which does not have to correlate with the personal door decisions.
- Some travellers might have used this train so often before, that they are able to predict where the doors will arrive when the train stands still. That way they can reduce their queuing time and have a higher probability to discover an unassigned seat.

- Although it is hard to guess from outside the train, it is important to know how many seats are still empty in the preferred coach. If the platform is very crowded, passengers might need to take another door, which forces the train to wait a lot longer.
- In bad weather there is an additional factor to consider. The passengers want to wait under the roof of the platform just until short before the doors of the train open and the boarding can begin.
- There are sometimes groups, like forms or gym clubs, that prolong the boarding process additionally, as they have reserved seats in one coach and therefore want to enter all together through the same door.

The goals of this project are as follows:

- Simulate a big crowd of passengers in a well-proportioned scenario like in Zurich or Sargans.
- Find a model that describes the behavior of a passenger in terms of movement and deciding for a door.
- Study the effects of a variation of the different simulation parameters.

3 Description of the Model

3.1 Model Overview

We intended to find a suitable model that is general enough to handle a lot of trainspecific additions but also precise enough to simulate decisions and movements of individual passengers.

Many general models use analytical approaches like in [3] and are then able to solve wave equations exactly. But most of them are limited to really simple scenarios, like a semicircular crowd in [3]. As our setup with many doors and obstacles would be much too complex to be solved exactly we decided to do a time discrete simulation.

In previous work cellular automatons have also been used quite often, like in [1]. But as we wanted to have precise information about every agent (for distance measuring, obstacle interaction etc.), we decided to use an agent-based approach.

In order to have enough resolution to build doors, obstacles and let the agents move around them, a cellular automaton would have needed at least a level of detail of about one square meter per cell. With a simulation size of $2000m^2$ and a neighborhood diameter of 10m a cellular automaton would approximately need as much computation power as an agent based simulation with around 450 agents (asymptotic runtime of the agent based approach $\mathcal{O}(N^2)$, with N the number of the agents). So we preferred the agent based approach, because of its higher precision.

The door decision parameters are inspired from [2]. It gave us a really clean game theoretical approach that has all needed possibilities for us to build on (door availability, door familiarity and additional conditions). It is also claimed in this paper that iterating the decision process has led quickly to a Nash equilibrium. The scenario used there with two doors at opposed sides of a square was much simpler than our boarding platforms.

But also in our simulations with up to 20 doors of 7 different types the decision simulation stabilized quickly. As always when using iterative simplifications, there are probably some border cases, where the simulation would not stabilize. We think that optimal strategies for this game would probably be non-pure strategies. As all agents decide simultaneously it is really unlikely that a optimal strategy would not need any probabilistic decisions.

For simulating the moving behaviour of the agents we found a nice approach in [4] that allowed us to represent all interactions between agents, trains, obstacles and doors.

We have designed large parts of our model in a train-specific way. But many ideas could easily be adopted to other traffic or crowd simulations.

3.2 Trains

A train is a set of wagons. There are three types of wagons, which are different in the way passengers can board. The three types are: First class, second class and bistro wagon. Whereas the first class passengers only board on first class wagons, the second class passengers enter either a second class or bistro wagon. There is also a difference in capacity of the wagons. Second class wagons accept more passengers than a first class or bistro wagon.

Further, we considered in our model, that a train usually arrives later than the passengers do. So the train will move into the station while the passengers are already waiting on the platform.

3.3 Agents

The passengers are the agents in our simulation. They are separated points, each of it with specific properties and a behavioral pattern. The properties are their mass and maximum velocity. Their behavior is more complex and mainly defined by their affinity to class, their mode to choose a door, a limit and frequency of reconsidering the chosen door, etc. (See section 3.3.2)

3.3.1 Agent States

During the simulation, an agent possibly changes between three kinds of states: *deboarding, moving, boarding.* For a *deboarding* agent one only has to check whether he can deboard and for a *boarded* agent the he does not act in any way anymore. The situation for a *moving* agent is more complex. In the *moving* state, the agent will consider to change for another train entrance (within the limitation that it is the same train, same class, etc., see again *section 3.3.2*). The agent also has to be moved as well as it has to be checked whether the agent can enter through a door.

3.3.2 Door Selection

The probably most interesting thing about the modeled situation is the question, on which factors the agents base their decision for a specific door. Obviously, this is an individual optimization problem, where the agents try to optimize their conditions. These conditions can be described by several parameters. In the current case, the passenger's distance to go, the number of other passengers with the same intention or the desire for a free seat in the wagon could be these parameters. To use these parameters all together, it is necessary to normalize them, so that you can compare them. A proper description of an *Exit Selection Model* can be found in [2]. In this paper about evacuation in a fire emergency, they propose to use factors like *estimated evacuation time* (sum of *estimated moving time* and *estimated queuing time*) as well as further factors like familiarity and visibility of the exits and the conditions at the exits. They further propose to separate the exits in some preference groups (depending on the further factors) and so to decide only between the possibilities with the highest preference.

In our model, we implemented these ideas as follows: Mainly, an agent has to reach his final goal, so he has to consider only the doors that lead to the defined destination. In other words, a person A who wants to take the train A only enters a door of train A (we also assume that the person strictly respects the class of the wagons) and a person B who left any train A will not board on train A again, but it will either leave the platform through the subway or board on any other train B. This first selection represents the recommended separation into preference groups.

For the actual door decision, we designed some different functions. We also considered that the order, frequency and number of times an agent can make its decision can influence the result.

The patience factor (used by [2]), that prefers the current strategy with a factor of $0 \le p \le 1$ over other strategies, is also considered.

The functions to evaluate the door's quality are the following:

Random A possible way to choose for a door is by chance. As one will see in the results this will neither lead to a small final boarding time nor does this mode describe a natural behavior of passengers, so we will not discuss this mode further.

Walk A realistic assumption might be that a passenger always minimizes its way to the door. So if r_i represents the agent *i*'s current position and v_i its velocity and b_k means the position of the door e_k , the agent *i*'s strategy s_i is

$$s_i = \min\left(\frac{d(e_k; r_i)}{v_i}\right) = \min\left(\frac{\|r_i - b_k\|}{v_i}\right) \tag{1}$$

Queuing As another natural behaviour, we considered that an agent chooses always the door where the least amount of other agents are heading to. So if f_k describes the frequency agents can pass the door e_k and $\lambda_i(e_k, s_{-i}, r_i)$ is the number of all other agents heading to the door e_k that are closer to it than agent *i* (s_{-i} are the strategies of all agents without agent *i*), then its strategy s_i is

$$s_i = \min\left(\frac{\lambda_i(e_k, s_{-i}, r_i)}{f_k}\right) \tag{2}$$

Sum The logical conclusion is that the natural behavior is a mix of the two strategies walk and queue. Whereas busy people rather walk a longer distance to reach a door with less other people, a lazy agent rather decides for the closer door without considering the number of people already queuing there. So if $\mu_i \in [0, 1]$ describes the laziness of agent *i*, its strategy s_i is

$$s_{i} = \min\left(\mu_{k} \frac{\|r_{i} - b_{k}\|}{v_{i}} + (1 - \mu_{k}) \frac{\lambda_{i}(e_{k}, s_{-i}, r_{i})}{f_{k}}\right)$$
(3)

3.3.3 Groups

While travelling on train, people are often formed in groups. A group is a set of agents that strictly decides for the same strategy. The strategy is either defined to be constant, chosen by the majority of the group or by a group leader. Despite of the difference that group members will not consider the members of the same group by doing their decision, the strategies are similar to the individual agent's strategies.

3.4 Doors

Obviously, a door is a defined area where you change from one part to another part of space. In our simulation the passengers on platform can enter a train or subway respectively do it the inverse way.

Like the agents, the doors have some properties. So each door is determined to be a first or second class wagon entrance respectively a subway entry. Further, there are a frequency and limit of the agent that can pass the door.

3.5 Obstacles

To limit the space where an agent is allowed to move, we included some obstacles in our model. The obstacles describe as well *real* obstacles on the platform (like waiting huts, subways, poles, etc.) as also the borders of the platform. An obstacle is defined by its position, size and period of time it is active. The last parameter is thought to be used to hold the agents back in a defined waiting area until a specific moment.

3.6 Dynamics

We based the movement of the agents on a model already used by [4], who chose it according to a homework from the lecture *Simulations using Particles* by Prof. Petros Koumoutsakos. In this model, the agents are moving like particle in a potential field, so the acceleration of an agent is described by the acting force on it, divided by its mass. In the model used for our simulation, we based the movement of an agent on the referred model. The force acting on an agent is given by the surrounding (See *section* 3.6). Based on this force, we calculate the actual acceleration.

$$d\vec{v}(t) = \frac{F_{res}(t)}{m_{agent}} \tag{4}$$

But the persons will not orbit over the platform like planets, but rather find their way as a smooth line towards their goal. This means that there has to be a limitation of their speed or a friction force, which lets the agent move in a more natural way. So the current velocity is limited to the maximum velocity of the agent.

$$\vec{v}_{new}(t) = \vec{v}(t - dt) + d\vec{v}(t)dt \tag{5}$$

$$\vec{v}(t) = \min\left\{ v_{agent,max} \frac{\vec{v}_{new}(t)}{\|\vec{v}_{new}(t)\|_2}, \vec{v}_{new}(t) \right\}$$
(6)

3.7 Forces

There are three kinds of *forces* acting on an agent. The sum of the influence of the doors, obstacles and other agents yields the resulting *force*.

3.7.1 Doors

The main direction of the agent's movement has to be toward its goal, which is always a door (to be precise, this door is door k which is the best strategy s_i for agent i). Therefore, there is a vector $\vec{e}_{i,D}^u(s_i, r_i, b_k)$ that directs from the agent i's position r_i to the position of the door b_k .

$$\vec{e}_{i,D}^{a}(s_{i}, r_{i}, b_{k}) = \frac{\vec{b_{k}} - \vec{r_{i}}}{\left\|\vec{b_{k}} - \vec{r_{i}}\right\|}$$
(7)

In the case of queuing in front of the door, the agents don't need to approach it up to the point where they have the door's exact position. The area within the space of the door has to be kept empty for eventual agents leaving the door or the next agent boarding it. So we add another vector $\vec{e}_{i,D}^r(s_i, r_i, b_k)$ directing the inverse direction. Its amount is proportional to the inverse of the distance $\|\vec{b}_k - \vec{r}_i\|$ between agent *i* and door *k* and the door range factor d_D .

$$\vec{e}_{i,D}^{r}(s_i, r_i, b_k) = -\frac{d_D}{\left\| \vec{b}_k - \vec{r}_i \right\|} \cdot \frac{\vec{b}_k - \vec{r}_i}{\left\| \vec{b}_k - \vec{r}_i \right\|}$$
(8)



Figure 1: At a rectangular obstacle (bold), there are nine sectors (dashed). An agent is always retracted by the nearest point of the obstacle (arrows).

Now, the force yielding by the door is

$$\vec{F}_{i,D}(s_i, r_i, b_k) = k_D \cdot \left(\vec{e}_{i,D}^a(s_i, r_i, b_k) + \vec{e}_{i,D}^r(s_i, r_i, b_k) \right)$$
(9)

where k_D is the factor that describes the general strength of the door forces. There will be an equilibrium point where the force becomes zero if $d_D = \|\vec{b_k} - \vec{r_i}\|$.

3.7.2 Obstacles

There are several ways for modelling obstacles. In the simulation from [4] it is proposed to model a wall or any other obstacle as a set of fixed point. Then a retraction force between each agent and each obstacle point can be calculated.

To decrease the amount of calculation we decided to introduce another model where an obstacle is represented by a rectangle. Then, an agent is always retracted either by the side respectively edge of the rectangle, which is closest to the agent. This model is valid for agents inside as well as outside the obstacle. To actually decide which side or edge of the obstacle has to be considered for the retraction, the space around each obstacle is split in eight sectors outside and one sector inside the obstacle. The force will be as shown in *figure 1*.

The vector $\vec{l}_{i,j}(\bar{\bar{D}}_{i,j})$ from the agent *i*'s position \vec{r}_i to the point of the obstacle which is the nearest to it, can be calculated as

$$\vec{l}_{i,j}(\bar{\bar{D}}_{i,j}) = \left(\bar{\bar{D}}_{i,j}^{PD}\vec{o}_j + \bar{\bar{D}}_{i,j}\vec{d}_{O,j}\right) - \bar{\bar{D}}_{i,j}^{PD}\vec{r}_i$$
(10)

where $\vec{o_j}$ is the center point of obstacle j, $\vec{d}_{O,j}$ the dimension of the rectangle. $\bar{D}_{i,j}$ is a matrix determined for each sector around an obstacle as follows (Compare also figure 1).

$$\bar{\bar{D}}_{i,j} := \begin{bmatrix} D_x & 0\\ 0 & D_y \end{bmatrix}$$
(11)

with

$$D_{x} = \begin{cases} -1 & \text{if } r_{x} < o_{j,x} - d_{O,j,x} \\ 1 & \text{if } r_{x} > o_{j,x} + d_{O,j,x} \\ 0 & \text{otherwise} \end{cases}$$
(12)

and

$$D_{y} = \begin{cases} -1 & \text{if } r_{y} < o_{j,y} - d_{O,j,y} \\ 1 & \text{if } r_{y} > o_{j,y} + d_{O,j,y} \\ 0 & \text{otherwise} \end{cases}$$
(13)

where $o_{j,y}$, $o_{j,y}$, $d_{O,j,y}$, $d_{O,j,y}$ are the components of o_j respectively d_{O_j} . $\bar{D}_{i,j}^{PD}$ is the same matrix than $\bar{\bar{D}}_{i,j}$ but with the elementwise absolute values.

Finally we find the force $\vec{F}_{i,O}(r_i, o_k, d_k)$ acting on agent *i* caused by obstacle *j*

$$\vec{F}_{i,O}(r_i, o_k, d_k) = -k_O \cdot \vec{l}(\bar{\bar{D}}_{i,j}) \cdot \frac{1}{\left\| \vec{l}(\bar{\bar{D}}_{i,j}) \right\|}$$
(14)

where k_O represents the general strength of the obstacle forces.

3.7.3 Agents

Agents should keep a certain distance between them, so that the queuing procedure becomes realistic. Therefore we have to implement a retraction force between them that is proportional to the inverse distance between each pair of agents.

With some investigations, we concluded that it makes sense if the agents behave similar to atoms in a crystal lattice (compare *figure 2*).

So the agents in front of the door compose a realistic crowd. Therefore, the force $\vec{F}_{i,A}(r_i, r_h)$ acting on agent *i* caused by agent *h* is calculated as

$$\vec{F}_{i,A}(r_i, r_h) = \left(\frac{1}{\|\vec{r}_h - \vec{r}_i\|^2} - \frac{d_A}{\|\vec{r}_h - \vec{r}_i\|^3}\right) \cdot \frac{\vec{r}_h - \vec{r}_i}{\|\vec{r}_h - \vec{r}_i\|}$$
(15)

where d_A is the agent's required space.



Figure 2: Qualitative diagram of the force between agents, similar to a crystal lattice.

3.8 Simulated Situations

Finally, the introduced objects have to be placed to simulate a specific situation on a train station. The position of the train as well as the number and class of its wagons have to be defined. Further, the subways and any obstacles on the platform have to be placed. In a final step, agents and all their personal properties need to be set.

In our simulation, we analyzed the situation where the train consists of SBB EuroCity wagons. Each train has two first class, a bistro and three second class wagons.

3.8.1 Two Trains

Our first situation represents a common situation at Zurich HB. There are two trains parallel at the same platform. There are entrances to the subway as well as some obstacles (piles) on the platform. There are travellers changing from one train to the other, some are leaving through the subway and others enter one of the trains.



Figure 3: Two Trains Situation. The dashed line defines the waiting area for the passengers. The small green circles mark the doors. The small squares along the train represent the available seats in the train by their color.

3.8.2 One Train

The other situation we implemented in our simulation represents any station where a single train arrives. Some passengers leave the train and exit through the subway. Some other passengers are waiting anywhere on the platform and are going to enter the train as soon as the outcoming passengers finished deboarding.



Figure 4: One Train Situation.



Figure 5: The agent (red triangle) has to move to the door (green circle) which is hidden behind the obstacle. If there are only the forces \vec{F}_D and \vec{F}_O , the agent will end up in the equilibrium point E. The correction force \vec{F}_{Corr} leads the agent around the obstacle.

4 Implementation

This chapter first alludes on some specific add-ons that had to be made to guarantee the model's proper functioning. Afterwards it should give a small overview about the created Matlab-Files and their functionality.

4.1 Model Add-ons

4.1.1 Forces Correction

As the agents sometimes ended up in a dead-end on their way to their goal, we had to improve our force-model by a correction force. If an agent is close to an obstacle, and the sum of the door force \vec{F}_D and obstacle force \vec{F}_O becomes very small, this correction force leads the agents around the obstacles as shown in *figures 5 and 6*. In the first of the two discussed situations, the door is "hidden" behind the obstacle. The resulting force $\vec{F}_D + \vec{F}_O$ leads to a dead-end equilibrium point on the right side of the obstacle. Therefore the correction force has to lead the agent around the closest edge of the obstacle.

The mathematical description of the force \vec{F}_{Corr} is

$$\vec{F}_{Corr} = \operatorname{sign}(\vec{F_O} \diamond \vec{F_D}) \cdot k_C \cdot \frac{\vec{o_O}}{\|\vec{o_O}\|}$$
(16)

where $(\vec{F_O} \diamond \vec{F_D})$ is a two dimensional vector product



Figure 6: Different to figure 5 the door is not directly behind the obstacle, but the straight way is blocked, too. As the forces \vec{F}_D and \vec{F}_O would already lead the agent the right way around the obstacle, but are not powerful enough, the correction force \vec{F}_{Corr} will lead the agent around the obstacle's edge.

$$\begin{bmatrix} a \\ b \end{bmatrix} \diamond \begin{bmatrix} c \\ d \end{bmatrix} = ad - bc \tag{17}$$

that is also used by *Prof. Dr. C. Glocker* in the lecture *Mechanik III.* \vec{o}_O is a vector orthogal to the obstacle retraction force vector \vec{F}_O , actually it is 90 degrees turned clockwise.

$$\vec{o}_O = \begin{bmatrix} 0 & -1\\ 1 & 0 \end{bmatrix} \vec{F}_O \tag{18}$$

In the example of figure 5, $(\vec{F_O} \diamond \vec{F_D})$ is negative and $\vec{o_O}$ is directed to south, so $\vec{F_{Corr}}$ is a vector with direction north and its value is the correction force factor k_C .

In the situation in *figure 6*, the door is not directly behind any object (in fact, this door represents a train entrance, whereas the door in *figure 5* is always a subway exit). In this case the correction force has to act the other way round. Because the vector product $(\vec{F}_O \diamond \vec{F}_D)$ will have the inverse sign than in the first example, the correction force for agents attracted by a train's door is

$$\vec{F}_{Corr} = -\text{sign}(\vec{F_O} \diamond \vec{F_D}) \cdot k_C \cdot \frac{\vec{o_O}}{\|\vec{o_O}\|}$$
(19)

4.2 Initialization

We split the initialization into several files. To simulate a specific testcase, the parameters have to be set in $run_testcase.m$. This file calls all the specific initialization files before the actual simulation starts.

init_globals This file defines a lot of constants that define the matrices that are used during the simulation.

init_szenario As we defined two scenarios, there are two of these files (one for the one train situation, the other for the two train situation). In these files, all the parameters for the agents, doors and obstacles are set, referring to the given specifications. There are some specific files to distribute the agents properly on the platform and to define the groups. By setting the agent's initial positions randomly, one has to ensure that they are not set inside of any obstacle. The group initial file sets all members of the same group close together on the platform.

init_statistics To collect the data during the simulation, a lot of numbers have to be stored into matrices. These matrices are initialized in this file.

init_style This file defines what the simulation should display and which data should be stored where. It is possible to display either a map with the agent's position or some specific curves. The simulation can also run without displaying anything.

4.3 Simulation

The *simulation.m* file includes the time iteration for the simulation. The order of actualizing the agents' state follows a random permutation. Mainly, the following procedures are done in every iteration step. They will be explained in the following sections.

- Update the agent's **state**: Check whether an agent is currently boarding or deboarding.
- Update **strategy**: Check whether the agent should change his door decision.
- Calculate **forces**: Determine the movement for every agent.
- **Move** agents: Set the new position for every agent according to the calculated force.

- **Plot** the current situation.
- Save data: Actualize statistics and save data and/or pictures.

4.4 Update state

One parameter of the doors is the time, until the next agent can pass it. If a door is available, the *agent_update.m* file checks first whether there is an agent inside that wants to deboard. If so, the agent's state will change to *moving*. This means that the agent is now on the platform and interacts with its surrounding.

If there are no more deboarder left, it is checked, whether there is an agent close enough to the door to enter it. If so, the agents state changes to *boarded* and the agent will be set on a seat in the coach. If the agent occupies the last available seat of the coach, the doors' activity will turn to inactive, so that no other agents will try to enter this coach.

4.4.1 Look for a seat

The files *agent_seat_search.m* and *coach_seat_search.m* are used to check where the agent takes a seat inside the train. These files will only be valid for the two introduced situations *One Train Model* and *Two Train Model*, because the two doors of every coach have to be named explicitly. The first file checks whether there is any space left on the chosen coach. The second file than finds the first available compartment for the agent.

4.5 Update strategy

In the initialization file a door decision frequency gets specified. The file *doordecision_frequency.m* then determines how often the agent can change its mind during the current step. The door decision frequency can be any positive number $|f| \in \mathbf{R}$.

The strategy will only be checked for *moving* agents. Their limit of allowed redecisions must not be reached. Then the *place in queue* as well as the *remaining distance* for each door will be calculated for agent i. The decision for the best door will be made by using the agent's kind of *door selection* (See section 3.3.2). The number of available redecisions will have decreased. If agent i is member of a group, the whole group's strategy will be the chosen door.

4.6 Calculate forces

The file *calculate_forces.m* calculates the force acting on every agent as it is described in *section 3.7*. As a result, the force acting on agent i will be stored so the next

procedure will determine the agent's movement.

4.7 Move agent

Also the moving procedure follows the rules described in the description of the model (see section 3.6) The steps walked by the agents are calculated using the explicit Euler formula.

4.8 Plot

As the scene has changed during the recent time step, the current situation gets displayed. There are two plotting modes and an off-switch:

- Plot map
- Plot graph
- no plot

Plot map shows the situation at the station with the agents as dots on the platform (*figure 7*). This will be very convenient to observe the behaviour of the full crowd of the agents.



Figure 7: In the *mapping mode*, the agents are displayed as red points (violet if they are member of a group). The doors are marked by a green circle (respectively red cross if an agent has recently used it). Obstacles are displayed with black lines (dotted if it is inactive). The train consists of images by MAERKLIN model railway coaches. The scale is represented in decimeter, as the image of the trains can not be scaled to less than one Matlab plot unit per pixel.



Figure 8: In the graph mode, these nine subplots will be displayed during the simulation.

Plot graph displays six time diagrams and three agent diagrams. As an example we take *figure 8* that represents a standard situation from the *One Train Model*. See *section 5.1.2* for a detailed explanation and interpretation of these plots.

- Approaching shows how many agents are heading to which door. The solid line represents the subway. The stability of the Nash equilibrium can be itentified by the lines' variation.
- Moving shows how many agents are on state *moving*, i.e. the number of agents walking on the platform. It can easily be identified, that the agents start deboarding at $t_0 = 10s$.
- **Distance** indicates the mean remaining distance to the agents' chosen door.
- (De-)boarded plots the number of agents that already (de-)borded at this door. The capacity of the door can be recognized by the frequency the curves rise.
- Waiting shows the number of agents queuing at a door. The plot differs between agents boarding on a train or leaving through a subway.
- **Time waited** shows for every single agent the time he had to queue. The agents are sorted from left to right (*first class boarding, f.c. deboarding, s.c. boarding, s.c. deboarding*). As long as an agent is still moving, its dot will

be plotted in blue. Boarded agents are plotted in green. The first class agents had to queue less time. As most agents boarded within 50 seconds, there are almost no blue dots left.

- **Redecisions** is the recent number of how many time each single agent preferred a new door. Again, the value for first class agents is smaller.
- **Distance walked** shows for every single agent the distance it actually walked (blue/green) as well as the linear distance from the agents start position to its final goal. Of course the actual distance has to be at least as big as the linear distance. The deboarding agents' distances are almost equal to their minimal distances.

In case of interest, every plot can easily be plotted in a single plot by calling the specific $plot_saved_\cdots.m$ file.

4.9 Save Data

The *save_data.m* file is responsible to store all statistics values of any interests. Most of these values can also be identified on the *Graph Plot*.

5 Simulation Results and Discussion

5.1 General Results

Before analyzing the effect of single parts of the simulation we noticed some interesting properties of our simulation in general. The graphs in this section have been calculated using our default setup that is described in *section 5.2*.

5.1.1 Observations on the map

Beginning The simulation starts 10 seconds before the train reaches its final position and opens its doors. At that time the waiting passengers are randomly distributed over the platform. *Figure 9* shows this situation as a 2d-plot.



Figure 9: Situation at the beginning of the simulation. The yellow squares mark compartments that are already used by two passengers. The red ones are already full, which represents, that there are fewer seats in the first class and bistro coaches. All passengers are distributed randomly over the waiting area but outside of all the obstacles.

Door opening During the first 10 seconds the passengers start approaching a door, although all doors are still moving. Then the doors open and first all leaving and changing passenger get off the train. As no agent is able to board yet, they start forming semi-circular crowds around the doors, which is visible in *figure 10*.



Figure 10: Train station after 10 seconds of simulation time. Around the most popular and most central doors semi-circular crowds start to get formed.

Boarding After 30 seconds, most of the agents have reached their favorite door and some of them already boarded. Around all popular doors the passengers build semi-circular waiting crowds and are in balance between moving closer to the door and not getting to close to any other agent or getting pressed to the train. *Figure* 11 also shows how the compartments inside the train are getting occupied.



Figure 11: Train station after 30 seconds of simulation time. People are waiting in front of their favorite door and the amount of free seats has decreased on all coaches (red compartments symbolize no more free seats)

Final state 65 seconds after the train arrived the last agent enters the train. When the simulation ends after 90 seconds, all agents have found a seat. It is important to remark that both bistro coaches and one second class coach have been filled up to the last seat. That means that some agents had to redecide for another door at another coach, which caused the train to wait longer. In the end the compartment allocation looks like in *figure 12*.



Figure 12: End of simulation after 90 seconds. All passengers boarded and some coaches are full.

5.1.2 Insights out of the graphs

The graph view explained in *figure 8* provides a comprehensive overview of all the recorded data. The following paragraphs show some of these observations.

Approaching agents Figure 13 contains a lot of information about how the agents decide for a door. During the first few seconds the curves are quite unstable which means that the door decision has not stabilized yet and the Nash equilibrium has not been found yet. But after the doors opened the curves get quite steady and the number of approaching agents decreases almost linearly.

The highest peak belongs to the leftmost exit which is located more central than the other two exit and therefore is approached by most of the second-class passengers. About 10 seconds after the door opening this curve starts to decrease again, which means that already more passengers are entering the exit than there are new agents approaching, who just got off the train. About 45 seconds after start of the simulation, there are already no more leaving agents on the platform.

Something interesting happens after 46 seconds or 47 seconds respectively. There are only four doors, where there are still agents approaching. The two light green lines belong to the rightmost second class door of each train, whereas the dark green lines correspond to the left doors of the two bistro coaches. The bistro coach is filled up, so all the waiting agents there have to redecide. Of course all of them decide for the door, that is just 5 meters away and still has some empty seats. In the figure it is visible how the agents swap from the dark to the light green lines. A few seconds later one of the two second class coaches is full too. The one on the opposite train had enough free seats to accommodate all waiting agents.

Figure 14 contains a detailed view of the agent distributions between these four most popular doors: the one on the left of the bistro-coach and right-most door of the second-class coaches. Here it is clearly visible, that the Nash equilibrium is not very stable at all. There are up the three agents, who redecide from one door to another, when meanwhile three other agents decide to do the opposite. And after just a few time steps they decide back. This confirms our assumption from the model overview (see section 3.1), that an optimal door-decision-strategy would need to be a mixed strategy and therefore could not be found using the iterative algorithm proposed in [2].



Figure 13: Course of the agents' door approaching behaviour



Figure 14: Detail view of the approaching agents to the two left-most doors of the bistro coaches (dark green) and the two right-most doors of the second class coaches (light green) in the time interval [10s, 15s].

Moving and waiting agents Figure 15 displays how many agents are on the platform. This amount is constant until the doors open. Then it increases until more agents board than get off the train and finally it decreases in a hyperbolic way until the last agent boards after about 70 seconds. After 45 seconds about 90% of the agents have already boarded. So a large part of the time that the train has to stay at the station is caused by only a few agents that either have to walk too long or redecide too late.

The number of waiting agents as shown in *figure 16* consists mainly of the boarding agents. The peak is at about 20 seconds, when most boarding agents arrived at their preferred door, but are still unable to enter the train, because deboarding has not finished yet. Afterward it decreases almost linearly, as agents can board homogeneously. For the exiting agents there are some short queues in front of the subway entrances from about 20 to 40 seconds.

Figure 17 shows the waiting time of each agent separately. The agents are grouped by class and mode. One can see that the leaving agents almost never have to queue. As the density of first class passengers per door is much lower, they also have to wait less than second class travellers. The passengers that change the train also have to wait less for a couple of different reasons:

One reason is, that they spend the first ten seconds or even longer inside the

train, which we do not count as waiting because we are only interested in how long the agents queue before they can board the train. When they get off and walk to the opposite train, the queue there will already have shortened a bit. Another reason is, that the leaving doors are equally distributed, which means that some agents get off somewhere near the end of the train, where almost no passengers queue on the opposite side.



Figure 15: Agents situated on the platform



Figure 16: Agents that have to wait just in front of their preferred door



Figure 17: Waiting time per agent. The vertical lines separate the 12 types of agents: [1 - 6] first class; [7 - 12] second class; [odd] train one; [even] train two; [1,2,7,8] boarding agents; [3,4,9,10] deboarding agents; [5,6,11,12] changing agents

Boarding and deboarding agents These two graphs (*figure 18* and *figure 19*) just show what we expected. The doors let people in and out at almost regular time steps.



Figure 18: Agents that are waiting behind a door in order to get off the train



Figure 19: One graph per door, describing the number of agents inside this door, not considering the agents that stay inside permanently

Door redecisions Similar to the distribution of the waiting time, *figure 19* shows that almost all redecisions are done by boarding agents. It is no surprise that this value correlates with the waiting time, as it is much harder to decide, if there are a lot of other passengers queueing in front of an agent, who thinks about redeciding for a door that is further away.



Figure 20: Number of redecisions per agent

Average distance The plot of the average distance between an agent and his chosen door (*figure 21*) reflects the different stages of the simulation. At first the distance falls linearly, which means that all agents move at full speed toward their chosen door. Then, as the first agents start to queue and new agents deboard, the distance decreases less and less. The two small peaks at about 46s and 47s reflect the two bistro coaches getting filled up. But as the next free door is just 5 meters away the peaks are quite small. It gets worse, when the second class coach has reached its maximum capacity and all of the still moving agents have to walk another twenty meters.

In the agent distribution in *figure 22* one can see, that again the boarding agent have the biggest walking overhead. Deboarding agents are able to walk almost straight to their exit. Most changing first class agents can walk directly to their preferred door too. This is caused by the topology of the train station, where the left subway obstacle blocks primarily second class travellers.



Figure 21: Average distance between all agents and their chosen door



Figure 22: Comparison between linear line distance and the walked distance

Simulation Errors In some cases, where we varied our default parameter values, sometimes imprecisions in our simulation have occured. It could happen, that a few agents get stuck somewhere in a dead spot where the resulting force is almost equal to zero. At first this could happen when the agent approached an obstacle that was in line with him and his chosen door. Than the retraction of the obstacle erased all the attraction of the door at a certain point. Because of that we introduced a correcting force that eliminated almost all of this situations (see *section 4.1.1*).

Another source of errors are inappropriately high time steps or velocities. Our implementation does not include an explicit obstacle collision detection. Therefore the obstacle retraction is calibrated high enough, such that an agent with default velocity can not move so far in a default time step, that he would cross any obstacle or train border.

5.2 Simulation Variables and Result Indicators

In order to study the influence of many simulation parameters on the boarding process, we specified a base case and a series of other test cases to compare it to $Table \ 1$ lists all parameters that can easily be varied in the initialization file. The values written in *italics* marks the default value. The abbreviations in square brackets for each parameter are often used in the code and in the appendix .

The default values are our assumptions about a regular, crowded train station. In the following sections we varied each group of parameters separately and compared the results to the base case presented above.

Szenario					
Number of Trains	one train [OT]				
	two trains [TT]				
Amount of agents					
People on Platform [PoP]	few (50 passengers per train)				
	many (100 passengers per train)				
	too many (200 passengers per train)				
People deboarding [Pd]	few (25 passengers per train)				
	many (50 passengers per train)				
	too many (100 passengers per train)				
People already seated [Pas]	few (50 passengers per train)				
	many (200 passengers per train)				
	too many (300 passengers per train)				
People attributes and behaviour					
Class ratio (first class share) [FCR]	0.2				
Maximum velocity Distribution	$1\frac{m}{s}$				
(mean and variance)	$1.5\frac{m}{2}$				
	$1.5 \pm 0.5 \frac{m}{2}$ (default)				
	$2.5 \pm 1.5 \frac{m}{s}$				
Group ratio	no groups				
	20% grouped				
	50% grouped				
Door decision parameters					
Door decision mode [DDM]	walk				
	sum				
	queue				
	wait				
	random				
Lazyness coefficient [LC]	$\{0, 0.1, \dots 0.5, \dots 0.9, 1.0\}$				
Door decision frequency [DSF]	1 decisions per second				
	20 decisions per second				
	100 decisions per second				
Limitation of the amount of door decisions [DL]	$\{1, 10, \infty(default)\}$				
Patience factor for door redecision	$\{0.5, 0.9, 1.0\}$				
Time					
Time until waiting area opens [TW]	1s/1s (instant)				
and time until doors open [TD]	5s/10s (default)				
	10s/20s (late)				
Simulation Duration	90s				

Table 1: different simulation parameters

Statistical measurements To compare the results of the different test setups, we defined a number of statistical values, which we calculated in every run of the simulation. Of course this is only a small subset of all the data, which was recorded during each simulation. We have chosen the values, which we think are most relevant in terms of how a train company could measure the quality and efficiency of their train stations.

All the values have been calculated considering only the agents that boarded a train during the simulation time, including the ones changing from one train to another. This is important because we are only interested in how long it takes until the train is able to depart and do not care, if some agents still are on the way out. Additionally, as exit doors have a much higher capacity than train doors, it is important that these doors are not considered when studying the distribution of the agents over the doors or the average waiting times.

So these are the evaluated values:

- maximal boarding time (latest time that any agent entered a train) [maximum]
- boarding time [average / standard deviation]
- covered distance (the distance an agent walked from its initial position or his leaving door) [average / standard deviation]
- waiting time (the amount of time the agent spent waiting close to his chosen door) [average / standard deviation]
- number of redecisions [average / standard deviation]
- distribution of boarding agents per door [standard deviation]

5.3 Test Series

Each following subsection presents a detailed analysis of one of the specified parameters. The statistical measurements of all test cases used for this section can be found in the appendix. The full test results, including the matlab workspace at the end of each simulation, are available on the homepage (see *section 8.1*).

5.3.1 Number of Agents

At first we varied the number of agents per train. The diagrams in *figure 23* show, that all measurements increase the more agents are added. It is not surprising, that more people result in longer ways, more redecisions, longer waiting queues and so on. When comparing the train scenarios, it is at first not clear, why the agents need

to walk, redecide and wait less, if they are in the two train situation. But it gets reasonable, if we consider, that there are only in this scenario additional changing agents. They have to wait less and can just walk across the platform, without trying to decide for a door while the train is still moving.

5.3.2 Door Decision Modes

It was our main goal to study different door decision algorithms. So we started first with just the five different methods described in *section 3.3.2* with default parameters. At first it is important to remark, that the values for the *random*-mode are not quite comparable (see *figure 24*). There the agents board so slowly, that after 90 seconds most agents are still on the platform and therefore not considered in these diagrams.

In most categories the *queue*-mode seems to be the most promising one, especially if we look at the maximal boarding time. Although the difference to the walk and the sum approach are not that big regarding the average boarding time, the train would be able to leave about 13 seconds earlier if all passengers would chose their door only based on the length of the queues. This method also distributes the agents much more uniformly over the doors than the other modes. The only downside of this mode is the big number of door decisions. This probably makes it really hard to apply this mode, if we would try it in reality.

The *sum*-mode is the algorithm, that we think describes best, what human passengers do. They normally look around and decide for the nearest door. Only if it is to crowded there, they will take the longer way to the next door. At first sight, it would make sense, if this was the optimal method. So why is it not optimal to minimize the sum of the walking and the queueing time?

- This would only be optimal from the point of view of a single agent and not in the global scale. For many agents this method seems in fact to be faster than the *queue*-mode, as the mean boarding time is just one second higher with a higher deviation.
- If a door A is much closer at the beginning than a door B, this does not imply, that an agent can board earlier there, even if both doors have no queues at all. Then during the entrance of the train and the initial deboarding, so from about 15 to 20 seconds, no boarding is possible anyways. This gives agents that move to doors further away the chance to catch up, while the others just queue before the door.
- Last and probably most important is the fact, that with this method some central coaches get filled up completely and the agents queuing there have to


Figure 23: Variation of the number of agents that are moving through the train station.

walk to another door. Unfortunately the agents do not see this event coming and all of them stay queuing there until the last seat is taken. The *queue*-mode adapts better to this issue, as the agents choose the doors more equally and therefore no coach gets filled up completely.

For the *walk*-method the same arguments as for the *sum*-methods apply, just in a intensified way. Here it strikes the eyes, how minimal the number of redecisions is. The *walk*-mode is the only algorithm, which can choose based on the initial position of the agent and does not have to take the other agents into consideration. The only two situations, in which such an agent has to decide again, is when either his chosen coach is full or when he has to make a large detour around an obstacle or a crowd and he comes across a new nearest door.

The *wait*-method seems to do a very good job in his main concern: minimizing the waiting time. The downside of this approach is the much longer distance and the huge number of redecisions. As said before the waiting time for the *random*-algorithm is not really comparable here, as most of the agents don't even get near the door before the simulation stops.

5.3.3 Laziness Coefficient Optimisation

As we have seen some remarkable differences between the *queue*-, *sum*- and *walk*modes, we wanted a more precise separation and simulated with different agent laziness coefficient in 0.1-steps. So laziness 0 corresponds to the *queue*-mode, 0.5 is exactly the same as the *sum*-mode and laziness 1.0 means minimizing the walking time.

For both, the maximal boarding time and the average boarding time, a laziness factor of 0.1 is optimal. 0.1 is even better than 0.0, because then agents might change their minds, if another door, that was not their chosen door but is closer, gets free. This is of special importance at the end, when only a few agents are left, hence the large gap in the plot of the maximal boarding time (see *figure 25*).

This leads to quite an astonishing "paradox": The more an agent tries to minimize his walking time, the more he actually has to walk. This can be explained by the fact that, with the default parameters and the *lazy*-mode, no coach gets filled, which saves a few agents to walk some additional, unscheduled extra-distance to another coach.

5.3.4 Door Decision Frequency

In order to see whether the simulation depends on the frequency of the door decisions, we varied the number of iterations per seconds of the door decision process (see *figure* 26). The effect seems to be visible, but not statistically significant.



Figure 24: Influence of the different door decision modes



Figure 25: Variation of the laziness coefficient



Figure 26: Boarding times with different update frequencies of the door decision process





5.3.5 Door Decision Limit

We also limited the number of times an agent is allowed to change his mind and approach a different door. *Figure 27* shows, that it is important for the agents to have more than just one chance to decide for a door. The difference between 10 and unlimited decisions is not significant.

5.3.6 Patience

A variation of the patience factor as in *figure 28* seems not to have a significant influence on the boarding times. A small patience factor of 0.5, which means that agents redecide only if another door is expected to let them in twice as fast, seems to be a little bit faster, if we look at the mean boarding time.



Figure 28: Different patience factors. 0.5 means that the agent compares all the estimated times to reach another door with half of his current one.

5.3.7 Velocity

Higher velocities reduce the final boarding time drastically (see *figure 29*). However the average boarding time does not decrease that much. This is the case, because during the first 15 seconds no agent can board anyway, so a higher velocity only helps the agents to get closer to the door before boarding starts. When we distribute the maximal velocity per agent randomly around $1.5\frac{m}{s}$ with standard deviation $0.5\frac{m}{s}$ this seems to be a little bit faster compared to the situation, where every agent has the same maximal velocity of $1.5\frac{m}{s}$. But the difference is not significant.



Figure 29: Variation of the maximal velocity of the agents

5.3.8 Groups

As expected having a lot of groups on the platform increases the mean boarding time and the average distance (see *figure 30*). The different amount of redecisions might be caused by the fact, that in a group all agents have to redecide together. But again, the influence seems to be quite small and the differences not statistically significant, considering that only five simulations of each group have been made.



Figure 30: Different group ratios. Each group consists of 10 agents.

5.3.9 Simulation Start relative to the Door Opening

When varying the point in time when the waiting area opens and when the doors of the train open, one can notice some significant differences (see *figure 31*).

There is a complex reason, why the maximal boarding time for the default start is bigger than for the two other starts. If the train arrives quicker, then the agents have no time to group around the still driving doors. Therefore there are more agents that decide for the doors at the end of the train. These doors are not considered that often in the default case because they are too far away at the beginning.

With the third case, where the train still drives for 20 seconds, the opposite happens. The agents pile up in front of the door, but are not fast enough to follow the train. The agents build a kind of tractrix behind the train until their shortfall gets too big and they redecide for the next door arriving. Because this process takes longer in this case, the agents stay more evenly distributed on the boarding platform and therefore also chose their boarding door more evenly. The duration of this initial train-following-process also correlates with the number of door decisions and the covered distance.

If we take a look at the mean boarding time we see, that a longer time period with closed doors increases it by about 7 seconds for every additional 10 seconds. This means that the additional ten seconds of preparation on the platform helps the agents to reduce the boarding time by three seconds.



Figure 31: Variation of the moment where the simulation starts relative to the trainentrance

5.3.10 Waiting Area

In our default case the waiting area is quite big and sprawls over almost the entire platform. In this test case we compared the base case to a smaller waiting area that only covers the space between the two central subway exits. We did this comparison for both train stations (see *figure 32*). The mean boarding time increases with the smaller area, because they are not able spread as quickly over the platform as in the base case. It also results in a longer way for the average agent.



Figure 32: Two sizes of the waiting area simulated in both train stations

6 Summary and Outlook

Conclusions The main aim of our project was to find an appropriate model that is able to simulate big crowds moving in a train station. We intended to find some specific parameters that affect the boarding behaviour of the passengers and with that an optimization of the minimal boarding time and to maximize distribution of the passengers in the different coaches.

The model we implemented was able to simulate the intended situations (i.e. Zurich, Sargans) on a platform quite well. The model is able to include various scenarios, like different exits and obstacles.

With the introduced decision modes for the passengers, we were able to analyze the effect of different decision parameters on the resulting final situation. Therefore, a strategy, which mainly consists of taking the door with the smallest queue, leads to a minimal final boarding time. All others characteristics as mean distance walked, mean time waited and the uniform distribution over all coaches are optimized by this strategy. Other factors like the frequency of decision making, limitations of maximum redecisions as well as the patience factor had a rather small influence.

Discussion We calculated five simulations in 36 different test cases. This already took several hours of computation. But in order to get more meaningful results there would be many more simulations needed. Our test suite was big enough to separate many differences clearly, but in some unclear cases more data samples would have been helpful.

There are also some imprecisions in our model. If the scenario is quite complex the force model does not imply a way from the agents position towards his chosen door. Another problem is, that if a crowd gets bigger and bigger the agent density increases to an unrealistic high level. That way it can happen, that an agent experiences a force so heavy, that he gets "pressed through the wall" and is unable to get out again. An additional obstacle-agent-comparison might be helpful there.

All in all our model and our implementation fulfills our expectations and was able to provide some interesting results.

Outlook There is always room for possible extensions:

- An extended door decision mode, which also takes the number of free seats behind the door into account, could help to optimize the boarding time even further.
- If one would like to simulate more complex scenarios, like complete train stations, with many more trains and obstacles, it would be essential to use much

more object oriented design in the implementation. That way every door would be linked to a coach and every coach would be part of a train. This would make the whole setup process much easier.

- In areas with more obstacles the force model would need to be extended with some sort of shortest-path detection like Dijkstra's algorithm. The combination of such a graph algorithm with the existing model of attracting and repulsive forces could get quite tricky though.
- To make the boarding process more realistic, it would be necessary to simulate the interior of the coaches also with freely moving agents. One could take into account, that boarding an already full coach is still possible, as long going as through the coach at some slower speed is not impossible.
- Another interesting observation is, that passengers might behave completely different in some special situations. For instance, when the train is on the verge of leaving and an agent wants to board, but just realizes that his door is broken, the agent will start running towards the next door. All agent are just able to run for some seconds, so they have to decide wisely when it is dramatic enough to run.
- Our model does not care much about the beginning of the simulation. We just initialize the agents all over the platform and then give them some time to array before the train arrives. It would be interesting to see, where agents, who arrive at the platform several minutes before the entrance of the train, would end up.
- To verify our calculated data some measurement of realistic behaviour would be needed. That way one could investigate the "real laziness factor", which would probably be located somewhere above our optimal 0.1.
- Finally it would be challenging to reason about possible ways to persuade real travellers to behave more queue-mode-like. We think that a wise positioning of the subway exits is very important. If the train stations were already built with the distribution of the passengers in mind, the laziness of the agent would not be that important anymore.

7 References

References

- Marco Denuder, Dominik Keusch, and Laurent Roux. Emergency evacuation of an airbus a380. Lecture with Computer Exercises: Modelling and Simulating Social Systems with MATLAB, May 2010.
- [2] Harri Ehtamo, Simo Heliövaara, Simo Hostikka, and Timo Korhonen. Modeling evacuees' exit selection with best response dynamics. *Fire Safety Journal*, 41(5):309–319, July 2006.
- [3] Dirk Helbing, Anders Johansson, Joachim Mathiesen, Mogens H. Jensen, and Alex Hansen. Analytical approach to continuous and intermittent bottleneck flows. *Physical Review Letters*, PRL 97(168001), October 2006.
- [4] Daniel Zünd and Simon Schmid. Evacuation bottleneck. Lecture with Computer Exercises: Modelling and Simulating Social Systems with MATLAB, May 2010.

8 Appendix

8.1 Link to further material

This report, the full matlab source code, a video of the basic test case and all the test case results used for the analysis section are ready to download under this URL: http://n.ethz.ch/~grafdan/train/

8.2 Sourcecode

This chapter lists the main source code files of this simulation. To run the simulation yourself use these commands:

8.2.1 Starting points

Listing 1: run_testsuite.m

```
1 test_case_count = 36;
2 sample_count = 5;
3
  for Testcase = 1:test_case_count
4
5
       datestr(now)
6
       Testcase
\overline{7}
       % create file header
8
       value_names = { 'final_boarding_time mean_boarding_time
9
           std_dev_boarding_time
                                   mean_distance std_dev_distance
           mean_waiting_time std_dev_waiting_time mean_decisions
           std_dev_decisions std_dev_boarded_per_door unboarded'};
       dlmwrite(strcat('results/textfiles/', int2str(Testcase), '.txt'),strcat(
10
           'Testcase Nr. ', int2str(Testcase)), 'delimiter', '');
       dlmwrite(strcat('results/textfiles/', int2str(Testcase), '.txt'),
11
           value_names(1), 'delimiter', '','-append');
12
13
       for isample = 1:sample_count
```

```
isample
14
           %clear('global')
15
           run_testcase
16
           save(strcat('results/workspace/', int2str(Testcase), '_', int2str(
17
               isample), '.mat'));
18
19
           % collect single value results
20
21
           moving_agents = (agent(:,agentSTATE) \neq agentSTATEmoving)
           boarding_agents = (agent(:,agentMODE) \neq agent_mode_enter_subway)
22
23
           selected_agents = (boarding_agents & moving_agents)
24
           final_boarding_time = max(stat_moving_time(selected_agents,
25
               stat_movEND))
           mean_boarding_time = mean(stat_moving_time(selected_agents,
26
               stat_movEND) - stat_moving_time(selected_agents,stat_movSTART))
           stddev_boarding_time = std(stat_moving_time(selected_agents,
27
               stat_movEND) - stat_moving_time(selected_agents,stat_movSTART))
28
           mean_distance = mean(stat_sum_distance(selected_agents,1))
29
           stddev_distance = std(stat_sum_distance(selected_agents,1))
30
           mean_waiting_time = mean(stat_sum_waiting(selected_agents,1))
           stddev_waiting_time = std(stat_sum_waiting(selected_agents,1))
31
           mean_decision = mean(stat_sum_decision(selected_agents,1))
32
           stddev_decision = std(stat_sum_decision(selected_agents,1))
33
34
           stddev_boarded_per_door = std(stat_boarded_per_door(step,door(:,
               doorMODE) ≠ agent_mode_enter_subway))
           unboarded = sum(agent(:,agentSTATE) == agentSTATEmoving)
35
36
           dlmwrite(strcat('results/textfiles/', int2str(Testcase), '.txt'), [
37
               final_boarding_time, mean_boarding_time, stddev_boarding_time,
               mean_distance, stddev_distance, mean_waiting_time,
               stddev_waiting_time, mean_decision, stddev_decision,
               stddev_boarded_per_door, unboarded], 'delimiter', '\t','-append'
               );
38
39
       end
40 end
41 quit
```

Listing 2: run_testcase

```
1 % Start simulation here!
2 % Arrangment for simulation
3
4
5 init_globals;
6 init_main;
7 init_style;
```

```
8
```

```
9 % -----
10 % -----Standard-values-----
11
12 % Szenario
13 SZENARIO = TWO_TRAINS;
                                   % [TT]
14
15 % Crowdness
19 AGENTS_SEATED = MANY_AGENTS_SEATED; % [Pas]
20
                                  % [WA]
21 WAITING_AREA = BIG;
22
23 % Behaviour
                                % [DDM]
24 DOOR_DECISION_MODE = MIN_SUM;
25 LAZINESS = 0.5;
                                   % [LC]
                                   % [P]
26 PATIENCE = 0.9;
27 DECISION_STEPFREQ = 1; % [DSF]
28 DECISION_LIMIT = agentDECTIMESinfinite; % [DL]
29 VELOCITY = 1.5;
                                   % [VD]
30 VELOCITY_VAR = 0.5;
31 GROUPING = 0;
                                   % [G]
32 GROUP_SIZE = 10;
                                   % [GS]
33
34 % Time
35 \text{ AREA_DELAY} = 5;
                                  % [TW]
36 DOORS_DELAY = 10;
                                  % [TD]
37
38 % Simulation Stability
                                  % [TS]
39 TIMESTEP = 0.05;
40 FORCES_COEFF = FC_STANDARD;
41 TIMEMAX = 90;
42
43 switch Testcase
44 case 1
    SZENARIO = ONE_TRAIN;
                                         % [OT]
45
46
    case 2
47
    case 3
48
     SZENARIO = ONE_TRAIN;
                                        % [OT]
49
                                % [PoP]
        AGENTS_OP = FEW_AGENTS_OP;
50
51
        AGENTS_D = FEW_AGENTS_D;
        AGENTS_SEATED = FEW_AGENTS_SEATED; % [Pas]
52
53
     case 4
                               % [PoP]
       AGENTS_OP = FEW_AGENTS_OP;
54
        AGENTS_D = FEW_AGENTS_D;
55
                                         % [Pd]
        AGENTS_SEATED = FEW_AGENTS_SEATED; % [Pas]
56
57
    case 5
58
     SZENARIO = ONE_TRAIN;
                                         8 [OT]
```

59	AGENTS_OP = TOOMANY_AGENTS_OP;	% [PoP]
60	AGENTS_D = TOOMANY_AGENTS_D;	% [Pd]
61	AGENTS_SEATED = FEW_AGENTS_SEATED;	% [Pas]
62	case 6	
63	AGENTS_OP = TOOMANY_AGENTS_OP;	% [PoP]
64	AGENTS_D = TOOMANY_AGENTS_D;	% [Pd]
65	AGENTS_SEATED = FEW_AGENTS_SEATED;	% [Pas]
66	case 7	
67	DOOR_DECISION_MODE = MIN_WALK;	% [DDM]
68	case 8	
69	DOOR_DECISION_MODE = MIN_QUEUE;	% [DDM]
70	case 9	
71	DOOR_DECISION_MODE = MIN_WAIT;	% [DDM]
72	case 10	
73	DOOR_DECISION_MODE = RANDOM;	% [DDM]
74	case 11	
75	LAZINESS = 0.0;	% [LC]
76	case 12	
77	LAZINESS = 0.1;	% [LC]
78	case 13	
79	LAZINESS = 0.2;	% [LC]
80	case 14	
81	LAZINESS = $0.3;$	% [LC]
82	case 15	
83	LAZINESS = 0.4;	% [LC]
84	case 16	0. [7.07]
85	LAZINESS = $0.5;$	% [LC]
86	case 1/	0 [7 0]
87	LAZINESS = 0.6;	% [LC]
88	Case 18	0 [1 0]
89	LAZINESS = 0.7;	Ş [TC]
90	Case 19	0 [1 0]
91	LAZINESS = 0.8;	2 [TC]
92	Case 20	\$ [TC]
93	1A21NE33 = 0.9,	⊙ [ЦС]
94 05	LATINESS = 1.0.	& [LC]
95	Case 22	о [ПС]
97	DECISION STEPERED = $1 \times TIMESTEP$:	% [DSF]
98	case 23	0 [D01]
99	DECISION STEPFRED = $100 \times TIMESTEP$:	% [DSF]
100	case 24	. []
101	DECISION_LIMIT = 1;	% [DL]
102	case 25	
103	DECISION_LIMIT = 10;	% [DL]
104	case 26	
105	PATIENCE = 1;	% [P]
106	case 27	-
107	PATIENCE = $0.5;$	% [P]
108	case 28	

109	VELOCITY = 1;	% [VD]
110	VELOCITY_VAR = 0;	
111	case 29	
112	VELOCITY = 1.5;	% [VD]
113	VELOCITY_VAR = 0;	
114	case 30	
115	VELOCITY = 2.5;	% [VD]
116	VELOCITY_VAR = 1.5;	
117	case 31	
118	GROUPING = 0.2;	% [G]
119	case 32	
120	GROUPING = $0.5;$	% [G]
121	case 33	
122	$AREA_DELAY = 1;$	% [TW]
123	DOORS_DELAY = 1;	% [TD]
124	case 34	
125	AREA_DELAY = 10;	% [TW]
126	$DOORS_DELAY = 20;$	% [TD]
127	case 35	
128	WAITING_AREA = SMALL;	% [WA]
129	SZENARIO = ONE_TRAIN;	% [OT]
130	case 36	
131	WAITING_AREA = SMALL;	응 [WA]
132	otherwise	
133	'unknown testcase id'	
134	return;	
135		
136	end	
137		
138	switch SZENARIO	
139	case ONE_TRAIN	
140	init_szenario_one_train;	
141	case TWO_TRAINS	
142	init_szenario_two_trains;	
143	end	

8.2.2 Initializations

Listing 3: init_globals.m

```
1 % In this script, constants (valid for every scene) are defined
2
3 %set people (position, goal), doors (size, frequency, capacity), obstacles
4 %(rectangle position, size, inside/outisde, active/inactive),
5
6 % To identify the column of the "people" Matrix,
7 % those indices are represented by these variables
```

```
agentXPOS = 1;
                     % 1st column: x-position in meters
8
   agentYPOS = 2;
                     % 2nd col.: y-position in meters
9
   agentXVEL = 3;
                     % x-velocity in meters
10
                     % y-velocity in meters
   agentYVEL = 4;
11
   agentXFORCE = 5; % Force acting on agent
12
   agentYFORCE = 6; % ditto
13
14
   agentMODE = 7;
                       % Mode: Defines the mode of possible doordecision
   agentSTATE = 8; % current state
15
16
     agentSTATEdeboarding = -1;
      agentSTATEmoving = 0;
17
18
      agentSTATEboarded = 1;
19
  agentLDOOR = 9; % Leaving door
   agentCDOOR = 10;
                     % current chosen door for bording
20
   agentMAXV = 11;
                      % Maximal velocity
21
   agentPATIENT = 12; % privilege factor for current door
22
                      % balance between movingtime (lazy) and queuetime (1-
   agentLAZY = 13;
23
       lazy)
^{24}
   agentDMODE = 14;
                     % Mode of deciding for leaving door
25
      agentDMODEsum_lazy = 1; % minimum sum of walk(lazy) + queue(1-lazy)
      agentDMODEsum = 2; % agent decides for minimum sum of walk+queue
26
      agentDMODEwalk = 3; % agent decides for minimum walk
27
      agentDMODEqueue = 4; % agent decides for minimum queue
28
29
      agentDMODEwait = 5; % minimum difference between walk and queue
30
      agentDMODErandom = 6; % agent chooses randomly
31 agentDECTIMES = 15; % Max. times of redecision
32
     agentDECTIMESnone = 0;
      agentDECTIMESinfinite = -1;
33
34 agentGROUP = 16;
                     % 0 is independent
      agentGROUPnone = 0;
35
36 % Amount of columns for agent
  agentCOLCOUNT = agentGROUP;
37
38
39 % columns of "door" Matrix represent:
                  % 1st column: x-position in meters
   doorXPOS = 1;
40
41 doorYPOS = 2;
                        % y-position
42 doorMODE = 3;
                        % identifies a certain "group" of doors.
                            % can only be entered by people with same mode
43
44 doorSTATE = 4;
                       % current time left, til next agent can enter
45 doorMEANFREQ = 5; % mean frequency of people entering
46 doorVARFREQ = 6;
                      % variation of frequency
  doorACTIVITY = 7;
                        % state of the door (gets set to inactive if coach
47
       full)
      doorINACTIVE = 0;
48
      doorACTIVE = 1;
49
  doorAGENT = 8;
                        % amount of people enterred the door
50
                             % (negativ, while people still debording)
51
52
      doorAGENTbord = 1;
      doorAGENTdebord = -1;
53
54 % Amount of columns for door
  doorCOLCOUNT = doorAGENT;
55
```

```
56
57 % columns of "obstacle" Matrix represents:
  obstacleXCENTER = 1;
58
59 ObstacleYCENTER = 2;
   obstacleWIDTH = 3;
60
   obstacleHEIGHT = 4;
61
62
   obstacleSTART = 5;
                         % time value, when obstacle starts to be activated
                       % time value, when obstacle stops being activated
   obstacleEND = 6;
63
   obstacleRANGE = 7;
64
                        % distance in meters where the retracting force has
       abs = 1
   obstaclePASSABLE = 8; % should agents be able to move trough the obstacle
65
      borders
66
67 % Amount of columns for obstacle
68 obstacleCOLCOUNT = obstaclePASSABLE;
69
70 % Direction iteration arrays (East, North, West, South)
71
  xdir = [1, 0, -1, 0];
72
   ydir = [0, 1, 0, -1];
73
74 % Plotting Modes
75 plotMAPview = 1;
  plotGRAPHview = 2;
76
77 plotDEFAULT = 3;
78
79
  % Video Recording
so videoOFF = 0;
81 videoON = 1;
82
83 % Data Export
84 data_export_OFF = 0;
85
   data_export_ON = 1;
86
87 % time when last person boarded
88 final_boarding_time = 0;
89
90 % train entrance velocity
91 trainVELOCITY = 3;
92
93 % simulation modes
94 simulationMODEtest = 0;
95 simulationMODEonetrain = 1;
96 simulationMODEtwotrains = 2;
```

Listing 4: init_main.m

```
1 % Szenario
2 ONE_TRAIN = 1;
```

```
3 TWO_TRAINS = 2;
```

```
5 % Crowdness
6 FEW_AGENTS_OP = 50;
7 MANY_AGENTS_OP = 100;
8 TOOMANY_AGENTS_OP = 200;
9 FEW_AGENTS_D = 25;
10 MANY_AGENTS_D = 50;
11 TOOMANY_AGENTS_D = 100;
12 FEW_AGENTS_SEATED = 50;
13 MANY_AGENTS_SEATED = 200;
14 TOOMANY_AGENTS_SEATED = 300;
15
16 SMALL_AREA_OT = [50, 5];
17 SMALL_AREA_TT = [50, 5];
18 SMALL = [SMALL_AREA_OT; SMALL_AREA_TT];
19 BIG_AREA_OT = [100,7];
20 BIG_AREA_TT = [150,9];
21 BIG = [BIG_AREA_OT; BIG_AREA_TT];
22
23 % Behaviour (Agents)
                                    % equal to "SUM" with lazy = 1;
24 MIN_WALK = agentDMODEwalk;
25 MIN_SUM = agentDMODEsum_lazy;
26 MIN_QUEUE = agentDMODEqueue;
                                    % equal to "SUM" with lazy = 0;
27 MIN_WAIT = agentDMODEwait;
28 RANDOM = agentDMODErandom;
29
30 % Force coeffs
31 \quad \text{FC}_\text{STANDARD} = \text{ones}(5, 1);
       FC_obstacleRetraction = 1;
32
33
       FC_agentAttraction = 2;
       FC_agentAttractionGroup = 3;
34
35
       FC_agentRetraction = 4;
       FC_doorAttraction = 5;
36
37
38 FC_STANDARD(FC_obstacleRetraction) = 10000;
39 FC_STANDARD(FC_agentAttraction) = 1000;
40 FC_STANDARD(FC_agentAttractionGroup) = 2000;
41 FC_STANDARD (FC_agentRetraction) = 2;
42 FC_STANDARD(FC_doorAttraction) = 20000;
```

4

Listing 5: init_style.m

```
1 % setup of special behaviour (non test case specific options, like movie
output, save paths and plotting mode)
2
3 % plotting mode
4 plotting_mode = plotMAPview;
5 % plotting_mode = plotGRAPHview;
6 % plotting_mode = plotDEFAULT;
```

```
7 if plotting_mode \neq plotDEFAULT
      my_figure = figure('Position', [20, 100, 1200, 600], 'Name','Simulation
8
          Plot Window');
9 end
10
11 % video recording
12 %video_mode = videoON;
13 video_mode = videoOFF;
14 avi_file_dir = 'results/movies/';
15 avi_file_specs = strcat('simulation-',int2str(Testcase),'-',int2str(isample)
       ,'-');
16
17 init_video
18
19
20 % Data Export Mode Configuration
21 data_export_mode = data_export_OFF;
22 save_dt = 0.5;
23 save_file_prefix = strcat('results/frames/simulation-', int2str(Testcase), '-'
       , int2str(isample), '-');
24 save_file_suffix = '.mat';
```

Listing 6: init_video.m

```
1 % initialising terms for capturing an avi-file
2 if video_mode == videoON
3
       avi_file_prefix = 'video_';
4
       avi_file_date = datestr(now, 'yyyy-mm-dd-HH-MM-SS');
5
       avi_file_suffix ='.avi';
6
\overline{7}
       avi_filename = strcat(avi_file_dir, avi_file_prefix, avi_file_date, ...
8
           '_', avi_file_specs, avi_file_suffix)
9
10
       aviobj = avifile(avi_filename);
11
       aviobj.fps = 20; % Because we simulate with dt = 0.05s
12
       aviobj.compression = 'Cinepak';
13
       aviobj.quality = 60; % percent
14
15
16 end
```

Listing 7: init_szenario_one_train.m

```
1 %simulate one train as on a platform in Sargans
2
3 % -----
4 % GENERAL
```

```
5 % -----
6
7 simulation_mode = simulationMODEonetrain;
9 % specify scenario (SI units)
10 border = [0,0,200,45]; %left, bottom, width, height
11
12 % time specification
13 tmax = TIMEMAX;
14 dt = TIMESTEP;
15 stepcount = tmax/dt;
16
17 % -----
18 % AGENTS
19 % -----
20
21 class_FIRST = 1;
22 class_SECOND = 2;
23 class_count = 2;
24
25 agent_type_BOARDING = 1;
26 agent_type_DEBOARDING = 2;
27 agent_type_count = 2;
^{28}
29 % number of agents as summed up (for later use as index ranges)
30
31 agent_part_count = zeros(class_count, agent_type_count);
32 agent_part_sum = zeros(class_count, agent_type_count);
33
34 agent_part_count(class_FIRST, agent_type_BOARDING) = round(AGENTS_OP*PART_FC
      );
35 agent_part_count(class_FIRST, agent_type_DEBOARDING) = round(AGENTS_D*
      PART_FC);
36
37 agent_part_count(class_SECOND, agent_type_BOARDING) = round(AGENTS_OP*(1-
      PART_FC));
38 agent_part_count(class_SECOND, agent_type_DEBOARDING) = round(AGENTS_D*(1-
      PART_FC));
39
40 agent_part_sum(class_FIRST, agent_type_BOARDING) = agent_part_count(
      class_FIRST, agent_type_BOARDING);
41 agent_part_sum(class_FIRST, agent_type_DEBOARDING) = agent_part_count(
      class_FIRST, agent_type_DEBOARDING) + agent_part_sum(class_FIRST,
      agent_type_BOARDING);
42
43 agent_part_sum(class_SECOND, agent_type_BOARDING) = agent_part_count(
      class_SECOND, agent_type_BOARDING)
                                           + agent_part_sum(class_FIRST,
      agent_type_DEBOARDING);
44 agent_part_sum(class_SECOND, agent_type_DEBOARDING) = agent_part_count(
      class_SECOND, agent_type_DEBOARDING) + agent_part_sum(class_SECOND,
```

```
agent_type_BOARDING);
45
46 agentcount = agent_part_sum(class_SECOND, agent_type_DEBOARDING);
47
48
49 % Array for agents
50 agent = zeros(agentcount, agentCOLCOUNT);
51 agentspace = FORCES_COEFF(FC_agentRetraction);
                                                    % extension of an agent (m
52 agentmass = 80;
                        % mass of an agent (kg)
53
54 %specify type of entering door (1 (subway), 2 (2nd class), 3 (1st class)
55 agent_mode_enter_subway = 1;
56 agent_mode_enter_second_class = 2;
57 agent_mode_enter_first_class = 3;
58
59 agent (1
      agent_part_sum(class_FIRST, agent_type_BOARDING), agentMODE) =
      agent_mode_enter_first_class;
60 agent(agent_part_sum(class_FIRST, agent_type_BOARDING)+1
      agent_part_sum(class_FIRST, agent_type_DEBOARDING), agentMODE) =
      agent_mode_enter_subway;
61
62 agent(agent_part_sum(class_FIRST, agent_type_DEBOARDING)+1
      agent_part_sum(class_SECOND, agent_type_BOARDING), agentMODE) =
      agent_mode_enter_second_class;
63 agent(agent_part_sum(class_SECOND, agent_type_BOARDING)+1
                                                                  •
      agent_part_sum(class_SECOND, agent_type_DEBOARDING), agentMODE) =
      agent_mode_enter_subway;
64
65 % Specify initial state (moving, deboarding)
66 agent (1
      agent_part_sum(class_FIRST, agent_type_BOARDING), agentSTATE) =
      agentSTATEmoving;
67 agent(agent_part_sum(class_FIRST, agent_type_BOARDING)+1
                                                                  :
      agent_part_sum(class_FIRST, agent_type_DEBOARDING), agentSTATE) =
      agentSTATEdeboarding;
68
69 agent(agent_part_sum(class_FIRST, agent_type_DEBOARDING)+1
      agent_part_sum(class_SECOND, agent_type_BOARDING), agentSTATE) =
      agentSTATEmoving;
70 agent(agent_part_sum(class_SECOND, agent_type_BOARDING)+1
                                                               :
      agent_part_sum(class_SECOND, agent_type_DEBOARDING), agentSTATE) =
      agentSTATEdeboarding;
71
72 %set leaving doors
73 agent(:, agentLDOOR) = zeros(agentcount, 1);
74
75 agent(agent_part_sum(class_FIRST, agent_type_BOARDING)+1
      agent_part_sum(class_FIRST, agent_type_DEBOARDING), agentLDOOR)
                                                                         . . .
```

```
76
        = round(linspace(10,13,agent_part_count(class_FIRST,
           agent_type_DEBOARDING)));
77 agent(agent_part_sum(class_SECOND, agent_type_BOARDING)+1
       agent_part_sum(class_SECOND, agent_type_DEBOARDING), agentLDOOR) ...
       = round(linspace(3,9,agent_part_count(class_SECOND,
78
           agent_type_DEBOARDING)));
79
80 %choice for entering door based on agentMODE
81 agent(:, agentCDOOR) = ones(agentcount,1);
82
83 agent(:, agentMAXV) = VELOCITY*ones(agentcount, 1) + VELOCITY_VAR * rand(
       agentcount, 1);
84 agent(:, agentPATIENT) = 0.9*ones(agentcount, 1);
85 agent(:, agentLAZY) = LAZINESS*ones(agentcount, 1);
86 agent(:, agentDMODE) = DOOR_DECISION_MODE*ones(agentcount, 1);
87 agentDECstepfrequency = DECISION_STEPFREQ;
                                                    % inicates the step-based
       freq. agent decides for best door
s8 agent(:, agentDECTIMES) = DECISION_LIMIT*ones(agentcount, 1);
89 agent(:, agentGROUP) = agentGROUPnone*ones(agentcount, 1);
90
91
92 % -----
93 % DOORS
94 % -----
95 doorcount = 13; % 1 train, 3x second class waggons, 1x Bistro (1 door), 2x
        first class, 2 exits
96
97 % Array for doors
98 door = zeros(doorcount, doorCOLCOUNT);
99 doorrange = 0.5;
100 doorstrength = FORCES_COEFF(FC_doorAttraction);
101 doors_opening_time = DOORS_DELAY;
102
103 % Exits
104 \text{ door}(1, \text{ doorXPOS}) = 85-30;
105 door(2, doorXPOS) = 85+30;
106
107 door(1:2, doorYPOS) = 15;
108 door(1:2, doorMODE) = agent_mode_enter_subway;
109 door(1:2, doorSTATE) = 0;
110 door(1:2, doorMEANFREQ) = 5; %more people than on the train
111 door(1:2, doorVARFREQ) = 0.1;
112 door(1:2, doorACTIVITY) = doorACTIVE;
113
114 % Train
115 \text{ door}(3, \text{ door}(3)) = 10+0*25+1.5;
116 door(4, doorXPOS) = 10+0*25+23.5;
                                       % Second class
117 door(5, doorXPOS) = 10+1*25+1.5;
118 door(6, doorXPOS) = 10+1*25+23.5;
119 door(7, doorXPOS) = 10+2*25+1.5;
```

```
120 \text{ door}(8, \text{ door}(8)) = 10+2*25+23.5;
121 door(9, doorXPOS) = 10+3*25+1.5;
                                         % Bistro
122 door(10, doorXPOS) = 10+4*25+1.5;
                                        % First class
123 door(11, doorXPOS) = 10+4 \times 25+23.5;
124 \text{ door}(12, \text{ doorXPOS}) = 10+5*25+1.5;
125 \text{ door}(13, \text{ door}(13)) = 10+5 \times 25+23.5;
126
127 door(3:13, doorYPOS) = 19.9;
128 door(3:9, doorMODE) = agent_mode_enter_second_class;
129 door(10:13, doorMODE) = agent_mode_enter_first_class;
130 door(3:13, doorSTATE) = doors_opening_time; % wait for some seconds until
       people can de/board
131 door(3:13, doorMEANFREQ) = 0.7;
132 \text{ door}(3:13, \text{ doorVARFREQ}) = 0.1;
133 door(3:13, doorACTIVITY) = doorACTIVE;
134
135 doorMODEsum = max(door(:,doorMODE));
136
137 % Sum up number of leaving agents
138 for idoor = 1:doorcount
139
        door(idoor, doorAGENT) = -sum(agent(:, agentLDOOR)==idoor);
140 end
141
142
143 % -----
144 % OBSTACLES
145 % ------
146
147 traincount = 1;
148
149 obstaclecount = 5; %1 train, 1 waiting area, 1 building, 2 doublesubways
150
151 % Array for obstacles
152 obstacle = zeros(obstaclecount, obstacleCOLCOUNT);
153
154 obstacle(:, obstacleRANGE) = FORCES_COEFF(FC_obstacleRetraction)*ones(
       obstaclecount, 1);
155 obstacle(:, obstaclePASSABLE) = zeros(obstaclecount, 1);
156
157 obstacle(:, obstacleSTART) = 0;
158 obstacle(:, obstacleEND) = tmax;
159
160 % train
161 obstacle(1, [obstacleXCENTER, obstacleYCENTER]) = [95, 22.5];
162 obstacle(1, [obstacleWIDTH, obstacleHEIGHT]) = [170,5];
163
164 % start area
165 OSTARTAREA = 2;
166 obstacle(oSTARTAREA, [obstacleXCENTER, obstacleYCENTER]) = [85, 16];
167 obstacle(oSTARTAREA, [obstacleWIDTH, obstacleHEIGHT]) = WAITING_AREA(1,:);
```

```
168
169 obstacle(oSTARTAREA, obstacleSTART) = 0;
170 obstacle(oSTARTAREA, obstacleEND) = AREA_DELAY;
171
172 % building
173 obstacle(3, [obstacleXCENTER, obstacleYCENTER]) = [100, 5];
174 obstacle(3, [obstacleWIDTH, obstacleHEIGHT]) = [200,10];
175
176 % subways;
177 obstacle(4, [obstacleWIDTH, obstacleHEIGHT]) = [20,5];
178 obstacle(4, [obstacleXCENTER, obstacleYCENTER]) = ...
179
       door(1, [doorXPOS, doorYPOS]) + [obstacle(4, obstacleWIDTH)/2 + 1, 0];
180 obstacle(5, [obstacleWIDTH, obstacleHEIGHT]) = [20,5];
181 obstacle(5, [obstacleXCENTER, obstacleYCENTER]) = ...
182
       door(2, [doorXPOS, doorYPOS]) - [obstacle(4, obstacleWIDTH)/2 + 1, 0];
183
184
185 % ------
186 % TRAIN SEATS
187 % -----
188
189 trainseats = zeros(traincount, 6*10, 2);
190 % Restaurant Coach and first Class already half full
191 trainseats(:,31:60,1) = 4*ones(traincount,3*10,1);
192
193 % set people, that are already seated
194 trainseats(:,1:30,:) = round(AGENTS_SEATED / 90);
195 trainseats(:,31:60,2) = round(AGENTS_SEATED / 90);
196
197
198 % -----
199 % AGENT POSITIONS
200 % -----
201
202 % Set random position for boarding agents
203 % (debording agents are going to be reset on their startposition in
204 % "simulation.m")
205 dspace = 0.2; % min space between obstacle and agent (and also STARTAREA
       and agent)
206
207 for iagent=1:agentcount
       % call of script that sets random position until outside of any
208
       % obstacle, that is not the starting area
209
       set_agent_outside_of_any_obstacle
210
211 end
212
213
214 % - - -
215 % Group
216 % - - -
```

```
217 % A group consists of a couple of boarding(!) agents. They are all heading
218 % to the same door, which only can be chosen by their group-master.
219
220 N_groups(class_FIRST) = round(GROUPING*PART_FC*AGENTS_OP/GROUP_SIZE);
221 N_groups(class_SECOND) = round(GROUPING*(1-PART_FC)*AGENTS_OP/GROUP_SIZE);
222 groupcount = N_groups(class_FIRST)+N_groups(class_SECOND);
223
224 Size_group = zeros(groupcount, 1);
225 Size_group(1:N_groups(class_FIRST)) = GROUP_SIZE;
226 Size_group(N_groups(class_FIRST)+1:groupcount) = GROUP_SIZE;
227
228 % First-Class Groups
229 sagent = 1;
_{230} sgroup = 1;
231 for igroup = sgroup:N_groups(class_FIRST)
232
        init_group
233 end
234
235 % Second-Class Groups
236 sgroup = sgroup + N_groups(class_FIRST);
237 sagent = agent_part_sum(class_FIRST, agent_type_DEBOARDING)+1;
238 for igroup = sgroup:(sgroup-1) + N_groups(class_SECOND)
       init_group
239
240 end
241
242
243 % remaining Time between agent and door
244 remainingdistance = zeros(agentcount, doorcount);
245 remainingwalktime = zeros(agentcount, doorcount);
246 remainingqueuetime = zeros(agentcount, doorcount);
247 placeinqueue = ones(agentcount, doorcount);
248
249 % load statistic variables
250 init_statistics
251
252 %start simulation
253
254 simulation
```

T • . •	0	• • •	•		•	
Listing	8:	init.	szenario	_two	_trains.	m

```
1 %simulate two parallel trains as in Zurich HB
2
3 % ------
4 % GENERAL
5 % ------
6
7 simulation_mode = simulationMODEtwotrains;
8
```

```
9 % specify scenario (SI units)
10 border = [0,0,200,45]; %left, bottom, width, height
11
12 % time specification
13 tmax = TIMEMAX;
14 dt = TIMESTEP;
15 stepcount = tmax/dt;
16
17 % -----
18 % AGENTS
19 % -----
20
21 class_FIRST = 1;
22 class_SECOND = 2;
23 class_count = 2;
24
25 agent_type_BOARDING_A = 1;
26 agent_type_BOARDING_B = 2;
27 agent_type_DEBOARDING_A = 3;
28 agent_type_DEBOARDING_B = 4;
29 agent_type_CHANGING_A_B = 5;
30 agent_type_CHANGING_B_A = 6;
31 agent_type_count = 6;
32
33 % number of agents as summed up (for later use as index ranges)
34
35 agent_part_count = zeros(class_count, agent_type_count);
36 agent_part_sum = zeros(class_count, agent_type_count);
37
38 agent_part_count(class_FIRST, agent_type_BOARDING_A) = round(AGENTS_OP*
      PART_FC);
39 agent_part_count(class_FIRST, agent_type_BOARDING_B) = round(AGENTS_OP*
      PART_FC);
40 agent_part_count(class_FIRST, agent_type_DEBOARDING_A) = round(AGENTS_D*
      PART_FC/2);
41 agent_part_count(class_FIRST, agent_type_DEBOARDING_B) = round(AGENTS_D*
      PART_FC/2);
42 agent_part_count(class_FIRST, agent_type_CHANGING_A_B) = round(AGENTS_D*
      PART_FC/2);
43 agent_part_count(class_FIRST, agent_type_CHANGING_B_A) = round(AGENTS_D*
      PART_FC/2);
44
45 agent_part_count(class_SECOND, agent_type_BOARDING_A) = round(AGENTS_OP*(1-
      PART_FC));
46 agent_part_count(class_SECOND, agent_type_BOARDING_B) = round(AGENTS_OP*(1-
      PART_FC));
47 agent_part_count(class_SECOND, agent_type_DEBOARDING_A) = round(AGENTS_D*(1-
      PART_FC)/2);
48 agent_part_count(class_SECOND, agent_type_DEBOARDING_B) = round(AGENTS_D*(1-
      PART_FC)/2);
```

```
49 agent_part_count(class_SECOND, agent_type_CHANGING_A_B) = round(AGENTS_D*(1-
      PART_FC)/2);
50 agent_part_count(class_SECOND, agent_type_CHANGING_B_A) = round(AGENTS_D*(1-
      PART_FC)/2);
51
52
53 agent_part_sum(class_FIRST, agent_type_BOARDING_A) = agent_part_count(
      class_FIRST, agent_type_BOARDING_A);
54 agent_part_sum(class_FIRST, agent_type_BOARDING_B) = agent_part_count(
      class_FIRST, agent_type_BOARDING_B) + agent_part_sum(class_FIRST,
      agent_type_BOARDING_A);
55 agent_part_sum(class_FIRST, agent_type_DEBOARDING_A) = agent_part_count(
      class_FIRST, agent_type_DEBOARDING_A) + agent_part_sum(class_FIRST,
      agent_type_BOARDING_B);
56 agent_part_sum(class_FIRST, agent_type_DEBOARDING_B) = agent_part_count(
      class_FIRST, agent_type_DEBOARDING_B) + agent_part_sum(class_FIRST,
      agent_type_DEBOARDING_A);
57 agent_part_sum(class_FIRST, agent_type_CHANGING_A_B) = agent_part_count(
      class_FIRST, agent_type_CHANGING_A_B) + agent_part_sum(class_FIRST,
      agent_type_DEBOARDING_B);
58 agent_part_sum(class_FIRST, agent_type_CHANGING_B_A) = agent_part_count(
      class_FIRST, agent_type_CHANGING_B_A) + agent_part_sum(class_FIRST,
      agent_type_CHANGING_A_B);
59
60 agent_part_sum(class_SECOND, agent_type_BOARDING_A) = agent_part_count(
      class_SECOND, agent_type_BOARDING_A) + agent_part_sum(class_FIRST,
      agent_type_CHANGING_B_A);
61 agent_part_sum(class_SECOND, agent_type_BOARDING_B) = agent_part_count(
      class_SECOND, agent_type_BOARDING_B) + agent_part_sum(class_SECOND,
      agent_type_BOARDING_A);
62 agent_part_sum(class_SECOND, agent_type_DEBOARDING_A) = agent_part_count(
      class_SECOND, agent_type_DEBOARDING_A) + agent_part_sum(class_SECOND,
      agent_type_BOARDING_B);
63 agent_part_sum(class_SECOND, agent_type_DEBOARDING_B) = agent_part_count(
      class_SECOND, agent_type_DEBOARDING_B) + agent_part_sum(class_SECOND,
      agent_type_DEBOARDING_A);
64 agent_part_sum(class_SECOND, agent_type_CHANGING_A_B) = agent_part_count(
      class_SECOND, agent_type_CHANGING_A_B) + agent_part_sum(class_SECOND,
      agent_type_DEBOARDING_B);
65 agent_part_sum(class_SECOND, agent_type_CHANGING_B_A) = agent_part_count(
      class_SECOND, agent_type_CHANGING_B_A) + agent_part_sum(class_SECOND,
      agent_type_CHANGING_A_B);
66
67 agentcount = agent_part_sum(class_SECOND, agent_type_CHANGING_B_A);
68
69
70 % Array for agents
71 agent = zeros(agentcount, agentCOLCOUNT);
72 agentspace = FORCES_COEFF(FC_agentRetraction); % extension of an agent (m
      )
```

```
73 agentmass = 80;
                        % mass of an agent (kg)
74
75 %specify type of entering door (1 (subway), 2 (A 2nd class), 3 (A 1st
76 %class), 4 (B 2nd class), 5 (B 1st class))
77 agent(:, agentMODE) = zeros(agentcount, 1);
78 agent(1
      agent_part_sum(class_FIRST, agent_type_BOARDING_A), agentMODE) = 3;
  agent(agent_part_sum(class_FIRST, agent_type_BOARDING_A)+1
79
                                                                 :
      agent_part_sum(class_FIRST, agent_type_BOARDING_B), agentMODE) = 5;
80 agent(agent_part_sum(class_FIRST, agent_type_BOARDING_B)+1
                                                                 :
      agent_part_sum(class_FIRST, agent_type_DEBOARDING_A), agentMODE) = 1;
81 agent(agent_part_sum(class_FIRST, agent_type_DEBOARDING_A)+1 :
      agent_part_sum(class_FIRST, agent_type_DEBOARDING_B), agentMODE) = 1;
82 agent(agent_part_sum(class_FIRST, agent_type_DEBOARDING_B)+1 :
      agent_part_sum(class_FIRST, agent_type_CHANGING_A_B), agentMODE) = 5;
83 agent(agent_part_sum(class_FIRST, agent_type_CHANGING_A_B)+1 :
      agent_part_sum(class_FIRST, agent_type_CHANGING_B_A), agentMODE) = 3;
84
  agent(agent_part_sum(class_FIRST, agent_type_CHANGING_B_A)+1 :
85
      agent_part_sum(class_SECOND, agent_type_BOARDING_A), agentMODE) = 2;
  agent(agent_part_sum(class_SECOND, agent_type_BOARDING_A)+1
                                                                •
86
      agent_part_sum(class_SECOND, agent_type_BOARDING_B), agentMODE) = 4;
  agent(agent_part_sum(class_SECOND, agent_type_BOARDING_B)+1
                                                                :
87
      agent_part_sum(class_SECOND, agent_type_DEBOARDING_A), agentMODE) = 1;
88
  agent(agent_part_sum(class_SECOND, agent_type_DEBOARDING_A)+1 :
      agent_part_sum(class_SECOND, agent_type_DEBOARDING_B), agentMODE) = 1;
89
  agent(agent_part_sum(class_SECOND, agent_type_DEBOARDING_B)+1 :
      agent_part_sum(class_SECOND, agent_type_CHANGING_A_B), agentMODE) = 4;
90 agent(agent_part_sum(class_SECOND, agent_type_CHANGING_A_B)+1 :
      agent_part_sum(class_SECOND, agent_type_CHANGING_B_A), agentMODE) = 2;
91
92
93 % Specify initial state (moving, deboarding)
94 agent(1
      agent_part_sum(class_FIRST, agent_type_BOARDING_A), agentSTATE) =
      agentSTATEmoving;
95 agent(agent_part_sum(class_FIRST, agent_type_BOARDING_A)+1
      agent_part_sum(class_FIRST, agent_type_BOARDING_B), agentSTATE) =
      agentSTATEmoving;
96 agent(agent_part_sum(class_FIRST, agent_type_BOARDING_B)+1
      agent_part_sum(class_FIRST, agent_type_DEBOARDING_A), agentSTATE) =
      agentSTATEdeboarding;
97 agent(agent_part_sum(class_FIRST, agent_type_DEBOARDING_A)+1 :
      agent_part_sum(class_FIRST, agent_type_DEBOARDING_B), agentSTATE) =
      agentSTATEdeboarding;
98 agent(agent_part_sum(class_FIRST, agent_type_DEBOARDING_B)+1 :
      agent_part_sum(class_FIRST, agent_type_CHANGING_A_B), agentSTATE) =
      agentSTATEdeboarding;
99 agent(agent_part_sum(class_FIRST, agent_type_CHANGING_A_B)+1 :
      agent_part_sum(class_FIRST, agent_type_CHANGING_B_A), agentSTATE) =
```

```
65
```

```
agentSTATEdeboarding;
100
   agent(agent_part_sum(class_FIRST, agent_type_CHANGING_B_A)+1 :
101
       agent_part_sum(class_SECOND, agent_type_BOARDING_A), agentSTATE) =
       agentSTATEmoving;
102 agent(agent_part_sum(class_SECOND, agent_type_BOARDING_A)+1
       agent_part_sum(class_SECOND, agent_type_BOARDING_B), agentSTATE) =
       agentSTATEmoving;
103 agent(agent_part_sum(class_SECOND, agent_type_BOARDING_B)+1
                                                                   •
       agent_part_sum(class_SECOND, agent_type_DEBOARDING_A), agentSTATE) =
       agentSTATEdeboarding;
104
   agent(agent_part_sum(class_SECOND, agent_type_DEBOARDING_A)+1 :
       agent_part_sum(class_SECOND, agent_type_DEBOARDING_B), agentSTATE) =
       agentSTATEdeboarding;
105 agent(agent_part_sum(class_SECOND, agent_type_DEBOARDING_B)+1 :
       agent_part_sum(class_SECOND, agent_type_CHANGING_A_B), agentSTATE) =
       agentSTATEdeboarding;
   agent(agent_part_sum(class_SECOND, agent_type_CHANGING_A_B)+1 :
106
       agent_part_sum(class_SECOND, agent_type_CHANGING_B_A), agentSTATE) =
       agentSTATEdeboarding;
107
108
109 %set leaving doors
110 agent(:, agentLDOOR) = zeros(agentcount, 1);
111
112
   agent (1
       agent_part_sum(class_FIRST, agent_type_BOARDING_A), agentLDOOR)
        = 0;
113
   agent(agent_part_sum(class_FIRST, agent_type_BOARDING_A)+1
114
       agent_part_sum(class_FIRST, agent_type_BOARDING_B), agentLDOOR)
115
        = 0:
116
   agent(agent_part_sum(class_FIRST, agent_type_BOARDING_B)+1
       agent_part_sum(class_FIRST, agent_type_DEBOARDING_A), agentLDOOR) ...
        = round(linspace(11,14,agent_part_count(class_FIRST,
117
           agent_type_DEBOARDING_A)));
   agent(agent_part_sum(class_FIRST, agent_type_DEBOARDING_A)+1
118
       agent_part_sum(class_FIRST, agent_type_DEBOARDING_B), agentLDOOR) ...
119
        = round(linspace(22,25,agent_part_count(class_FIRST,
           agent_type_DEBOARDING_B)));
   agent(agent_part_sum(class_FIRST, agent_type_DEBOARDING_B)+1 :
120
       agent_part_sum(class_FIRST, agent_type_CHANGING_A_B), agentLDOOR) ...
        = round(linspace(11,14,agent_part_count(class_FIRST,
121
           agent_type_CHANGING_A_B)));
   agent(agent_part_sum(class_FIRST, agent_type_CHANGING_A_B)+1 :
122
       agent_part_sum(class_FIRST, agent_type_CHANGING_B_A), agentLDOOR) ...
        = round(linspace(22,25,agent_part_count(class_FIRST,
123
           agent_type_CHANGING_B_A)));
124
   agent(agent_part_sum(class_FIRST, agent_type_CHANGING_B_A)+1 :
125
       agent_part_sum(class_SECOND, agent_type_BOARDING_A), agentLDOOR)
```

```
126
        = 0;
127 agent(agent_part_sum(class_SECOND, agent_type_BOARDING_A)+1
                                                                   •
       agent_part_sum(class_SECOND, agent_type_BOARDING_B), agentLDOOR)
       = 0;
128
   agent(agent_part_sum(class_SECOND, agent_type_BOARDING_B)+1
129
       agent_part_sum(class_SECOND, agent_type_DEBOARDING_A), agentLDOOR) ...
        = round(linspace(4,10,agent_part_count(class_SECOND,
130
           agent_type_DEBOARDING_A)));
131 agent(agent_part_sum(class_SECOND, agent_type_DEBOARDING_A)+1 :
       agent_part_sum(class_SECOND, agent_type_DEBOARDING_B), agentLDOOR) ...
132
        = round(linspace(15,21,agent_part_count(class_SECOND,
           agent_type_DEBOARDING_B)));
133
   agent(agent_part_sum(class_SECOND, agent_type_DEBOARDING_B)+1 :
       agent_part_sum(class_SECOND, agent_type_CHANGING_A_B), agentLDOOR) ...
        = round(linspace(4,10,agent_part_count(class_SECOND,
134
           agent_type_CHANGING_A_B)));
135 agent(agent_part_sum(class_SECOND, agent_type_CHANGING_A_B)+1 :
       agent_part_sum(class_SECOND, agent_type_CHANGING_B_A), agentLDOOR) ...
136
        = round(linspace(15,21,agent_part_count(class_SECOND,
           agent_type_CHANGING_B_A)));
137
138 %choice for entering door based on agentMODE
139 agent(:, agentCDOOR) = ones(agentcount,1);
140
141 agent(:, agentMAXV) = VELOCITY*ones(agentcount, 1) + VELOCITY_VAR * rand(
       agentcount, 1);
142 agent(:, agentPATIENT) = 0.9*ones(agentcount, 1);
143 agent(:, agentLAZY) = LAZINESS*ones(agentcount, 1);
144 agent(:, agentDMODE) = DOOR_DECISION_MODE * ones (agentcount, 1);
145 agentDECstepfrequency = DECISION_STEPFREQ; % inicates the step-based freq.
       agent decides for best door
146 agent(:, agentDECTIMES) = DECISION_LIMIT*ones(agentcount, 1);
147 agent(:, agentGROUP) = agentGROUPnone*ones(agentcount, 1);
148
149
150
151
152 % -----
153 % DOORS
154 % -----
155 doorcount = 25; % 2 trains, each 3x second class, 1x Bistro (1 door), 2x
       first class, 3 exits
156
157 % Array for doors
158 door = zeros(doorcount, doorCOLCOUNT);
159 doorrange = 0.5;
160 doorstrength = FORCES_COEFF(FC_doorAttraction);
161 doors_opening_time = DOORS_DELAY;
162
163
```

```
164 % Exits
165 \, \text{door}(1, \, \text{doorXPOS}) = 49.9;
166 door(2, doorXPOS) = 155.1;
167 door(3, doorXPOS) = 195;
168
169 door(1:3, doorYPOS) = 20;
170 agent_mode_enter_subway = 1;
171 door(1:3, doorMODE) = agent_mode_enter_subway;
172 door(1:3, doorSTATE) = 0;
173 door(1:3, doorMEANFREQ) = 10; %more people than on the train
174 \text{ door}(1:3, \text{ doorVARFREQ}) = 0.1;
175 door(1:3, doorACTIVITY) = doorACTIVE;
176
177 % First Train
178 \text{ door}(4, \text{ door}(4)) = 15+0*25+1.5;
179 door(5, doorXPOS) = 15+0*25+23.5;
                                            % Second class
180 door(6, doorXPOS) = 15+1*25+1.5;
181 door(7, doorXPOS) = 15+1*25+23.5;
182 door(8, doorXPOS) = 15+2*25+1.5;
183 door(9, doorXPOS) = 15+2*25+23.5;
184 \text{ door}(10, \text{ door}(10)) = 15+3+25+1.5;
                                              % Bistro
185 door(11, doorXPOS) = 15+4*25+1.5;
                                              % First class
186 door(12, doorXPOS) = 15+4*25+23.5;
187 door(13, doorXPOS) = 15+5*25+1.5;
188 door(14, doorXPOS) = 15+5*25+23.5;
189
190 door(4:14, doorYPOS) = 24.9;
191 door(4:10, doorMODE) = 2;
192 door(11:14, doorMODE) = 3;
193 door(4:14, doorSTATE) = doors_opening_time; % wait for some seconds until
        people can de/board
194 door(4:14, doorMEANFREQ) = 0.7;
195 door(4:14, doorVARFREQ) = 0.1;
196 door(4:14, doorACTIVITY) = doorACTIVE;
197
198 % Second Train
199 door(15, doorXPOS) = 15+0*25+1.5;
200 \text{ door}(16, \text{ door}(16)) = 15+0*25+23.5;
                                             % Second class
201 \text{ door}(17, \text{ door}(17)) = 15+1+25+1.5;
202 \text{ door}(18, \text{ door}(18)) = 15+1+25+23.5;
203 \text{ door}(19, \text{ door}(19)) = 15+2+25+1.5;
204 \text{ door}(20, \text{ door}(20)) = 15+2+25+23.5;
205 door(21, doorXPOS) = 15+3*25+1.5;
                                              % Bistro
                                            % First class
206 \text{ door}(22, \text{ door}(22)) = 15+4 \times 25+1.5;
207 \text{ door}(23, \text{ door}(23)) = 15+4 \times 25+23.5;
208 \text{ door}(24, \text{ door}(25)) = 15+5*25+1.5;
209 \text{ door}(25, \text{ door} XPOS) = 15+5*25+23.5;
210
211 door(15:25, doorYPOS) = 15.1;
212 \text{ door}(15:21, \text{ doorMODE}) = 4;
```

```
213 \text{ door}(22:25, \text{ doorMODE}) = 5;
214 door(15:25, doorSTATE) = doors_opening_time; % wait for some seconds until
       people can de/board
_{215} door(15:25, doorMEANFREQ) = 0.7;
216 door(15:25, doorVARFREQ) = 0.1;
217 door(15:25, doorACTIVITY) = doorACTIVE;
218
219 doorMODEsum = max(door(:,doorMODE));
220
221 % Sum up number of leaving agents
222 for idoor = 1:doorcount
223
       door(idoor, doorAGENT) = -sum(agent(:, agentLDOOR)==idoor);
224 end
225
226
227 % -----
228 % OBSTACLES
229 % ------
230
231 traincount = 2;
232
233 obstaclecount = 9; %2 trains, 2 triple subway entrances, 1 start area
234
235 % Array for obstacles
236 obstacle = zeros(obstaclecount, obstacleCOLCOUNT);
237
238 obstacle(:, obstacleSTART) = 0;
239 obstacle(:, obstacleEND) = tmax;
240
241 obstacle(:, obstacleRANGE) = FORCES_COEFF(FC_obstacleRetraction)*ones(
       obstaclecount, 1);
242 obstacle(:, obstaclePASSABLE) = zeros(obstaclecount, 1);
243
244 % trains
245 obstacle(1, [obstacleXCENTER, obstacleYCENTER]) = [100, 27.5];
246 obstacle(1, [obstacleWIDTH, obstacleHEIGHT]) = [170,5];
247 obstacle(2, [obstacleXCENTER, obstacleYCENTER]) = [100, 12.5];
248 obstacle(2, [obstacleWIDTH, obstacleHEIGHT]) = [170,5];
249
250 % subway entrances
251 obstacle(3, [obstacleXCENTER, obstacleYCENTER]) = [55, 20];
252 obstacle(3, [obstacleWIDTH, obstacleHEIGHT]) = [9.5, 5];
253 obstacle(4, [obstacleXCENTER, obstacleYCENTER]) = [150, 20];
254 obstacle(4, [obstacleWIDTH, obstacleHEIGHT]) = [9.5, 5];
255
256 % further small obstacles
257 obstacle(5, [obstacleXCENTER, obstacleYCENTER]) = [80, 20];
258 obstacle(5, [obstacleWIDTH, obstacleHEIGHT]) = [3, 1.5];
259 obstacle(6, [obstacleXCENTER, obstacleYCENTER]) = [105, 20];
260 obstacle(6, [obstacleWIDTH, obstacleHEIGHT]) = [3, 1.5];
```

```
261 obstacle(7, [obstacleXCENTER, obstacleYCENTER]) = [130, 20];
262 obstacle(7, [obstacleWIDTH, obstacleHEIGHT]) = [3, 1.5];
263 obstacle(8, [obstacleXCENTER, obstacleYCENTER]) = [30, 20];
264 obstacle(8, [obstacleWIDTH, obstacleHEIGHT]) = [3, 1.5];
265
266
267 %start area
268 OSTARTAREA = 9;
269 obstacle(oSTARTAREA, [obstacleXCENTER, obstacleYCENTER]) = [90, 20];
270 obstacle(oSTARTAREA, [obstacleWIDTH, obstacleHEIGHT]) = WAITING_AREA(2,:);
271
272 obstacle(oSTARTAREA, obstacleSTART) = 0;
273 obstacle(oSTARTAREA, obstacleEND) = AREA_DELAY;
274
275
276 % ------
277 % TRAIN SEATS
278 % ------
279
280 trainseats = zeros(traincount, 6*10, 2);
281 % Restaurant Coach and first Class already half full
282 trainseats(:,31:60,1) = 4*ones(traincount,3*10,1);
283
284 % set people, that are already seated
285 trainseats(:,1:30,:) = round(AGENTS_SEATED / 90);
286 trainseats(:,31:60,2) = round(AGENTS_SEATED / 90);
287
288 % -----
289 % AGENT POSITIONS
290 % -----
291
292 % Set random position for boarding agents
293 % (debording agents are going to be reset on their startposition in
294 % "simulation.m")
295 dspace = 0.2; % min space between obstacle and agent
296
297 for iagent=1:agentcount
      % call of script that sets random position until outside of any
298
299
       % obstacle, that is not the starting area
       set_agent_outside_of_any_obstacle
300
301 end
302
303
304 % - - -
305 % Group
306 % - - -
307 % A group consists of a couple of boarding(!) agents. They are all heading
308 % to the same door, which only can be chosen by their group-master.
309
310 N_groups(class_FIRST) = round(GROUPING*PART_FC*AGENTS_OP*2/GROUP_SIZE);
```

```
311 N_groups(class_SECOND) = round(GROUPING*(1-PART_FC)*AGENTS_OP*2/GROUP_SIZE);
312 groupcount = N_groups(class_FIRST)+N_groups(class_SECOND);
313
314 Size_group = zeros(groupcount, 1);
315 Size_group(1:N_groups(class_FIRST)) = GROUP_SIZE;
316 Size_group(N_groups(class_FIRST)+1:groupcount) = GROUP_SIZE;
317
318 % First-Class Groups
319 sagent = 1;
320 sgroup = 1;
321 for igroup = sgroup:N_groups(class_FIRST)
322
       init_group
323 end
324 sgroup = sgroup + N_groups(class_FIRST);
325
326 % Second-Class Groups
327 sagent = agent_part_sum(class_FIRST, agent_type_CHANGING_B_A)+1;
328 for igroup = sgroup:(sgroup-1) + N_groups(class_SECOND)
329
        init_group
330 end
331
332
333 % -----
334
335
336 % remaining Time between agent and door
337 remainingdistance = zeros(agentcount, doorcount);
338 remainingwalktime = zeros(agentcount, doorcount);
339 remainingqueuetime = zeros(agentcount, doorcount);
340 placeinqueue = ones(agentcount, doorcount);
341
342
343
344 % load statistic variables
345 init_statistics
346
347 %start simulation
348 simulation
```

Listing 9: init_group.m

```
1 % loop through all members of a group and init agent-values for XPOS,
2 % YPOS,GROUP and DECTIMES
3
4 for iagent = sagent:(sagent-1)+Size_group(igroup)
5 % all agents of the group get grouped around a circle, that gets scaled
6 % according to the number of people inside the group (10 people -> in
7 % circle with diameter of 2 meters
8 radius = Size_group(igroup)/10;
```

```
% only master of group can decide
9
       if iagent ≠ sagent
10
           agent(iagent, agentDECTIMES) = agentDECTIMESnone;
11
       else
12
13
           dspace = 1.5 * radius;
           set_agent_outside_of_any_obstacle;
14
15
           group_CENTER = agent(sagent, [agentXPOS, agentYPOS]);
       end
16
       agent(iagent, agentGROUP) = igroup;
17
       phi = 2*pi()*iagent/Size_group(igroup); % Angle to set group in a cyrcle
18
19
       agent(iagent, [agentXPOS, agentYPOS]) = group_CENTER + radius*[cos(phi),
            sin(phi)];
20 end
21 sagent = sagent + Size_group(igroup);
```

Listing 10: init_statistics.m

```
1 % initialistation of saved statistic data
2 %-----
3 % Arrays for saving data
4 %-----
5
6 % # agents heading to spec. door
7 stat_approaching_to_door = zeros(stepcount, doorcount);
8 % # agents on platform
9 stat_moving_agents = zeros(stepcount, 1);
10 % mean distance of all moving agents to their door
11 stat_distance_to_go = zeros(stepcount, 1);
12 % # agents boarded on spec. door
13 stat_boarded_per_door = zeros(stepcount, doorcount);
14 % start end end time of moving
15 stat_moving_time = zeros(agentcount, 2);
16
      stat_movSTART = 1;
17
      stat_movEND = 2;
18 % # agents waiting in queue
19 stat_waiting_agents = zeros(stepcount, 2);
20 % sum. waiting time
21 stat_sum_waiting = zeros(agentcount, 1);
22 % walking distance per agent
23 stat_sum_distance = zeros(agentcount, 1);
24 % min. distance between leavingdoor and chosendoor
25 stat_min_distance = zeros(agentcount, 1);
26 % startpostion (is needed to calculate stat_min_distance
27 stat_start_position = zeros(agentcount, 2); % set in simulation.m
28 % sum of redecicions
29 stat_sum_decision = zeros(agentcount, 1);
```
Listing 11: set_agent_outside_of_any_obstacle.m

```
% move agent randomly inside the StartArea until a minimal distance of ...
   % 'dspace' to every other obstacle and the border of the startarea is
2
       garanteed
3
   % Consider also, that a person will not start from a position too far from
4
   % its desteny, i.e. that a first-class passenger is rather startig from a
5
   % point not so far from the nearest first-class entrance.
6
   %agent(iagent, agentXPOS) = obstacle(oSTARTAREA, obstacleXCENTER) - obstacle
8
       (oSTARTAREA, obstacleWIDTH)/2 + ...
        (obstacle(oSTARTAREA, obstacleWIDTH)-2*dspace)*rand(1,1)+dspace;
9 %
10 %agent(iagent, agentYPOS) = obstacle(oSTARTAREA, obstacleYCENTER) - obstacle
       (oSTARTAREA, obstacleHEIGHT)/2 + ...
11 %
        (obstacle(oSTARTAREA, obstacleHEIGHT)-2*dspace)*rand(1,1)+dspace;
12
13 is_agentPOSok = 0;
14 while is_agentPOSok == 0
       agent(iagent, agentXPOS) = obstacle(oSTARTAREA, obstacleXCENTER) -
15
           obstacle(oSTARTAREA, obstacleWIDTH)/2 + ...
           (obstacle(oSTARTAREA, obstacleWIDTH)-2*dspace)*rand(1,1)+dspace;
16
       agent(iagent, agentYPOS) = obstacle(oSTARTAREA, obstacleYCENTER) -
17
           obstacle(oSTARTAREA, obstacleHEIGHT)/2 + ...
           (obstacle(oSTARTAREA, obstacleHEIGHT)-2*dspace)*rand(1,1)+dspace;
18
19
       is_agentPOSok = 1; % assuming pos is ok
20
21
22
       % Check if agent ist not too far away from its nearest possible door
       for idoor = 1:doorcount
23
       remainingdistance(iagent, idoor) = norm(agent(iagent, [agentXPOS,
24
           agentYPOS]) - door(idoor, [doorXPOS, doorYPOS]));
     end
25
       min_remainingdistance_MODE = min(remainingdistance(iagent, door(:,
26
           doorMODE) == agent(iagent, agentMODE)));
       if min_remainingdistance_MODE > border(3)/4
                                                        % more than quarter
27
           scene?
           is_agentPOSok = 0;
28
29
       end
30
31
32
       % now check for obstacles
       for iobstacle = 1:obstaclecount
33
34
           if iobstacle ≠ oSTARTAREA
                                        % dont check start area
               if agent(iagent, agentXPOS) > (obstacle(iobstacle,
35
                   obstacleXCENTER) - obstacle(iobstacle, obstacleWIDTH)/2 -
                   dspace)
                   if agent(iagent, agentXPOS) < (obstacle(iobstacle,</pre>
36
                       obstacleXCENTER) + obstacle(iobstacle, obstacleWIDTH)/2
                       + dspace)
```

37	<pre>if agent(iagent, agentYPOS) > (obstacle(iobstacle,</pre>
	obstacleYCENTER) - Obstacle(lobstacle,
	obstacleHEIGHT)/2 - dspace)
38	if agent(iagent, agentYPOS) < (obstacle(iobstacle,
	obstacleYCENTER) + obstacle(iobstacle,
	obstacleHEIGHT)/2 + dspace)
39	is_agentPOSok = 0; % Position was NOT ok
40	<pre>%agent(iagent, agentXPOS) = obstacle(oSTARTAREA,</pre>
	obstacleXCENTER) - obstacle(oSTARTAREA,
	obstacleWIDTH)/2 +
41	<pre>% (obstacle(oSTARTAREA, obstacleWIDTH)-2*</pre>
	dspace) *rand(1,1)+dspace;
42	<pre>%agent(iagent, agentYPOS) = obstacle(oSTARTAREA,</pre>
	obstacleYCENTER) - obstacle(oSTARTAREA,
	obstacleHEIGHT)/2 +
43	<pre>% (obstacle(oSTARTAREA, obstacleHEIGHT)-2*</pre>
	dspace) *rand(1,1)+dspace;
44	end
45	end
46	end
47	end
48	end
40	end
50	end
- 30	ena

8.2.3 Simulation

Listing 12: simulation.m

```
1 % simulation
\mathbf{2}
3 % set deboarding agents on the doorpoint
4 for iagent = 1:agentcount
       if agent(iagent, agentSTATE) == agentSTATEdeboarding
\mathbf{5}
6
           agent(iagent, [agentXPOS, agentYPOS]) = door(agent(iagent,
               agentLDOOR), [doorXPOS, doorYPOS]);
7
       end
       stat_start_position(iagent, :) = agent(iagent, [agentXPOS, agentYPOS]);
8
9 end
10
11 %timestep iteration
12 for step = 1:stepcount
       t = step*dt;
13
14
       if t \leq DOORS_DELAY
15
           train_entrance;
16
17
       end
```

```
18
       % random order for agents
       order = randperm(agentcount);
19
       doordecision_frequency;
20
       % decrement door state ("blocking" time)
21
22
       door(:,doorSTATE) = door(:,doorSTATE) - dt*ones(doorcount,1);
       calculate_distances;
23
24
       for i = 1:agentcount
25
           iagent = order(i);
26
           % people update in random order (board, deboard, status change)
27
^{28}
           agent_update
29
           if agent(iagent, agentSTATE) == agentSTATEmoving
30
                door_decision;
                calculate_forces;
31
32
           end
33
       end
       % move agents simultaneously
34
35
       move_agents
36
37
       % draw
38
       paint
       video_capture
39
       pause(0.02)
40
41
       save_data
42
       data_export
43
44 end
```

Listing 13: train_entrance.m

```
1 % let train drive into its final position before the doors open
2
3 if step == 1
      % set initial position of train and doors
4
      obstacle(1:traincount, obstacleXCENTER) = obstacle(1:traincount,
5
          obstacleXCENTER) - DOORS_DELAY * trainVELOCITY;
      door(door(:,doorMODE) \u2274 agent_mode_enter_subway, doorXPOS) = door(door
6
          (:,doorMODE) \neq agent_mode_enter_subway, doorXPOS) - DOORS_DELAY *
          trainVELOCITY;
7 end
8 obstacle(1:traincount, obstacleXCENTER) = obstacle(1:traincount,
      obstacleXCENTER) + dt * trainVELOCITY;
9 door(door(:,doorMODE) \u2274 agent_mode_enter_subway, doorXPOS) = door(door(:,
      doorMODE) \u2224 agent_mode_enter_subway, doorXPOS) + dt * trainVELOCITY;
```

Listing 14: doordecision_frequency.m

```
1 % The frequency of redecision is checked
2 if agentDECstepfrequency \leq 1
       if mod(step, 1/agentDECstepfrequency) < mod((step-1), 1/</pre>
3
           agentDECstepfrequency);
            Ndec = 0;
4
       else
\mathbf{5}
6
            Ndec = 1;
       end
7
8 else
       if mod(step, agentDECstepfrequency) < mod((step-1),</pre>
9
           agentDECstepfrequency);
10
            Ndec = floor(agentDECstepfrequency) + 0;
11
       else
            Ndec = floor(agentDECstepfrequency) + 1;
12
13
       end
14 end
```

Listing 15: calculate_distances.m

Listing 16: agent_update.m

```
1 % agent's state update (board, deboard: status change)
2
3 % check if agent can debord
4 if agent(iagent, agentSTATE) == agentSTATEdeboarding;
       ldoor = agent(iagent, agentLDOOR);
\mathbf{5}
       if door(ldoor, doorSTATE) \leq 0;
6
           % let agent debord
7
           agent(iagent, agentSTATE) = agentSTATEmoving;
8
           stat_moving_time(iagent, stat_movSTART) = t;
9
           % block door for a moment
10
           door(ldoor, doorSTATE) = 1/(door(ldoor, doorMEANFREQ)+...
11
               randn(1) * door(ldoor, doorVARFREQ));
12
           % decrease counter of people left in door
13
           door(ldoor, doorAGENT) = door(ldoor, doorAGENT) - doorAGENTdebord;
14
       end
15
16 end
17
```

```
18 % check if agent can bord
  if agent(iagent, agentSTATE) == agentSTATEmoving;
19
       cdoor = agent(iagent, agentCDOOR);
20
       if remainingdistance(iagent, cdoor) < doorrange & ...
21
               door(cdoor, doorSTATE) < 0
22
           % check whether there is a free seat in this coach
23
24
           agent_seat_search;
           if free_seat_found == 1
25
               % let agent board
26
               agent(iagent, agentSTATE) = agentSTATEboarded;
27
               stat_moving_time(iagent, stat_movEND) = t;
28
29
               stat_sum_distance(iagent) = stat_sum_distance(iagent) + norm(
                   agent(iagent, [agentXPOS, agentYPOS]) - door(agent(iagent,
                   agentCDOOR), [doorXPOS, doorYPOS]));
               % block door for a moment
30
               door(cdoor, doorSTATE) = 1/(door(cdoor, doorMEANFREQ)+...
31
                    randn(1) *door(cdoor, doorVARFREQ));
32
33
               % increase counter of people borded
34
               door(cdoor, doorAGENT) = door(cdoor, doorAGENT) + doorAGENTbord;
35
           else
36
               % lock the door
               door(cdoor, doorACTIVITY) = doorINACTIVE;
37
               % give the agent a chance to possibly redecide for a new door
38
39
               if agent(iagent, agentDECTIMES) == agentDECTIMESnone
40
                   agent(iagent, agentDECTIMES) = 1;
41
               end
           end
42
       end
43
44 end
```

Listing 17: agent_seat_search.m

```
1 % tries to find a free seat for iagent starting from its current chosen
2 % door
3
4 % important note: this file is specifically designed for the two train
5 % station layouts and needs to be updated if any door-configuration gets
6 % changed
7 free_seat_found = 0;
  % making szenario-specific-separations
8
  if simulation_mode == simulationMODEonetrain
9
10
       if cdoor \leq 2
           free_seat_found = 1;
11
12
           return;
       end
13
14
       % each door of second class coach and bistro wagon
15
       ctrain = 1;
16
       if (cdoor \leq 9)
           ccoach = (cdoor - mod(cdoor+1, 2) - 1) / 2;
17
```

```
18
       % first class coaches
       else
19
           ccoach = (cdoor - mod(cdoor,2)) / 2;
20
       end
21
22 end
23
24 if simulation_mode == simulationMODEtwotrains
25
       if cdoor \leq 3
           free_seat_found = 1;
26
           return;
27
^{28}
       end
29
       % each door of second class coach and bistro wagon
30
       if (cdoor \leq 10)
           ctrain = 1;
31
           ccoach = (cdoor - mod(cdoor, 2) - 2) / 2;
32
33
       % first class coaches
       elseif (cdoor \leq 14)
34
35
           ctrain = 1;
36
            ccoach = (cdoor - mod(cdoor+1, 2) -1) / 2;
37
       elseif (cdoor \leq 21)
38
           ctrain = 2;
39
           ccoach = (cdoor - mod(cdoor+1, 2) - 1) / 2 - 6;
40
41
       else
42
           ctrain = 2;
43
           ccoach = (cdoor - mod(cdoor, 2)) / 2 - 6;
       end
44
45
46 end
47 coach_seat_search;
```

Listing 18: coach_seat_search.m

```
start_compartment = 10*(ccoach-1)+1;
2 end_compartment = 10*ccoach;
3
4 for icompartment = start_compartment : end_compartment
       for iside = 1 : 2
\mathbf{5}
            if trainseats(ctrain, icompartment, iside) < 4
6
              trainseats(ctrain, icompartment, iside) = trainseats(ctrain,
\overline{7}
                  icompartment, iside) + 1;
              free_seat_found = 1;
8
              return;
9
            end
10
       end
11
12
13 end
```

Listing 19: door_decision.m

1	% choose best door for the current moving agents
2	
3	for idec = 1:Ndec
4	<pre>if agent(lagent, agentSIALE) == agentSIALEmoving </pre>
5	if agent(iagent, agentDECTIMES) ≠ agentDECTIMESnone
6	<pre>placeinqueue(lagent, :) = ones(l, doorcount);</pre>
7	for kagent = 1:agentcount
8	<pre>if agent(kagent, agentSTATE) == agentSTATEmoving</pre>
9	<pre>kdoor = agent(kagent,agentCDOOR);</pre>
10	<pre>if(remainingdistance(kagent,kdoor) < remainingdistance(</pre>
	lagent, kdoor))
11	% agents in the same group are not considered
12	if ((agent(kagent, agentGROUP) ≠ agent(iagent,
	agentGROUP)) (agent(iagent, agentGROUP) ==
	agentGROUPnone))
13	placeinqueue(iagent, kdoor) = placeinqueue(
	iagent, kdoor) + 1;
14	end
15	end
16	end
17	end
18	
19	% calculate expected remaining time
20	<pre>ior idoor = 1:doorcount</pre>
21	<pre>11 ((agent(lagent, agentMODE) == door(ldoor, doorMODE)) &&</pre>
	(agent(lagent, agentLDUOR) \neq 1000r) & (d00r(1000r,
	doorACIIVIIY) == doorACIIVE))
22	remainingwaiktime(lagent, ldoor) = remainingdistance(
	Tagent, Idoor) / agent(Tagent, agentMAXV);
23	remainingqueuetime(lagent, ldoor) = placeinqueue(lagent
	, Idoor//door(Idoor, doorMEANFREQ);
24	else
25	remainingwaiktime(lagent, ldoor) = 9999; %%%%%%% NOI
0.0	PROPER
20	NETTUED
07	NEIINER
27	end
28	ena
29	& profer current decision with the patient factor
30	<pre>% pieler current decision with the patient factor remainingualktime (ingent ingent ingent ingent (DOOR)) =</pre>
31	remainingwalktime(iagent, agent(iagent, agent(DOOR)) + (
	agent (iagent agent PATIENT)).
20	ayenc(layenc, ayencrAllENI)), remainingqueuetime(isgent sgent(isgent sgent(DOOR)) = (
32	remainingqueuetime(iagent, agent(iagent, agentCDOOR)) = (
	<pre>remainingqueuecime(ragenc, agenc(ragenc, agenccDOOR))) * (agent (iagent agentPATIENT)).</pre>
3.2	ayenc(tayenc, ayencrAllEN1)),
34	& Choose appropriate door (depending choosing-mode)
954 95	witch agent (iggent agent DMODE)
55	Switcon agenc(iagenc, agencbrobe)

36	case agentDMODEsum_lazy
37	[remainingtime(iagent), newdoor] = min(
	<pre>remainingwalktime(iagent, :)*agent(iagent,</pre>
	<pre>agentLAZY) + remainingqueuetime(iagent, :)*(1-agent</pre>
	<pre>(iagent, agentLAZY)));</pre>
38	case agentDMODEsum
39	[remainingtime(iagent), newdoor] = min(
	remainingwalktime(iagent, :) + remainingqueuetime(
	<pre>iagent, :));</pre>
40	<pre>case agentDMODEwalk</pre>
41	[remainingtime(iagent), newdoor] = min(
	<pre>remainingwalktime(iagent, :));</pre>
42	case agentDMODEqueue
43	[remainingtime(iagent), newdoor] = min(
	<pre>remainingqueuetime(iagent, :));</pre>
44	case agentDMODEwait
45	[remainingtime(iagent), newdoor] = min(abs(
	remainingqueuetime(iagent, :) - remainingwalktime(
	iagent, :)));
46	case agentDMODErandom
47	<pre>[remainingtime(iagent), newdoor] = max(rand(doorcount</pre>
	,1) .* (remainingwalktime(iagent, :)' < 9000*ones(
	<pre>doorcount,1)));</pre>
48	end
49	
49 50	
49 50 51	if agent(iagent, agentCDOOR) \neq newdoor
49 50 51 52	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left</pre>
49 50 51 52 53	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite;</pre>
49 50 51 52 53 54	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent,</pre>
49 50 51 52 53 54	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent,</pre>
 49 50 51 52 53 54 55 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) - 1; end</pre>
 49 50 51 52 53 54 55 56 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) - 1; end % All members of a group to the same door </pre>
 49 50 51 52 53 54 55 56 57 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) - 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); </pre>
 49 50 51 52 53 54 55 56 57 58 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) - 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) </pre>
 49 50 51 52 53 54 55 56 57 58 59 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) - 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = rendered</pre>
49 50 51 52 53 54 55 56 57 58 59	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) - 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) =</pre>
 49 50 51 52 53 54 55 56 57 58 59 60 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = acent(CROUP) ==igroup) =</pre>
 49 50 51 52 53 54 55 56 57 58 59 60 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1.</pre>
 49 50 51 52 53 54 55 56 57 58 59 60 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1; elec</pre>
 49 50 51 52 53 54 55 56 57 58 59 60 61 62 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) - 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1; else agent(iagent_agentCDOOR) = newdoor;</pre>
 49 50 51 52 53 54 55 56 57 58 59 60 61 62 62 62 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) - 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1; else agent(iagent, agentCDOOR) = newdoor; stat_sum_decision(iagent) = stat_sum_decision(iagent) =</pre>
 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1; else agent(iagent, agentCDOOR) = newdoor; stat_sum_decision(iagent) = stat_sum_decision(iagent) + 1.</pre>
 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1; else agent(iagent, agentCDOOR) = newdoor; stat_sum_decision(iagent) = stat_sum_decision(iagent) + 1; end</pre>
 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1; else agent(iagent, agentCDOOR) = newdoor; stat_sum_decision(iagent) = stat_sum_decision(iagent) + 1; end end end end</pre>
 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) - 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1; else agent(iagent, agentCDOOR) = newdoor; stat_sum_decision(iagent) = stat_sum_decision(iagent) + 1; end end end end end end</pre>
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = agent(iagent, agentgent) = 1; end % All members of a group to the same door igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1; else agent(iagent, agentCDOOR) = newdoor; stat_sum_decision(iagent) = stat_sum_decision(iagent) + 1; end end end end end end </pre>
 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 	<pre>if agent(iagent, agentCDOOR) ≠ newdoor % decrease number of possible redicisiontimes left if agent(iagent, agentDECTIMES) ≠ agentDECTIMESinfinite; agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = agent(iagent, agentDECTIMES) = agent(iagent, agentGequent); if (igroup = agent(iagent, agentGROUP); if (igroup ≠ agentGROUPnone) agent(agent(:, agentGROUP)==igroup, agentCDOOR) = newdoor; stat_sum_decision(agent(:, agentGROUP)==igroup) = stat_sum_decision(agent(:, agentGROUP)==igroup) + 1; else agent(iagent, agentCDOOR) = newdoor; stat_sum_decision(iagent) = stat_sum_decision(iagent) + 1; end end end end end end end end</pre>

Listing 20: calculate_forces.m

```
1 % calculate the resulting force acting on "iagent"
2
3 % temporary sum of forces
4 agentforce = [0,0];
5 doorforce = [0,0];
6 obstacleforce = [0,0];
7 % people attraction and retraction
  for kagent = 1:agentcount
8
       if (kagent ≠ iagent & agent(kagent, agentSTATE) == agentSTATEmoving)
9
           vec_agentdist = agent(kagent, [agentXPOS, agentYPOS]) - ...
10
               agent(iagent, [agentXPOS, agentYPOS]);
11
           norm_agentdist = norm(vec_agentdist);
12
13
14
           % group people keep more together
           if (agent(iagent, agentGROUP) == agent(kagent, agentGROUP) & ...
15
                   agent(iagent, agentGROUP) > 0)
16
               strength = FORCES_COEFF(FC_agentAttractionGroup);
17
           else
18
               strength = FORCES_COEFF(FC_agentAttraction);
19
           end
20
21
           agentforce = agentforce - strength/(norm_agentdist)^3 ...
22
               * vec_agentdist/norm_agentdist;
23
           % agent attraction
24
           agentforce = agentforce + (strength/agentspace)/(norm_agentdist^2)
25
               . . .
26
               * vec_agentdist/norm_agentdist;
27
       end
28 end
29
30 % Door attraction and retraction
31 vec_doordist = door(agent(iagent, agentCDOOR), [doorXPOS, doorYPOS]) - ...
       agent(iagent, [agentXPOS, agentYPOS]);
32
33 norm_doordist = norm(vec_doordist);
34
35 % door attraction
36 doorforce = vec_doordist/norm_doordist;
37 % door retraction while occupied
38 if door(agent(iagent, agentCDOOR), doorSTATE) > 0
39
       doorforce = doorforce - doorrange/norm_doordist * ...
           vec_doordist/norm_doordist;
40
41 end
42 doorforce = doorstrength * doorforce;
43
44
45 % Obstacle retraction
46 for kobstacle = 1:obstaclecount
47 iforce = [0,0];
48
```

```
if obstacle(kobstacle,obstacleSTART) < t && obstacle(kobstacle,
49
           obstacleEND) > t
           agentx = agent(iagent, agentXPOS);
50
           agenty = agent(iagent, agentYPOS);
51
52
           obstaclex = obstacle(kobstacle, obstacleXCENTER);
           obstacley = obstacle(kobstacle, obstacleYCENTER);
53
           obstaclew = obstacle(kobstacle, obstacleWIDTH);
54
55
           obstacleh = obstacle(kobstacle, obstacleHEIGHT);
            %if inside obstacle
56
           if (abs(agentx - obstaclex) < obstaclew/2) && (abs(agenty -</pre>
57
                obstacley) < obstacleh/2)</pre>
58
59
                mindistance = max([obstaclew, obstacleh]);
                closestwall = 0;
60
61
                for idir = 1:4
                    % orthogonal distance to the closest wall
62
                    distance = abs(xdir(idir)) * abs(obstaclex+xdir(idir)*
63
                        obstaclew/2 - agentx) + abs(ydir(idir)) * abs(obstacley+
                        ydir(idir)*obstacleh/2 - agenty);
64
65
                    if mindistance > distance
                        mindistance = distance;
66
67
                         closestwall = idir;
68
                    end
69
                end
70
                iforce(1) = (-xdir(closestwall)*obstacle(kobstacle,obstacleRANGE
71
                    ))/mindistance;
72
                iforce(2) = (-ydir(closestwall)*obstacle(kobstacle,obstacleRANGE
                    ))/mindistance;
73
74
                obstacleforce = obstacleforce + iforce;
75
           %outside of the obstacle
76
           else
77
                x\Delta = 0; y\Delta = 0;
78
                if agentx > obstaclex + obstaclew/2
79
80
                    x\Delta = 1;
81
                end
                if agentx < obstaclex - obstaclew/2</pre>
82
                    x\Delta = -1;
83
84
                end
                if agenty > obstacley + obstacleh/2
85
86
                    y\Delta = 1;
                end
87
                if agenty < obstacley - obstacleh/2
88
                    y\Delta = -1;
89
                end
90
91
                edge = [0,0];
92
```

```
%nearest point is an edge
94
                 if (x\Delta \neq 0) && (y\Delta \neq 0)
95
                     edge(1) = obstaclex + x\Delta * obstaclew/2;
96
97
                     edge(2) = obstacley + y\Delta*obstacleh/2;
98
99
                 %neares point is a side
                 else
100
101
                     if x\Delta \neq 0
                          edge(1) = obstaclex + x\Delta * obstaclew/2;
102
103
                          edge(2) = agenty;
104
                     else
                          edge(1) = agentx;
105
106
                          edge(2) = obstacley + y\Delta*obstacleh/2;
107
                     end
108
                 end
109
110
                 %calculate distance and resulting force
111
                 vec_diff = agent(iagent, [agentXPOS, agentYPOS]) - edge;
112
                 iforce = vec_diff/norm(vec_diff) * obstacle(kobstacle,
                     obstacleRANGE) / norm (vec_diff);
                 obstacleforce = obstacleforce + iforce;
113
114
115
            end
116
        end
117 end
118
119 % Correction Force (helps to avoid obstacles)
120 if (abs(doorforce(1) + obstacleforce(1)) < 1000 && remainingdistance(iagent
        , agent(iagent, agentCDOOR)) > 5*doorrange) ...
        % "cross product" between obstacleforce and doorforce
121
122
        cross_OxD = obstacleforce(1) *doorforce(2) -obstacleforce(2) *doorforce(1);
        orth_O = obstacleforce *[0, -1; 1, 0];
123
        %orth_0 = [obstacleforce(2), -obstacleforce(1)];
124
        corrforce = sign(cross_OxD) * 99999999*orth_O; %very high
125
        % For Agents entering Train inverse way round
126
127
        if agent(iagent, agentMODE) \neq agent_mode_enter_subway;
128
            corrforce = -corrforce;
129
        end
130 else
        corrforce = [0, 0];
131
132 end
133
134 agent(iagent, [agentXFORCE, agentYFORCE]) = doorforce + agentforce +
        obstacleforce + corrforce;
```

Listing 21: move_agents.m

1 % move agents: a -> dv -> v -> ds -> s

93

```
3 for iagent = 1:agentcount
       dv = agent(iagent, [agentXFORCE, agentYFORCE])/agentmass;
4
5
       agent(iagent, [agentXVEL, agentYVEL]) = ...
6
           agent(iagent, [agentXVEL, agentYVEL]) + dv*dt;
\overline{7}
8
       if norm(agent(iagent, [agentXVEL, agentYVEL])) > agent(iagent, agentMAXV
9
           )
       agent(iagent, [agentXVEL, agentYVEL]) = agent(iagent, [agentXVEL,
10
           agentYVEL]) / norm(agent(iagent, [agentXVEL, agentYVEL])) * agent(
           iagent, agentMAXV);
11
       end
12
       if agent(iagent, agentSTATE) == agentSTATEmoving
13
           ds = (agent(iagent, [agentXVEL, agentYVEL])) * dt;
14
       else
15
16
           ds = [0, 0];
17
       end
18
19
       % sum distance per agent
       stat_sum_distance(iagent) = stat_sum_distance(iagent) + norm(ds);
20
21
22
       agent(iagent, [agentXPOS, agentYPOS]) = agent(iagent, [agentXPOS,
           agentYPOS]) + ds;
23 end
```

8.2.4 Statistical evaluation

2

Listing 22: save_data.m

```
1 % save/update statistical data
2
3 stat_moving_agents(step) = sum(agent(:,agentSTATE)==agentSTATEmoving);
4
5 for iagent = 1 : agentcount
       if agent(iagent, agentSTATE) == agentSTATEmoving
6
           % calculate average distance
7
8
           stat_distance_to_go(step) = stat_distance_to_go(step) + (
               remainingdistance(iagent, agent(iagent, agentCDOOR))/
               stat_moving_agents(step));
9
10
           % count waiting agents
11
12
           if (remainingdistance(iagent, agent(iagent, agentCDOOR)) < 3*
               doorrange)
13
               % Differentiate between boarder and deboarder
```

```
14
               if agent(iagent, agentMODE) == agent_mode_enter_subway
                    stat_waiting_agents(step,1) = stat_waiting_agents(step,1) +
15
                       1;
               else
16
                   stat_waiting_agents(step, 2) = stat_waiting_agents(step, 2) +
17
                       1;
               end
18
                   stat_sum_waiting(iagent) = stat_sum_waiting(iagent) + dt;
19
               else
20
           end
21
22
23
           % calculate min distance between startposition and heading door
24
           stat_min_distance(iagent) = norm(stat_start_position(iagent, :) -
               door(agent(iagent, agentCDOOR), [doorXPOS, doorYPOS]));
25
       end
26 end
27
  for kdoor = 1:doorcount
28
29
       for iagent = 1:agentcount
30
           if (agent(iagent, agentSTATE) == agentSTATEmoving) && (agent(iagent,
               agentCDOOR) == kdoor)
               stat_approaching_to_door(step,kdoor) = stat_approaching_to_door(
31
                   step,kdoor) + 1;
32
           end
33
       end
34 end
35
36 % if all agents boarded, save the time:
37 if (sum(agent(agent(:, agentMODE) ≠ agent_mode_enter_subway, agentSTATE) ≠
       agentSTATEboarded) == 0) ...
           && (final_boarding_time == 0)
38
39
       final_boarding_time = t;
40 end
41
42 stat_boarded_per_door(step,:) = door(:, doorAGENT);
```

8.2.5 Plotting

Listing 23: paint.m

```
1 % plot current situation
2
3 if plotting_mode == plotDEFAULT
4 if(mod(t,10) == 0)
5 t
6 end
7 else
```

```
figure(my_figure);
8
       if plotting_mode == plotMAPview
9
           clf
10
11
12
           img_scale_factor = 10;
13
14
           axis(img_scale_factor*[border(1), border(1)+border(3), border(2),
               border(2)+border(4)]);
           axis equal;
15
           hold on;
16
17
18
           rgb = imread('Train_EC.jpg');
           for itrain = 1:traincount
19
                xtrain = obstacle(itrain, obstacleXCENTER) - obstacle(itrain,
20
                    obstacleWIDTH) /2;
                ytrain = obstacle(itrain, obstacleYCENTER) - obstacle(itrain,
21
                   obstacleHEIGHT)/2;
22
23
                % shift dots outside of the train image
24
                if traincount == 2
25
                    if itrain == 1
                        y_offshift = 5;
26
                    else
27
28
                        y_{offshift} = -5;
29
                    end
30
                else
                    y_offshift = 5;
31
                end
32
33
                image(img_scale_factor*(xtrain), img_scale_factor*(ytrain), rgb);
34
35
36
                for a = 1:60
                    for b = 1:2
37
                        if (trainseats (itrain, a, b) \ge 4)
38
                            plot(img_scale_factor*( xtrain + 150/60 * (a-0.5)),
39
                                img_scale_factor*( y_offshift + ytrain - 2 + 3*b
                                ), 'sr');
40
                        elseif(trainseats(itrain, a, b) == 3)
41
                            plot(img_scale_factor*( xtrain + 150/60 * (a-0.5)),
                                img_scale_factor*( y_offshift + ytrain - 2 + 3*b
                                ), 'sm');
                        elseif(trainseats(itrain, a, b) == 2)
42
                            plot(img_scale_factor*( xtrain + 150/60 * (a-0.5)),
43
                                img_scale_factor*( y_offshift + ytrain - 2 + 3*b
                                ), 'sy');
                        elseif(trainseats(itrain, a, b) == 1)
44
                            plot(img_scale_factor*( xtrain + 150/60 * (a-0.5)),
45
                                 img_scale_factor*( y_offshift + ytrain - 2 + 3*b
                                ), 'sc');
                        else
46
```

47	plot(img_scale_factor*(xtrain + 150/60 * (a-0.5)),
	img_scale_factor*(y_offshift + ytrain - 2 + 3*b
), 'sg');
48	end
49	end
50	end
51	end
52	
53	<pre>% plot the moving agents (red color = no group, blue color = some</pre>
54	% group)
55	<pre>plot(img_scale_factor*agent(((agent(:,agentSTATE)==agentSTATEmoving)</pre>
57	<pre>& (agent(:,agentGROUP) == agentGROUPnone)),agentYPOS),</pre>
58	<pre>plot (img scale factor*agent(((agent(:,agentSTATE)==agentSTATEmoving)))</pre>
50	<pre>& (agent(:,agentGROUP) ≠agentGROUPnone)),agentXPOS), img scale factor+agent((/agent(: agentSTATE)==agentSTATEmoving))</pre>
	<pre>& (agent(:,agentGROUP)≠agentGROUPnone)),agentYPOS),</pre>
60	·m.·)
61	$plot(img_{gaalo} factor_door(door(\cdot, doorSTATE))) = 10+$
02	<pre>door(door(:,doorSTATE)>10*dt,doorYPOS), 'xr')</pre>
63	<pre>door(lmg_scale_factor*door(door(:,doorSTATE)<10*dt,doorXPOS), 10*</pre>
64	
65	<pre>ior iobstacle = (traincount+1):obstaclecount</pre>
66	<pre>rect(1,:) = [obstacle(iobstacle,obstacleXCENTER) - obstacle(</pre>
67	<pre>rect(2,:) = [obstacle(iobstacle,obstacleXCENTER) - obstacle(iobstacle,obstacleWIDTH)/2 , obstacle(iobstacle, obstacleYCENTER) + obstacle(iobstacle, obstacle/EICHT)/21;</pre>
69	$roct(3 \cdot) = [obstacle(iobstacle obstacle) + obstacle($
00	<pre>iobstacle,obstacleWIDTH)/2 , obstacle(iobstacle, obstacleYCENTER) + obstacle(iobstacle,obstacleHEIGHT)/2];</pre>
69	<pre>rect(4,:) = [obstacle(iobstacle,obstacleXCENTER) + obstacle(</pre>
	<pre>obstacley(ENTER) = obstacle(iobstacle, obstacleHEIGHT)/2]; wet/(5, i) [obstacle(iobstacle, obstacleHEIGHT)/2];</pre>
70	<pre>iobstacle,obstacle(iobstacle,obstaclextenter) - obstacle(iobstacle,obstacleWIDTH)/2 , obstacle(iobstacle, obstacleYCENTER) - obstacle(iobstacle,obstacleHEIGHT)/2];</pre>
71	if (obstacle(iobstacle,obstacleSTART) \leq t) && (obstacle(iobstacle,obstacleEND) \geq t)
72	<pre>plot(img_scale_factor*rect(:,1),img_scale_factor*rect(:,2),</pre>
73	else
74	<pre>plot(img_scale_factor*rect(:,1),img_scale_factor*rect(:,2),</pre>
75	end

```
76
            end
77
            text(img_scale_factor*(border(1)+1), img_scale_factor*(border(2)+1),
78
                num2str(t));
79
       end
80
81
       if plotting_mode == plotGRAPHview
82
            clf
83
            plot_saved_data
^{84}
85
       end
86 end
```

Listing 24: plot_saved_data.m

```
1 timevector = dt:dt:t;
2
3 subplot(3,3,1)
4 plot_saved_approaching;
\mathbf{5}
6 subplot(3,3,2)
7 plot_saved_moving;
8
9 subplot(3,3,3)
10 plot_saved_distance;
11
12 subplot(3,3,4)
13 plot_saved_deboarded;
14
15 subplot(3,3,5)
16 plot_saved_boarded;
17
18 subplot(3,3,6)
19 plot_saved_waiting;
20
21 subplot(3,3,7)
22 plot_saved_time_waited;
^{23}
24 subplot(3,3,8)
25 plot_saved_redecisions;
26
27 subplot(3,3,9)
28 plot_saved_distance_walked;
```

Listing 25: plot_saved_approaching.m

1 hold on

```
2 \quad if(1 \leq \text{doorMODEsum})
     plot(timevector, stat_approaching_to_door(1:step, door(:, doorMODE)==1),
3
             '-')
4 end
5 if (2 \leq \text{doorMODEsum})
       plot(timevector, stat_approaching_to_door(1:step, door(:, doorMODE)==2),
6
             '-.')
7 end
s if (3 \leq doorMODEsum)
       plot(timevector, stat_approaching_to_door(1:step, door(:, doorMODE)==3),
9
            '--')
10 end
11 if (4 \leq \text{doorMODEsum})
       plot(timevector, stat_approaching_to_door(1:step, door(:, doorMODE)==4),
12
             ':')
13 end
14 if (5 \leq doorMODEsum)
15
       plot(timevector, stat_approaching_to_door(1:step, door(:, doorMODE)==5),
            '-')
16 end
17 xlabel('time')
18 ylabel('# approaching')
19 hold off
```

Listing 26: plot_saved_moving.m

```
1 plot(timevector, stat_moving_agents(1:step, :))
2 xlabel('time')
3 ylabel('# moving')
```

Listing 27: plot_saved_distance.m

```
1 plot(timevector, stat_distance_to_go(1:step, :))
2 xlabel('time')
3 ylabel('distance')
```

```
Listing 28: plot_saved_deboarded.m
```

```
1 plot(timevector, (stat_boarded_per_door(1:step, :) < 0) .* abs(
        stat_boarded_per_door(1:step, :)))
2 xlabel('time')
3 ylabel('# deboarded')</pre>
```

```
1 plot(timevector, (stat_boarded_per_door(1:step, :) > 0) .*
        stat_boarded_per_door(1:step, :))
2 xlabel('time')
3 ylabel('# boarded')
```

Listing 30: plot_saved_waiting.m

```
1 plot(timevector, stat_waiting_agents(1:step, :))
2 hold on
3 plot(timevector, sum(stat_waiting_agents(1:step, :)')', 'r')
4 hold off
5 legend('to subway', 'to train', 'sum')
6 xlabel('time')
7 ylabel('# waiting')
```

Listing 31: plot_saved_time_waited.m

```
1 hold on
2 plot(1:agentcount, (agent(:, agentSTATE) == agentSTATEmoving) .*
        stat_sum_waiting, 'b.')
3 plot(1:agentcount, (agent(:, agentSTATE) == agentSTATEboarded) .*
        stat_sum_waiting, 'g.')
4 plot_separation_lines;
5 legend('moving', 'boarded')
6 hold off
7 xlabel('moving', 'boarded')
8 ylabel('time waited')
```

Listing 32: plot_saved_redecisions.m

```
1 hold on
2 plot(1:agentcount, (agent(:, agentSTATE) == agentSTATEmoving) .*
    stat_sum_decision, 'b.')
3 plot(1:agentcount, (agent(:, agentSTATE) == agentSTATEboarded) .*
    stat_sum_decision, 'g.')
4 plot_separation_lines;
5 legend('moving', 'boarded')
6 hold off
7 xlabel('moving', 'boarded')
8 ylabel('redecisions')
```

Listing 33: plot_saved_distance_walked.m

```
1 hold on
2 plot(1:agentcount, (agent(:, agentSTATE) == agentSTATEmoving) .*
    stat_sum_distance, 'b.')
3 plot(1:agentcount, (agent(:, agentSTATE) == agentSTATEboarded) .*
    stat_sum_distance, 'g.')
4 plot(1:agentcount, stat_min_distance, 'r.')
5 plot_separation_lines;
6 legend('moving', 'boarded', 'minimal-dist')
7 hold off
8 xlabel('agent')
9 ylabel('distance walked')
```

Listing 34: plot_separation_lines.m

8.2.6 Saving and loading simulation data

Listing 35: video_capture.m

```
1 % add picture to video
2 if video_mode == videoON
3
       new_Frame = getframe(my_figure);
4
       aviobj = addframe(aviobj, new_Frame);
\mathbf{5}
6
        if t == tmax
\overline{7}
8
            aviobj = close(aviobj);
9
        end
10
11 end
```

Listing 36: data_export.m

```
1 % save current workspace to file if activated
2 if data_export_mode == data_export_ON
3 if (mod(step,round(save_dt/dt)) == 0)
```

```
4 save(strcat(save_file_prefix,int2str(step),save_file_suffix))
5 end
6 end
```

Listing 37: load_and_playback.m

```
% playback saved simulation keyframes in real time
1
\mathbf{2}
       init_style
  for i = 1:round(tmax/save_dt)
3
       filename = strcat(save_file_prefix,num2str(i*(round(save_dt/dt))),
4
           save_file_suffix)
       load(filename)
5
       plotting_mode = plotMAPview;
6
       paint
7
       pause(save_dt)
8
9
  end
```

8.3 Simulation Results

This is the complete list of all simulated test cases, that have been used for the analysis in *section 5.3*. There are 5 samples per test case. The number of the test case corresponds to the variable that needs to be set in order to initialize the Matlabprogram. For each measurement in each test case, there is the calculated average and standard deviation in the two grey bottom lines. The last column indicates the number of agents that have not reached their destination until the end of the simulation. Especially for the very crowded setup and the random door decision mode, this number can get quite relevant. The order of the other columns matches the list of statistical measurements in *section 5.2*.

1400 6407 1.1497 64.821 7.700 2.790 2.870 0.000 5.800 5.800 5.800 6.870 0.000 5.800 5.800 5.800 6.870 0.000 5.800 7.800 8.877 1.877 8.497 1.970 2.870 6.800 6.800 0.000 5.800 7.800 8.870 7.800 8.400 0.000 5.800 7.800 8.400 8.400 8.400 8.400 8.400 8.400 8.400 8.400 <t< th=""><th>f_b_time</th><th>m_b_time</th><th>d_b_time</th><th>m_distance</th><th>d_distance</th><th>m_waiting_t</th><th>d_waiting_t</th><th>m_decisions</th><th>a_aecisions</th><th>a_acor_aistrib.</th><th>unboarded</th></t<>	f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	a_aecisions	a_acor_aistrib.	unboarded
24.260 0.288 1.2487 1.748 1.783 1.783 0.783 0.493 2.4247 0.283 0.403 4.000 0.2300 0.237 0.230 0.236 <t< td=""><td>61.900</td><td>29.647</td><td>11.497</td><td>46.921</td><td>17.070</td><td>12.769</td><td>7.812</td><td>5.750</td><td>2.679</td><td>8.272</td><td>4.000</td></t<>	61.900	29.647	11.497	46.921	17.070	12.769	7.812	5.750	2.679	8.272	4.000
11.100 11.174 4.6897 17.29 17.29 6.783 5.492 2.844 8.300 4.00 3.000 63.000 0.395 11.281 64.000 17.280 17.280 6.780 6.000 2.247 8.300 6.400 5.000 6.000 6.400 6.000	62.450	30.328	12.605	47.197	18.597	12.847	7.837	5.090	2.462	8.933	11.000
Biol Biol <t< td=""><td>61 100</td><td>29 186</td><td>11 570</td><td>45 697</td><td>17 655</td><td>11 729</td><td>6 783</td><td>5 542</td><td>2 934</td><td>8 380</td><td>4 000</td></t<>	61 100	29 186	11 570	45 697	17 655	11 729	6 783	5 542	2 934	8 380	4 000
shore shore <t< td=""><td>62 500</td><td>20.100</td><td>11.404</td><td>46.675</td><td>17.000</td><td>12,000</td><td>0.700</td><td>4.976</td><td>2.007</td><td>0.000</td><td>2,000</td></t<>	62 500	20.100	11.404	46.675	17.000	12,000	0.700	4.976	2.007	0.000	2,000
63.00 63.00 63.00 71.26 73.56 <th< td=""><td>03.500</td><td>29.541</td><td>11.494</td><td>40.075</td><td>17.829</td><td>12.000</td><td>8.342</td><td>4.876</td><td>2.247</td><td>8.530</td><td>3.000</td></th<>	03.500	29.541	11.494	40.075	17.829	12.000	8.342	4.876	2.247	8.530	3.000
e3.9893.8893.8994.916110.3911.395.2332.5.232.5.230.6.090.0000.0000.1410.3400.1410.0400.000 <td>65.600</td> <td>30.396</td> <td>12.861</td> <td>48.090</td> <td>19.040</td> <td>12.624</td> <td>8.000</td> <td>4.906</td> <td>2.437</td> <td>8.912</td> <td>4.000</td>	65.600	30.396	12.861	48.090	19.040	12.624	8.000	4.906	2.437	8.912	4.000
10.200 0.170 0.170 0.170 0.170 0.170 0.187 <	62.910	29.820	12.005	46.916	18.038	12.394	7.755	5.233	2.552	8.605	5.200
Description Description <thdescription< th=""> <thdescription< th=""></thdescription<></thdescription<>	3.023	0.275	0.450	0.750	0.611	0.249	0.340	0.154	0.069	0.092	10.700
<t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>											
b. Im. d. b. Im. <thd. b.="" im.<="" th=""> <thd.< td=""><td></td><td></td><td></td><td></td><td></td><td>2 Standard TT</td><td>2x150 Agents</td><td></td><td></td><td></td><td></td></thd.<></thd.>						2 Standard TT	2x150 Agents				
7.550 0.8637 1 0.475 0.425 0.332 2.568 6.523 0.000 0.000 7.550 0.831 1.274 4.126 1.1247 8.425 3.358 2.588 6.523 0.000 0.000 7.510 0.835 1.284 4.1360 11.847 7.480 3.58 2.413 6.690 0.000 7.510 0.837 1.280 4.1450 11.817 8.108 0.77 6.243 0.000 0.000 7.510 0.837 1.2807 1.1217 7.555 3.78 2.470 6.453 0.007 0.000 0.005 0.035 0.035 0.035 0.035 0.025 0.025 0.171 0.025 0.047 0.000 0.000 0.010 1.949 0.1400 1.949 0.235 0.235 2.471 0.660 2.416 0.607 0.000 0.1400 0.494 0.453 0.235 0.245 0.245 0.4100 1.447	f b time	m b time	d b time	m distance	d distance	m waiting t	d waiting t	m decisions	d decisions	d door distrib.	unboarded
5 5 7 2000 12.794 42.200 12.794 24.200 15.790 00.000 7.200 26.305 11.346 11.346 7.480 3.338 2.413 6.559 0.000 7.200 26.357 12.244 41.365 11.416 8.313 3.686 2.537 6.445 0.000 7.500 26.857 12.244 41.306 11.416 8.102 3.741 0.4231 6.4690 0.000 7.500 26.857 12.245 41.300 16.850 0.116 0.4231 0.000 0.471 0.000 0.471 0.000 7.500 2.501 7.510 2.501 7.510 2.501 7.510 2.501 4.500 0.000 1.000 0.000 7.500 2.501 7.510 2.501 7.510 2.501 4.510 4.500 4.510 4.510 4.500 4.510 4.500 4.510 4.510 4.510 4.510 4.510 4.510 4.510	75 650	26.887	12 870	/1 778	19/05	11.637	8 / 25	3 032	2 568	6 623	0.000
Base Base Lange Lange <thla< td=""><td>75.000</td><td>20.007</td><td>12.073</td><td>40.510</td><td>10.005</td><td>11.007</td><td>0.420</td><td>0.302</td><td>2.300</td><td>0.020</td><td>0.000</td></thla<>	75.000	20.007	12.073	40.510	10.005	11.007	0.420	0.302	2.300	0.020	0.000
9.530 0.831 1.2.944 4.2.162 11.915 11.946 7.408 3.3.89 2.4.13 8.980 0.000 73.10 0.837 12.264 41.900 11.807 11.271 7.65 3.780 2.476 6.943 0.000 0.857 12.264 41.900 11.857 11.271 7.65 3.780 2.476 6.943 0.000 0.857 0.857 12.274 41.990 11.857 4.377 5.376 0.002 0.002 0.005 0.007 0.000 1.000	75.800	27.306	13.328	42.510	19.835	11.287	8.249	3.784	2.402	0.870	0.000
77.200 28.658 11 11.244 41.580 11.680 11.347 6.313 3.686 2.537 6.453 0.000 75.510 28.657 12.244 41.580 11.416 8.028 0.728 7.750 0.2478 6.459 0.000	75.250	26.831	12.794	42.162	19.155	11.346	7.498	3.536	2.413	6.959	0.000
F5.100 B2.8070 B2.807 B2.8	77.200	26.526	11.859	41.453	17.680	11.591	8.313	3.868	2.537	6.543	0.000
55.87 28.827 12.828 41.900 11.416 8.029 0.77 0.005 6.690 0.000 1.9 0.095 0.282 0.197 0.885 0.17 0.025 0.055 0.697 0.005 0.0	75.150	26.587	12.264	41.596	18.087	11.217	7.655	3.728	2.470	6.455	0.000
Jackson Jackson Landson In June Dackson Dackson <t< td=""><td>75 010</td><td>06 907</td><td>10 605</td><td>41.000</td><td>10.050</td><td>11 416</td><td>0.000</td><td>2 770</td><td>0 470</td><td>6 600</td><td>0.000</td></t<>	75 010	06 907	10 605	41.000	10.050	11 416	0.000	2 770	0 470	6 600	0.000
UB:9 U.202 U.120 U.120 U.120 U.120 U.100 U.100 <thu< td=""><td>75.010</td><td>20.027</td><td>12.025</td><td>41.900</td><td>10.000</td><td>11.410</td><td>0.020</td><td>3.770</td><td>2.470</td><td>0.090</td><td>0.000</td></thu<>	75.010	20.027	12.025	41.900	10.000	11.410	0.020	3.770	2.470	0.090	0.000
	0.677	0.095	0.326	0.187	0.858	0.035	0.177	0.023	0.005	0.047	0.000
Lame m. b. lame d. lame m. statumo d. distance m. valing d. sum m. statumo d. door d. doo						2 Four OT	75 Agonto				
b. time m. b. time d. b. time d. distance d. distance <th< td=""><td></td><td></td><td></td><td></td><td></td><td>3 Few OT</td><td>/5 Agents</td><td></td><td></td><td></td><td></td></th<>						3 Few OT	/5 Agents				
41.000 21.584 7.473 83.580 1.2271 5.682 4.569 5.511 2.765 5.140 3.000 77.500 21.521 6.855 33.501 1.655 4.570 3.750 5.265 2.243 5.917 0.00 - 7.250 21.521 6.858 35.808 1.1436 5.457 3.750 5.740 3.155 4.71 1.000 4.300 20.337 6.329 6.732 6.328 5.777 4.649 5.786 2.020 6.330 0.000 1.300 4.300 0.237 6.321 1.138 0.286 0.775 3.405 1.448 0.000 - 3.455 2.000 3.450 0.200 3.887 0.000 - 3.457 0.000 - 3.457 0.000 - 3.457 0.000 - 3.457 0.000 - 3.457 0.000 - 3.457 0.000 - 3.457 0.000 - 3.457 0.000	f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
7.4602.11926.6505.31011.654.5705.7605.2672.4364.9670.0036.6021.8066.3896.3892.57310.3925.5174.6495.7633.1554.7110.0037.42921.8050.43820.5320.53710.3925.5174.6495.7633.1554.7110.0037.4290.5320.5320.5320.53710.3925.4334.4275.3330.3154.7285.3370.0037.4290.5320.5370.63011.3510.5434.4275.3330.4061.6170.4063.670.4071.4004.2240.3350.5720.85711.2515.814.4233.1532.3013.670.4071.003.50010.600.37511.2515.814.5163.4362.3113.753.750.3070.003.50010.600.3270.0275.6274.5623.1662.3100.0070.0070.0073.50010.600.3280.30510.745.5224.5163.5622.6661.1310.0010.0070.0073.50010.600.3270.32710.621.6662.5601.1340.5652.6661.1320.3071.1310.3013.50013.5013.5013.5513.7013.632.6652.66613.731.4663.6652.6652.6661.1321.134	40.100	21.594	7.473	35.350	12.271	5.626	4.569	5.511	2.765	5.140	3.000
71200215210.6860.39911.4365.5663.9674.6495.755.2575.2775.257 <t< td=""><td>37.450</td><td>21.192</td><td>6.855</td><td>35.310</td><td>11.655</td><td>4.570</td><td>3.750</td><td>5.265</td><td>2.439</td><td>4.967</td><td>1.000</td></t<>	37.450	21.192	6.855	35.310	11.655	4.570	3.750	5.265	2.439	4.967	1.000
98.800 91.806 93.873 10.392 91.97 4.649 5.757 4.637 5.740	37 250	21 521	6 886	35 998	11 436	5 366	3 906	4 646	2 514	5 085	2 000
0.000 0.000 <th< td=""><td>07.200</td><td>01.000</td><td>0.000</td><td>05.000</td><td>10.000</td><td>5.000</td><td>4.040</td><td>5.755</td><td>2.017</td><td>5.000</td><td>1.000</td></th<>	07.200	01.000	0.000	05.000	10.000	5.000	4.040	5.755	2.017	5.000	1.000
94.000 92.037 0.2037 0.2037 0.2136 6.727 0.216 0.216 6.728 1.200 74.59 21.200 6.732 0.231 0.231 0.231 0.236 0.036 0.036 0.036 0.036 1.300 74.50 21.200 6.732 0.231 0.236 0.242 0.336 0.242 0.036 0.036 1.300 74.60 0.506 28.74 0.706 5.777 4.620 3.405 1.877 3.425 0.000 0.000 38.600 16.133 7.16 2.767 4.620 3.405 2.807 0.000 35.300 17.306 6.760 2.326 0.774 5.522 4.515 3.533 2.275 3.837 0.000 1.100 6.760 2.336 0.774 5.520 0.435 2.455 3.437 0.435 2.457 0.367 0.427 2.1070 6.1648 0.393 0.574 0.425 4.458 0.461	30.050	21.800	0.399	33.733	10.392	5.917	4.049	5./55	3.237	5.317	1.000
37.3 (b) 2.129 6.722 6.723 6.724 9.724 9.734 9.730 9.210 9.200	34.300	20.337	6.048	33.662	9.347	5./8/	4.507	5.740	3.155	4./41	0.000
4.294 0.332 0.291 0.830 1.334 0.296 0.173 0.210 0.136 0.046 1.300 A 450 17.764 6.505 28.754 9.706 5.767 4.4620 3.405 1.477 3.425 0.000 34.650 16.177 6.956 30.199 11.251 5.891 4.620 3.405 1.474 3.425 0.000 36.860 16.167 6.968 30.199 11.260 5.181 4.427 3.159 2.202 3.847 0.000 35.300 17.966 6.952 29.173 10.657 5.657 4.525 4.365 2.969 0.000 35.300 17.966 6.952 29.173 10.657 5.657 4.525 4.966 1.162 3.807 0.000 35.300 11.990 6.052 29.473 10.657 5.582 4.596 0.207 0.287 3.859 0.020 0.102 0.102 0.102 0.102 0.102 0.102 0.102 0.103 0.000 0.000 0.000 0.000 0.000 0.000 <td>37.150</td> <td>21.290</td> <td>6.732</td> <td>35.211</td> <td>11.020</td> <td>5.453</td> <td>4.276</td> <td>5.383</td> <td>2.826</td> <td>5.050</td> <td>1.400</td>	37.150	21.290	6.732	35.211	11.020	5.453	4.276	5.383	2.826	5.050	1.400
number num num nume	4 204	0 332	0 201	0.830	1 334	0.286	0 173	0.210	0 136	0.046	1 300
IPWENT Verbet Ve	4.234	0.002	0.291	0.030	1.004	0.200	0.173	0.210	0.130	0.040	1.300
b. time b. time d. b. time m. distance d. distance m. waling 1 d. waling 1 m. decisions d. docisions d. docisions <thd. docisions<="" th=""> <thd< td=""><td></td><td></td><td></td><td></td><td></td><td>4 Few TT 2</td><td>75 Agente</td><td></td><td></td><td></td><td></td></thd<></thd.>						4 Few TT 2	75 Agente				
Laume In	f he direct	ma la time a	ما ام الاست	na diatana a	ما مانمغمیت		d mailing t	ma de aisterre	م مام ما - ! - :	الاستعاد وممام الم	under and set
34.460 17.764 5.05 28.754 9.706 5.767 4.620 3.405 1.877 3.425 0.000 36.800 16.133 7.16 29.766 11.620 5.181 4.713 3.159 2.207 3.847 0.000 35.300 17.986 6.760 29.368 10.774 5.471 4.902 3.413 2.371 3.657 0.000	I_D_time	III_D_time	u_b_time	m_distance	u_distance	m_waiting_t	u_waiting_t	III_aecisions	u_aecisions	u_aoor_distrib.	unboarded
96.860 11.8.017 9.988 30.189 11.251 5.891 4.629 3.175 1.948 4.388 0.000 34.800 11.39 7.316 2.3766 11.202 5.181 4.713 3.159 2.201 3.847 0.000 35.30 17.966 6.832 29.378 10.704 5.457 4.925 3.436 2.207 3.869 0.000 35.700 18.004 6.892 29.450 10.784 5.587 4.925 4.365 2.297 3.869 0.000 35.700 10.008 0.493 0.527 0.001 0.497 0.120 0.000 35.700 15.201 69.423 25.544 20.217 1.984 5.656 2.696 11.193 23.00 11.932 11.932 11.930 11.932 21.000 13.30 3.333 3.333 3.333 3.333 3.333 3.333 3.333 3.333 3.333 3.333 3.333 3.334 3.334 1.334 66.657	34.650	17.764	6.505	28.754	9.706	5.767	4.620	3.405	1.877	3.425	0.000
84860118-1381-3381-3381-3381-3481-5481-5481-5481-5481-5481-5481-5481-55<	36.850	18.617	6.968	30.189	11.251	5.891	4.629	3.175	1.948	4.388	0.000
44.00 17.966 6.700 2.938 10.704 5.417 4.092 3.413 2.971 3.787 0.000 53.500 17.966 6.392 29.173 10.67 5.657 4.552 4.365 2.962 3.869 0.000 1.170 0.103 6.896 29.400 10.674 5.522 4.565 3.637 0.965 0.227 0.966 0.000 52.56 4.529 15.971 69.423 25.544 20.227 11.864 5.566 2.966 11.103 2.000 1 1.000 2 1.000	36.850	18.133	7.316	29.766	11.620	5.181	4.713	3.159	2.200	3.847	0.000
35.350 17.986 8.332 29.173 10.637 5.657 4.525 4.365 2.982 3.869 0.000 35.700 16.094 6.886 2.9450 10.774 5.562 4.516 3.533 2.275 3.857 0.000 1.170 0.108 0.088 0.303 0.527 0.661 0.061 0.247 0.195 0.100 0.000 2.5 15.921 69.423 25.544 20.27 11.844 5.566 2.696 11.192 10.000 23.00 32.00 41.288 15.507 69.375 25.288 20.016 12.362 5.440 2.517 10.890 24.000 23.00	34.800	17.969	6.760	29.368	10.704	5.417	4.092	3.413	2.371	3.757	0.000
Based Privade Ex.Prio Prode Ex.Prio Prode Ex.Prio Prode Ex.Prio Prode Dots Dots <thdo< td=""><td>35 350</td><td>17 986</td><td>6.032</td><td>20 173</td><td>10.637</td><td>5 657</td><td>4 525</td><td>4 365</td><td>2 082</td><td>3 869</td><td>0.000</td></thdo<>	35 350	17 986	6.032	20 173	10.637	5 657	4 525	4 365	2 082	3 869	0.000
35.700 16.094 6.296 29.450 10.784 5.582 4.516 3.633 2.275 3.857 0.000 1.170 0.103 0.088 0.333 0.527 0.081 0.061 0.247 0.195 0.120 0.000 5 200 15.921 15.921 69.423 25.544 0.227 11.844 5.565 2.666 11.93 20.00 42.081 15.907 69.375 25.238 20.016 12.362 5.446 2.547 10.750 21.000 83.360 41.398 15.846 67.790 24.007 19.527 11.841 5.546 2.510 10.890 24.000 83.820 41.308 15.848 68.651 25.606 19.732 11.934 5.523 2.744 11.000 21.400 4.328 0.091 0.063 0.603 0.303 0.133 0.064 0.044 0.303 4.300 9.003 9.911 16.328 64.215 7.4027 19.423	00.000	17.000	0.002	20.170	10.007	0.007	4.020	4.000	2.002	0.000	0.000
1.170 0.103 0.088 0.303 0.627 0.081 0.027 0.195 0.120 0.000 5 Too Marry Lag 5 Too Marry Lag 0.4311 arg 0.4311 arg 0.4311 arg 0.4315 arg 0.431 arg 0.4316 arg 0.4316 arg 0.4316 arg 0.431 arg 0.431 arg 0.431 arg 0.441 arg 0.543 arg <td< td=""><td>35.700</td><td>18.094</td><td>6.896</td><td>29.450</td><td>10.784</td><td>5.582</td><td>4.516</td><td>3.503</td><td>2.275</td><td>3.857</td><td>0.000</td></td<>	35.700	18.094	6.896	29.450	10.784	5.582	4.516	3.503	2.275	3.857	0.000
S Too Mary C U SOD Agents L time m. bime d. bime m. distance d. distance m. waiting_1 m. decisions d. doc. distance d. doc. d. doc. <th>1.170</th> <th>0.103</th> <th>0.088</th> <th>0.303</th> <th>0.527</th> <th>0.081</th> <th>0.061</th> <th>0.247</th> <th>0.195</th> <th>0.120</th> <th>0.000</th>	1.170	0.103	0.088	0.303	0.527	0.081	0.061	0.247	0.195	0.120	0.000
Is time m.b.time d. b.time m.distance d.distance m.waiting t.d.vaiting t.m.decisions d.door.distrib.urboarded 82 950 42 529 15 921 64 distance d.distance m.waiting t.m.decisions d.door.distrib.urboarded 82 950 42 529 15 921 64 323 25 63 2 29 400 28 950 4 decisions d.door.distrib.urboarded 84 700 42 820 68 375 26 328 2000 33 30 41 200 11 934 5.523 2.744 11.000 21.400 88 42 0.603											
b b m b m b m b m b m b m d						5 Too Many O	T 300 Agents				
B2 850 42 529 15 821 69 427 69 437 22 854 20 07 11.884 5.565 2.866 11.102 23.00 81 200 41.834 15.704 69.375 26.328 20.016 12.362 5.840 3.079 11.021 10.000 21.000 83.350 41.938 15.548 66.752 26.432 19.424 11.811 5.546 2.610 10.890 24.000 83.800 42.000 15.836 66.65 26.060 19.732 11.934 5.523 2.74 11.000 24.000 83.800 39.991 16.329 64.216 2.7401 18.276 10.877 3.964 2.606 9.531 27.000 99.000 39.991 16.329 64.216 2.7401 18.276 10.477 3.964 2.600 9.637 20.001 99.000 39.991 16.329 64.216 2.742 18.492 10.443 3.802 2.559 9.037 30.000 99.000	f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
B4.700 42.288 15.907 69.375 26.238 20.016 12.362 5.440 3.079 11.021 19.001 12.00 41.834 15.704 67.916 22.700 19.425 11.573 5.346 2.547 10.750 21.000 24.000 83.820 42.200 15.799 67.790 24.027 19.527 12.041 5.517 2.588 11.147 20.000 21.400 83.820 42.200 15.836 68.651 25.600 10.732 0.086 0.044 0.045 0.003 4.300 45.56 0.091 0.002 0.603 0.903 0.133 0.086 0.044 0.045 0.003 4.300 94.00 39.91 16.326 64.216 27.401 18.276 10.877 3.964 2.606 9.531 27.00 9.919 10.329 9.64.216 27.412 18.431 11.144 3.943 2.600 9.323 28.000 9.446 2.949 27.400 9.410	82.950	42.529	15.921	69.423	25.544	20.227	11.884	5.565	2.696	11.193	23.000
No.0 No.0 <th< td=""><td>84 700</td><td>12 288</td><td>15 907</td><td>69 375</td><td>26.238</td><td>20.016</td><td>12 362</td><td>5.840</td><td>3 079</td><td>11 021</td><td>19,000</td></th<>	84 700	12 288	15 907	69 375	26.238	20.016	12 362	5.840	3 079	11 021	19,000
11.034 11.034 11.034 11.035 23.046 2.347 10.730 21.000 83.06 41.084 15.790 26.432 19.424 11.613 5.546 2.619 10.890 24.000 10.890 24.000 83.820 42.200 15.836 68.661 25.606 19.732 11.934 5.523 2.744 11.000 21.400 4.526 0.091 0.008 0.603 0.903 0.133 0.086 0.044 0.045 0.033 4.300 94.00 39.91 15.326 64.216 27.401 18.276 10.877 3.964 2.606 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 27.00 9.531 9.73 20.00 <td>81.000</td> <td>41.004</td> <td>15 704</td> <td>67.016</td> <td>26.200</td> <td>10.465</td> <td>11 572</td> <td>5.040</td> <td>0.070</td> <td>10.750</td> <td>21,000</td>	81.000	41.004	15 704	67.016	26.200	10.465	11 572	5.040	0.070	10.750	21,000
333 30 1 10.848 10.849 10.849 10.840 24.00 24.000 83.820 41.938 10.840 24.021 19.827 12.041 5.346 2.810 10.840 24.000 83.820 42.200 15.836 68.651 25.606 19.732 11.934 5.543 2.744 11.000 24.000 45.26 0.091 0.008 0.603 0.903 0.133 0.066 0.044 0.045 0.033 4.300 45.90 42.16 7.401 18.276 10.877 3.964 2.606 9.531 27.000 99.050 40.101 15.957 64.873 27.823 18.082 10.443 4.108 2.613 9.865 27.000 99.000 39.911 16.411 64.373 27.823 18.082 10.413 3.943 2.600 9.323 28.000 90.003 9.040 40.423 17.175 64.973 27.45 18.239 10.413 3.943 2.600 <th< td=""><td>81.200</td><td>41.034</td><td>15.704</td><td>07.910</td><td>23.790</td><td>19.403</td><td>11.573</td><td>5.540</td><td>2.347</td><td>10.750</td><td>21.000</td></th<>	81.200	41.034	15.704	07.910	23.790	19.403	11.573	5.540	2.347	10.750	21.000
96.900 42.410 15.799 67.790 24.027 19.527 12.041 5.317 2.588 11.147 20.000 83.820 42.200 15.836 68.651 25.606 19.732 11.394 5.523 2.744 11.000 21.400 4.526 0.091 0.091 0.638 0.663 0.903 0.133 0.066 0.044 0.045 0.033 4.300 99.400 39.991 16.329 64.216 27.401 18.276 10.877 3.964 2.606 9.531 27.000 9.805 40.100 15.857 64.873 27.823 18.402 10.443 3.804 2.613 9.865 27.000 99.000 39.929 16.394 64.231 27.452 18.239 10.419 3.945 2.665 9.437 2.000 89.100 39.911 16.414 0.453 0.6452 27.662 18.239 10.419 3.945 2.655 9.439 27.400 89.100 0.044 <	83.350	41.938	15.848	68.752	26.432	19.424	11.811	5.546	2.810	10.890	24.000
83.820 42.200 15.836 668.651 25.606 19.732 11.934 5.523 2.744 11.000 21.400 4.526 0.091 0.008 0.603 0.003 0.133 0.086 0.044 0.045 0.005	86.900	42.410	15.799	67.790	24.027	19.527	12.041	5.317	2.588	11.147	20.000
4.526 0.091 0.008 0.603 0.093 0.133 0.096 0.044 0.045 0.033 4.300 b time d_b time m_distance d_distance m_waiting_1 d_waiting_1 m_decisions d_decisions d_door_distrib. unboarded 94.00 39.991 16.329 64.216 27.401 18.276 10.443 3.802 2.559 9.037 30.000 99.400 39.929 16.344 64.231 27.823 18.082 10.443 3.802 2.559 9.037 30.000 99.100 39.929 16.344 64.231 27.435 18.239 10.419 3.945 2.746 9.440 25.000 89.100 40.071 16.453 64.529 27.622 18.266 10.765 3.952 2.625 9.439 27.400 89.000 29.366 16.033 46.150 24.820 12.246 10.131 1.730 0.744 9.171 2.000 87.000 29.366 16.0	83.820	42.200	15.836	68.651	25.606	19.732	11.934	5.523	2.744	11.000	21.400
A.326 0.005 <th< td=""><td>4 506</td><td>0.001</td><td>0.009</td><td>0.602</td><td>0.002</td><td>0 199</td><td>0.096</td><td>0.044</td><td>0.045</td><td>0.022</td><td>4 200</td></th<>	4 506	0.001	0.009	0.602	0.002	0 199	0.096	0.044	0.045	0.022	4 200
bitme bitme distance dis	4.520	0.091	0.008	0.003	0.903	0.133	0.000	0.044	0.045	0.033	4.300
b_time m_b_time d_b_time m_distance d_distance m_waiting_t m_decisions d_decisions d_door_distrib. unboarded 89.400 39.991 16.329 64.216 27.401 18.276 10.877 3.964 2.606 9.531 27.000 99.050 40.100 15.957 64.873 27.823 18.082 10.443 3.802 2.559 9.337 30.000 99.050 39.911 16.411 64.349 27.212 18.431 11.144 3.943 2.600 9.323 28.000 99.100 39.911 16.453 64.231 27.435 18.239 10.419 3.945 2.746 9.440 25.000 89.100 30.944 0.197 0.135 0.169 0.028 0.1012 0.005 0.091 3.300 80.00 29.446 15.995 46.219 23.876 12.491 10.131 1.730 0.744 9.171 2.000 87.400 29.366 16.603 46.						6 Too Mony TT	2x200 Agonto				
b_time d_b_time m_distance d_distance m_waiting_t m_waiting_t m_decisions d_decisions d_door_distrib unboarded 89.400 39.940 15.957 64.873 27.823 18.082 10.443 3.802 2.606 9.531 27.000 89.100 40.423 17.175 64.978 28.240 18.402 10.443 3.802 2.559 9.037 30.000 89.100 39.929 16.394 64.231 27.422 18.230 10.419 3.945 2.746 9.440 25.000 89.100 39.929 16.394 64.231 27.422 18.236 10.765 3.952 2.625 9.439 27.400 89.100 0.044 0.197 0.135 0.169 0.020 0.103 0.012 0.091 3.300 89.100 9.946 15.95 46.219 23.876 12.491 10.131 1.730 0.744 9.171 2.000 88.00 29.466 15.995 46.21						0 100 Wally 11	ZASOU Agents				
99.900 39.991 16.329 64.216 27.401 18.276 10.477 3.964 2.606 9.531 27.000 99.050 40.100 15.957 64.978 27.823 18.062 10.443 3.802 2.559 9.037 30.000 99.100 39.911 16.411 64.349 27.22 18.431 11.144 3.943 2.600 9.323 28.000 89.000 39.911 16.453 64.252 27.62 18.239 10.419 3.945 2.766 9.440 25.00 9.300 3.952 2.625 9.439 25.00 89.000 39.9446 15.957 64.219 23.76 12.491 10.131 1.730 0.74 9.171 2.000 87.400 29.366 16.603 46.159 24.85 12.339 9.029 1.722 0.666 9.818 2.000 87.400 29.366 15.757 46.715 23.95 12.638 9.630 1.713 0.708 9.870	f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
89.050 40.100 15.957 64.873 27.823 18.082 10.443 3.802 2.559 9.037 30.000 89.100 40.423 17.175 64.978 28.240 18.402 10.443 4.108 2.613 9.865 27.000 89.100 39.929 16.394 64.231 27.435 18.239 10.419 3.945 2.625 9.440 25.000 89.100 0.044 0.197 0.135 64.529 27.622 18.286 10.765 3.952 2.625 9.430 27.400 89.100 0.044 0.197 0.135 64.529 0.020 0.103 0.012 0.005 4.620 15.99 46.150 24.820 12.491 10.131 1.730 0.744 9.171 2.000 88.800 29.366 15.603 46.150 24.820 12.270 9.470 1.784 0.807 8.861 0.000 73.600 29.366 15.647 46.643 22.485 12.638 <	89.400	39.991	16.329	64.216	27.401	18.276	10.877	3.964	2.606	9.531	27.000
B9.400 40.423 17.175 64.978 28.240 18.402 10.943 4.108 2.613 9.865 27.00 99.100 39.911 16.411 64.349 27.212 18.431 11.144 3.943 2.600 9.323 28.000 80.000 39.929 16.394 64.231 27.435 18.239 10.019 3.945 2.746 9.440 25.000 80.003 0.044 0.197 0.135 0.169 0.020 0.103 0.012 0.005 0.091 3.300 Storter: Transition of the store of the s	89.050	40.100	15.957	64.873	27.823	18.082	10.443	3.802	2.559	9.037	30.000
89.100 39.911 16.411 64.349 27.212 18.431 11.144 3.943 2.600 9.323 28.000 89.190 40.071 16.453 64.231 27.435 18.239 10.419 3.945 2.746 9.440 25.000 89.190 40.071 16.453 64.529 27.622 18.286 10.765 3.952 2.625 9.439 27.400 0.038 0.044 0.197 0.135 0.169 0.020 0.103 0.012 0.005 0.091 3.300 Toordecisions d_decisions d_decisions d_decisions d_decisions d_decisions d_door_distrib. unboarded 88.00 29.446 15.995 46.219 23.876 12.491 10.131 1.730 0.744 9.171 2.000 87.400 29.366 15.247 46.643 22.485 12.339 9.029 1.722 0.696 9.818 2.000 88.00 29.526 15.509 46.715 23.956 12.638 9.630 1.713 0.733 9.478 3.000	89.400	40.423	17.175	64.978	28.240	18.402	10.943	4.108	2.613	9.865	27.000
borror borror brord brord <td>89 100</td> <td>39 911</td> <td>-</td> <td>64 349</td> <td>27 212</td> <td>18 431</td> <td>11 144</td> <td>3 943</td> <td>2 600</td> <td>9 323</td> <td>28.000</td>	89 100	39 911	-	64 349	27 212	18 431	11 144	3 943	2 600	9 323	28.000
03.000 03.325 10.394 04.201 27.405 10.295 10.419 3.345 2.740 9.440 25.000 89.190 40.071 16.453 64.529 27.622 18.286 10.765 3.952 2.625 9.439 27.400 0.038 0.044 0.197 0.135 0.169 0.020 0.103 0.012 0.005 0.091 3.300 Toportecision Walk to time d_b_time m_distance d_distance m_waiting_t d_waiting_t m_decisions d_decisions d_door_distrib unboarded 88.800 29.396 15.247 46.643 22.485 12.339 9.029 1.722 0.696 9.818 2.000 88.800 29.396 15.247 46.643 22.485 12.688 9.630 1.713 0.708 9.870 6.000 88.800 29.396 15.509 46.766 23.194 12.668 9.630 1.713 0.701 9.667 5.000 85.460 29.410 <	80.000	20.020	16 204	64 221	27 425	19 220	10.410	2 045	2.746	0.440	25.000
89.190 40.071 16.453 64.529 27.622 18.286 10.765 3.952 2.625 9.439 27.400 0.038 0.044 0.197 0.135 0.169 0.020 0.103 0.012 0.005 0.091 3.300 F bordecisions d_btime m_distance d_distance m_waiting_t d_waiting_t m_decisions d_decisions d_door_distrib. unboarded 88.800 29.446 15.995 46.219 23.876 12.491 10.131 1.730 0.744 9.171 2.000 87.400 29.366 16.603 46.150 24.820 12.270 9.470 1.784 0.807 8.861 0.007 88.700 29.315 15.375 46.715 23.956 12.638 9.630 1.713 0.708 9.870 6.000 88.800 29.526 15.509 46.766 23.194 12.668 9.654 1.722 0.710 9.667 5.000 85.460 29.410	09.000	39.929	10.394	04.231	21.433	10.239	10.419	3.945	2.740	9.440	20.000
0.038 0.044 0.197 0.135 0.169 0.020 0.103 0.012 0.005 0.091 3.300 <i>I</i> burne m_bitme d_b_time m_distance m_waiting_t 0_waiting_t m_decisions 0_decisions 0_decisions 0_door_distrib. whobarded 88.800 29.446 15.995 46.219 23.876 12.491 10.131 1.730 0.744 9.171 2.000 87.400 29.366 16.603 46.150 24.820 12.270 9.470 1.784 0.807 8.861 0.000 73.600 29.396 15.247 46.643 22.485 12.339 9.029 1.722 0.696 9.818 2.000 88.800 29.315 15.375 46.716 23.996 12.688 9.650 1.712 0.701 9.667 5.000 44.308 0.000 0.310 0.085 0.769 0.031 0.157 0.001 0.002 0.195 6.000 5.050 25.401 15.74 46.499 <	89.190	40.071	16.453	64.529	27.622	18.286	10.765	3.952	2.625	9.439	27.400
T Doordecision Walk b_time d_b_time m_distance d_distance m_waiting_t d_waiting_t m_decisions d_decisions d_door_distrib. unboarded 88.800 29.446 15.995 46.219 23.876 12.491 10.131 1.730 0.744 9.171 2.000 87.400 29.366 16.603 46.150 24.820 12.270 9.470 1.784 0.807 8.861 0.000 73.600 29.396 15.247 46.643 22.485 12.339 9.029 1.712 0.708 9.870 6.000 88.700 29.315 15.575 46.715 23.956 12.688 9.630 1.713 0.708 9.870 6.000 85.460 29.410 15.76 46.499 23.666 12.481 9.583 1.734 0.733 9.478 3.000 44.308 0.006 0.310 0.085 0.769 0.031 0.157 0.001 0.002 0.195 6.000	0.038	0.044	0.197	0.135	0.169	0.020	0.103	0.012	0.005	0.091	3.300
Abstract											
Liner d_b_time m_distance d_distance m_waiting_t m_waiting_t m_decisions d_decisions d_door_distrib. unboarded 88.800 29.446 15.995 46.219 23.876 12.491 10.131 1.730 0.744 9.171 2.000 87.400 29.366 16.603 46.150 24.820 12.270 9.470 1.784 0.807 8.861 0.000 73.600 29.396 15.247 46.643 22.485 12.339 9.029 1.722 0.696 9.818 2.000 88.700 29.315 15.375 46.715 23.956 12.688 9.630 1.713 0.708 9.870 6.000 88.800 29.410 15.746 46.499 23.666 12.481 9.653 1.734 0.733 9.478 3.000 44.308 0.006 0.310 0.085 0.769 0.031 0.157 0.001 0.002 0.195 6.000 15.200 25.481 8.403						7 Doordooi	sion Walk				
□ □	f h time	m h time	d b time	m diatance	d distores	m woiting t	d woiting t	m dooisiana	d dooisisse	d door distrib	unhoorded
b88.800 29.446 15.995 46.219 23.876 12.491 10.131 1.730 0.744 9.171 2.000 87.400 29.366 16.603 46.150 24.820 12.270 9.470 1.784 0.807 8.861 0.000 73.600 29.396 15.247 46.643 22.485 12.339 9.029 1.722 0.696 9.818 2.000 88.700 29.351 15.375 46.715 23.956 12.638 9.630 1.713 0.708 9.870 6.000 88.800 29.526 15.509 46.766 23.194 12.668 9.654 1.722 0.710 9.667 5.000 85.460 29.410 15.76 46.499 23.666 12.481 9.583 1.734 0.733 9.478 3.000 44.308 0.006 0.310 0.085 0.769 0.031 0.157 0.001 0.002 0.195 6.000 51.200 25.481 8.403 40.439 <td>i_u_ume</td> <td>III_D_UINe</td> <td>u_u_ume</td> <td>m_uistance</td> <td></td> <td>m_walling_t</td> <td>u_waiting_t</td> <td>III_UECISIONS</td> <td></td> <td>u_uoor_uistrib.</td> <td>Depresentation</td>	i_u_ume	III_D_UINe	u_u_ume	m_uistance		m_walling_t	u_waiting_t	III_UECISIONS		u_uoor_uistrib.	Depresentation
87.400 29.366 16.603 46.150 24.820 12.270 9.470 1.784 0.807 8.861 0.000 73.600 29.396 15.247 46.643 22.485 12.339 9.029 1.722 0.696 9.818 2.000 88.700 29.315 15.375 46.715 23.956 12.638 9.630 1.713 0.708 9.870 6.000 88.700 29.315 15.509 46.766 23.194 12.668 9.654 1.722 0.710 9.667 5.000 85.460 29.410 15.764 46.499 23.666 12.481 9.653 1.734 0.733 9.478 3.000 44.308 0.006 0.310 0.085 0.666 12.481 9.583 1.734 0.032 0.195 6.000 16.100 0.310 0.085 0.666 12.481 9.583 1.734 0.032 0.025 0.000 0.002 0.000 0.002 0.000 0.000 0.000	88.800	29.446	15.995	46.219	23.876	12.491	10.131	1.730	0.744	9.171	2.000
73.600 29.396 15.247 46.643 22.485 12.339 9.029 1.722 0.696 9.818 2.000 88.700 29.315 15.375 46.715 23.956 12.638 9.630 1.713 0.708 9.870 6.000 88.800 29.526 15.509 46.766 23.194 12.668 9.654 1.722 0.710 9.667 5.000 85.460 29.410 15.76 46.499 23.666 12.481 9.583 1.734 0.703 9.478 3.000 44.308 0.08 0.300 0.085 0.769 0.031 0.167 0.001 0.002 0.19 6.000 Stordecistant multimation of the store of	87.400	29.366	16.603	46.150	24.820	12.270	9.470	1.784	0.807	8.861	0.000
88.700 29.315 15.375 46.715 23.956 12.638 9.630 1.713 0.708 9.870 6.000 88.800 29.526 15.509 46.766 23.194 12.668 9.654 1.722 0.710 9.667 5.000 85.460 29.410 15.764 46.499 23.666 12.481 9.583 1.734 0.733 9.478 3.000 44.308 0.006 0.310 0.085 0.769 0.031 0.157 0.001 0.002 0.195 6.000 b_time m_b_time d_b_time m_distance d_distance m_waiting_t d_waiting_t m_decisions d_decisions d_door_distrib unboarded 65.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 65.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.400 25.655 9.467 40.798 15.944 8.466 6.776 35.500 40.514<	73.600	29.396	15.247	46.643	22.485	12.339	9.029	1.722	0.696	9.818	2.000
Line Line <thlin< th=""> Line Line L</thlin<>	88,700	29.315	15.375	46.715	23.956	12.638	9.630	1.713	0.708	9.870	6.000
bittory 25.020 15.09 40.700 25.194 12.000 9.054 1.722 0.710 9.057 5.000 85.460 29.410 15.746 46.499 23.666 12.481 9.583 1.734 0.733 9.478 3.000 44.308 0.006 0.310 0.085 0.769 0.031 0.157 0.001 0.002 0.195 6.000 Boordecisions d_decisions d_decisions d_doc_distrib. unboarded 61.200 25.481 8.403 40.439 12.990 9.368 6.685 27.016 29.284 4.260 0.000 65.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.450 25.655 9.467 40.798 15.944 8.466 6.776 35.500 40.514 4.345 3.000 65.700 25.538 9.043 41.006 15.087 8.458 6.428 38.316	00.700	20.526	15 500	46 766	22 104	12.669	0.654	1 722	0.710	0.667	5.000
85.460 29.410 15.746 46.499 23.666 12.481 9.583 1.734 0.733 9.478 3.000 44.308 0.006 0.310 0.085 0.769 0.031 0.157 0.001 0.002 0.195 6.000 bt/me m_bt/me d_b_time m_distance d_distance m_waiting_t d_waiting_t m_decisions d_decisions d_door_distrib unboarded 61.200 25.481 8.403 40.439 12.990 9.368 6.685 27.016 29.284 4.260 0.000 65.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.450 25.655 9.467 40.798 15.944 8.456 6.776 35.500 40.514 4.345 3.000 65.010 25.631 9.426 41.100 15.399 7.808 6.428 38.316 42.204 4.403 1.000 65.100 25.637	00.000	23.320	10.009	-0.700	20.134	12.000	0.004	1.122	0.710	3.007	5.000
44.308 0.006 0.310 0.085 0.769 0.031 0.157 0.001 0.002 0.195 6.000 b_time m_b_time d_b_time m_distance d_distance m_waiting_t d_waiting_t m_decisions d_decisions d_door_distrib. unboarded 61.200 25.481 8.403 40.439 12.990 9.368 6.685 27.016 29.284 4.260 0.000 65.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.450 25.665 9.467 40.798 15.944 8.466 6.776 35.500 40.514 4.345 3.000 64.700 25.638 9.043 41.006 15.087 8.458 6.428 38.316 42.204 4.403 1.000 62.100 25.601 9.426 41.160 15.399 7.808 6.494 44.692 50.620 4.520 1.000 62.100 25.537 <td>85.460</td> <td>29.410</td> <td>15.746</td> <td>46.499</td> <td>23.666</td> <td>12.481</td> <td>9.583</td> <td>1.734</td> <td>0.733</td> <td>9.478</td> <td>3.000</td>	85.460	29.410	15.746	46.499	23.666	12.481	9.583	1.734	0.733	9.478	3.000
B Doordecision Queue m_decisions d_decisions d_door_distrib. unboarded 61.200 25.481 8.403 40.439 12.990 9.368 6.685 27.016 29.284 4.260 0.000 55.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.450 25.665 9.467 40.798 15.944 8.466 6.776 35.500 40.514 4.345 3.000 56.700 25.538 9.043 41.006 15.087 8.458 6.428 38.316 42.204 4.403 1.000 65.100 25.501 9.426 41.160 15.399 7.808 6.494 44.692 50.620 4.520 1.000 62.100 25.537 9.118 40.806 14.799 8.475 6.563 35.768 40.168 4.399 1.400 11.949 0.010 0.188 0.083 1.269 0.321 0.025 42.202	44.308	0.006	0.310	0.085	0.769	0.031	0.157	0.001	0.002	0.195	6.000
bitme m_bitme d_b_time m_distance d_distance m_waiting_t d_waiting_t m_decisions d_decisions d_door_distrib unboarded 61.200 25.481 8.403 40.439 12.990 9.368 6.685 27.016 29.284 4.260 0.000 65.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.450 25.658 9.467 40.798 15.944 8.466 6.776 35.500 40.514 4.345 3.000 65.070 25.538 9.043 41.006 15.994 8.456 6.428 38.316 42.204 4.403 1.000 65.100 25.651 9.043 41.160 15.399 7.808 6.428 38.316 42.024 4.403 1.000 62.100 25.537 9.118 40.806 14.799 8.475 6.653 35.768 40.168 4.399 1.000 1.300 11.949 </td <td></td>											
B_btime d_b_time d_bime d_bime <thdbime< th=""> <thdbim< th=""> d_bim<td></td><td></td><td></td><td></td><td></td><td>0 Do anda -</td><td>ion Our</td><td></td><td></td><td></td><td></td></thdbim<></thdbime<>						0 Do anda -	ion Our				
Lb_time m_b_time d_btime m_distance d_distance m_waiting_t d_waiting_t m_decisions d_decisions d_door_distrib. unboarded 61.200 25.481 8.403 40.439 12.990 9.368 6.685 27.016 29.284 4.260 0.000 65.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.450 25.665 9.467 40.798 15.944 8.466 6.776 35.500 40.514 4.345 3.000 65.700 25.538 9.043 41.006 15.087 8.458 6.428 38.316 42.204 4.403 1.000 65.700 25.601 9.426 41.160 15.399 7.808 6.494 44.692 50.620 4.520 1.000 62.100 25.537 9.118 40.806 14.799 8.475 6.563 35.768 40.168 4.399 1.400 11.949 0.01						o Doordecis	son Queue				
61.200 25.481 8.403 40.439 12.990 9.368 6.685 27.016 29.284 4.260 0.000 65.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.402 9.565 9.467 40.798 15.944 8.466 6.776 35.500 40.514 4.345 3.000 56.700 25.538 9.043 41.006 15.097 8.458 6.428 38.316 42.204 4.403 1.000 65.100 25.601 9.426 41.160 15.399 7.808 6.494 44.692 50.620 4.520 1.000 62.100 25.637 9.118 40.806 14.799 8.475 6.562 35.768 40.168 4.399 1.400 11.949 0.010 0.188 0.033 1.269 0.321 0.025 42.202 58.944 0.010 1.300 <	t_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
65.050 25.402 9.253 40.627 14.575 8.277 6.432 33.317 38.217 4.466 2.000 62.450 25.665 9.467 40.798 15.944 8.466 6.776 35.500 40.514 4.345 3.000 56.700 25.538 9.043 41.000 15.087 8.458 6.428 38.316 42.204 4.403 1.000 65.100 25.601 9.426 41.160 15.399 7.808 6.494 44.692 50.620 4.520 1.000 62.100 25.537 9.118 40.806 14.799 8.475 6.563 35.768 40.168 4.395 1.040 11.949 0.010 0.188 0.083 1.269 0.321 0.025 42.202 58.944 0.010 1.300	61.200	25.481	8.403	40.439	12.990	9.368	6.685	27.016	29.284	4.260	0.000
62.450 25.665 9.467 40.798 15.944 8.466 6.776 35.500 40.514 4.345 3.000 56.700 25.538 9.043 41.006 15.087 8.458 6.428 38.316 42.204 4.403 1.000 65.100 25.601 9.426 41.160 15.399 7.808 6.494 44.692 50.620 4.520 1.000 62.100 25.537 9.118 40.806 14.799 8.475 6.563 35.768 40.168 4.399 1.400 11.949 0.010 0.188 0.083 1.269 0.321 0.025 42.202 58.944 0.010 1.300	65.050	25.402	9.253	40.627	14.575	8.277	6.432	33.317	38.217	4.466	2.000
Cline Cline <th< td=""><td>62 450</td><td>25 665</td><td>9 467</td><td>40 798</td><td>15 944</td><td>8 466</td><td>6 776</td><td>35 500</td><td>40 514</td><td>4 345</td><td>3 000</td></th<>	62 450	25 665	9 467	40 798	15 944	8 466	6 776	35 500	40 514	4 345	3 000
50.700 23.505 5.045 41.000 15.067 0.436 0.426 35.310 42.204 4.403 1.000 65.100 25.601 9.426 41.160 15.399 7.808 6.494 44.692 50.620 4.520 1.000 62.100 25.537 9.118 40.806 14.799 8.475 6.563 35.768 40.168 4.399 1.400 11.949 0.010 0.188 0.083 1.269 0.321 0.025 42.202 58.944 0.010 1.300	56 700	25.500	0.042	41,006	15.097	9.459	6 429	29.216	12 204	4 402	1.000
box.100 25.001 9.420 41.100 15.399 7.808 6.494 44.692 50.620 4.520 1.000 62.100 25.537 9.118 40.806 14.799 8.475 6.563 35.768 40.168 4.399 1.400 11.949 0.010 0.188 0.083 1.269 0.321 0.025 42.202 58.944 0.010 1.300	00.700	20.000	0.400	41.400	15.007	7 000	0.420	44.000	72.204	4.500	1.000
62.100 25.537 9.118 40.806 14.799 8.475 6.563 35.768 40.168 4.399 1.400 11.949 0.010 0.188 0.083 1.269 0.321 0.025 42.202 58.944 0.010 1.300	05.100	∠0.0U1	9.420	41.100	10.399	1.000	0.494	44.092	JU.02U	4.020	1.000
11.949 0.010 0.188 0.083 1.269 0.321 0.025 42.202 58.944 0.010 1.300											
	62.100	25.537	9.118	40.806	14.799	8.475	6.563	35.768	40.168	4.399	1.400

1 Standard OT 150 Agents

						9 Doordec	ision wait				
f_b_time	m_b_time	d_b_time	m_distand	се	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
73.550	36.762	17.086	64.487		30.654	2.593	1.714	453.460	283.250	7.017	8.000
69.650	34.515	15.604	59.869		26.887	2.631	1.825	406.080	241.010	6.887	8.000
71 800	37 282	17 124	64 554		29 698	2 999	2 527	458 170	291 140	7 553	8 000
65 700	34 534	15 362	60 559		27 247	2 613	1 969	410 590	260 270	6.890	8,000
79.400	27 162	10.002	64 700		22 512	2 749	1.069	449.670	209.590	6 993	7,000
70.400	37.102	10.000	04.709		32.312	2.740	1.900	440.070	290.000	0.000	7.000
71.820	36.051	16.702	62	2.836	29.400	2.717	2.000	435.394	274.850	7.046	7.800
22.113	1.979	1.498	5	5.793	5.574	0.028	0.098	623.982	564.190	0.084	0.200
						10 Doordecis	ion Random				
f_b_time	m_b_time	d_b_time	m_distand	се	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
89.050	55.495	19.459	97.933		35.225	2.903	2.199	916.030	332.740	9.297	185.000
89.900	54.120	20.045	96.691		35.207	2.679	1.838	886.390	340.120	9.764	176.000
89.050	54.520	18.165	96.837		33.334	2.605	1.795	899.890	323.040	8.797	186.000
89.050	55.136	18.319	97.886		32.792	2.753	2.515	905.450	310.660	9.480	181.000
89.150	54.549	18.868	96.719		34.699	2.539	2.066	896.380	328.380	9.385	182.000
80.240	54 764	19 071	07	7 01 0	24 251	2 606	2 082	000 929	226 089	0 244	192.000
0 1 20	0.009	0.610	51	1.210	1 050	2.030	0.002	100.020	100 416	0.104	15 500
0.138	0.298	0.619	Ľ	0.407	1.206	0.020	0.080	120.396	122.410	0.124	15.500
						11 a:	rv 0.0				
f h timo	m h timo	d h timo	m distant	~~	d distanco	m waiting t	d waiting t	m docisions	d docisions	d door distrib	upboardod
				ce		n_waiting_t	u_waiting_t				
50.000	25.390	9.204	40.731		15.516	0.721	0.190	27.700	30.521	4.220	0.000
50.300	25.315	9.010	40.345		14.601	8.202	0.547	35.344	39.317	4.489	2.000
09.100	25.8/1	9.998	41.506		10.1/2	8.004	0.090	41.220	44.501	4.520	1.000
/1.900	25.564	9.712	40.751		16.357	8.747	6.641	30.484	34.906	4.403	1.000
72.400	25.600	9.791	40.948		16.376	8.332	6.283	36.400	40.234	4.403	1.000
66.580	25.549	9.543	40	0.856	15.765	8.413	6.472	34.231	37.908	4.408	1.000
46.407	0.046	0.174	C	0.180	0.611	0.101	0.049	27.845	28.809	0.013	0.500
						12 La:	zv 0.1				
f b time	m b time	d b time	m distant	се	d distance	m waiting t	d waiting t	m decisions	d decisions	d door distrib	unboarded
/8 100	25 108	8 518	30 5/2		13 /69	0 246	6 727	15 232	14.408	4 726	0.000
40.100	25.150	0.010	20.204		10.707	0.470	7.009	16.216	16.012	4.720	0.000
40.200	23.010	0.002	00.101		10.444	9.472	7.000	10.010	10.915	4.700	0.000
45.250	24.702	0.000	39.101		10.050	9.200	0.000	10.000	10.303	4.551	0.000
46.350	24.927	8.960	39.625		13.956	9.229	6.683	18.080	18.296	4.645	0.000
50.700	25.263	8.914	39.732		13.960	9.433	6.860	14.832	14.786	4.614	0.000
47.720	25.032	8.718	39	9.489	13.711	9.316	6.825	16.223	16.158	4.644	0.000
4.308	0.041	0.042	C	0.050	0.063	0.016	0.016	1.640	2.534	0.006	0.000
						13 La:	zv 0.2				
f b time	m b time	d b time	m distan	се	d distance	13 La: m waiting t	zy 0.2	m decisions	d decisions	d door distrib	unboarded
f_b_time	m_b_time	d_b_time	m_distant	се	d_distance	13 Laz m_waiting_t 9 440	zy 0.2 d_waiting_t 7 114	m_decisions	d_decisions	d_door_distrib.	unboarded
f_b_time 51.900	m_b_time 25.575 25.516	d_b_time 10.178	m_distant 40.594	се	d_distance 15.922 15.506	13 La: m_waiting_t 9.440 9.437	zy 0.2 d_waiting_t 7.114 7.236	m_decisions 11.176 9.700	d_decisions 8.947 8.527	d_door_distrib. 5.491 5.350	unboarded 0.000
f_b_time 51.900 48.200	m_b_time 25.575 25.516 25.382	d_b_time 10.178 9.982	m_distant 40.594 40.420	се	d_distance 15.922 15.506	13 La: m_waiting_t 9.440 9.437 9.250	d_waiting_t 7.114 7.236	m_decisions 11.176 9.700	d_decisions 8.947 8.527 7.408	d_door_distrib. 5.491 5.350 5.107	unboarded 0.000 0.000
f_b_time 51.900 48.200 47.400	m_b_time 25.575 25.516 25.382	d_b_time 10.178 9.982 9.345	m_distand 40.594 40.420 39.851	ce	d_distance 15.922 15.506 14.772	13 La: m_waiting_t 9.440 9.437 9.259 0.214	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057	m_decisions 11.176 9.700 9.156 10.100	d_decisions 8.947 8.527 7.408	d_door_distrib. 5.491 5.350 5.197	unboarded 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 46.700	m_b_time 25.575 25.516 25.382 25.551	d_b_time 10.178 9.982 9.345 9.287	m_distant 40.594 40.420 39.851 40.420	ce	d_distance 15.922 15.506 14.772 14.318	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.512	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 0.057	m_decisions 11.176 9.700 9.156 10.492	d_decisions 8.947 8.527 7.408 8.710	d_door_distrib. 5.491 5.350 5.197 5.403 5.403	unboarded 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250	m_b_time 25.575 25.516 25.382 25.551 25.416	d_b_time 10.178 9.982 9.345 9.287 9.125	m_distand 40.594 40.420 39.851 40.420 40.571	ce	d_distance 15.922 15.506 14.772 14.318 14.579	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542	d_waiting_t 7.114 7.236 6.274 7.057 6.676	m_decisions 11.176 9.700 9.156 10.492 10.240	d_decisions 8.947 8.527 7.408 8.710 8.618	d_door_distrib. 5.491 5.350 5.197 5.403 5.178	unboarded 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583	m_distand 40.594 40.420 39.851 40.420 40.571 40	се 0.371	d_distance 15.922 15.506 14.772 14.318 14.579 15.019	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488 0.007	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217	m_distant 40.594 40.420 39.851 40.420 40.571 40.571	ce 0.371 0.091	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488 0.007	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217	m_distant 40.594 40.420 39.851 40.420 40.571 40.571	ce 0.371 0.091	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450	13 Laz m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592	d_decisions 8.947 8.527 7.408 8.710 8.618 8.618 8.442 0.359	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488 0.007	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217	m_distand 40.594 40.420 39.851 40.420 40.571 40 0	ce 0.371 0.091	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018 14 La:	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480 f_b_time	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488 0.007 m_b_time	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217 d_b_time	m_distant 40.594 40.420 39.851 40.420 40.571 40 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ce 0.371 0.091	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018 14 La: m_waiting_t	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 ry 0.3 d_waiting_t	m_decisions 11.176 9.700 9.156 10.492 10.240 10.2592 m_decisions	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib.	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480 f_b_time 50.450	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488 0.007 m_b_time 25.217	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217 d_b_time 9.417	m_distand 40.594 40.420 39.851 40.420 40.571 40 0 0 0 0 0 0 0 0 0 	ce 0.371 0.091 ce	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.542 9.542 0.018 14 La: m_waiting_t 10.447	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3 d_waiting_t 6.742	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 unboarded 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 f_b_time 50.450 50.950	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488 0.007 m_b_time 25.217 25.569	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217 d_b_time 9.417 9.946	m_distant 40.594 40.420 39.851 40.420 40.571 40 571 40 0 0 0 0 39.801 39.764	Ce 0.371 0.091 Ce	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3 d_waiting_t 6.742 6.783	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172	d_decisions 8.947 8.527 7.408 8.710 8.618 8.618 0.359 d_decisions 3.868 4.165	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000 unboarded 0.000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 f_b_time 50.450 50.950 52.300	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488 0.007 m_b_time 25.217 25.569 25.633	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217 d_b_time 9.417 9.946 9.921	m_distand 40.594 40.420 39.851 40.420 40.571 40.571 40.571 40.571 39.801 39.801 39.764 40.182	Ce 0.371 0.091 Ce	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3 d_waiting_t 6.742 6.742 6.783 6.875	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.6670 6.004	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 unboarded 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 f_b_time 50.450 50.950 52.300 50.900	m_b_time 25.575 25.516 25.382 25.51 25.416 25.488 0.007 m_b_time 25.217 25.569 25.569 25.633 25.315	d_b_time 10.178 9.962 9.345 9.287 9.125 9.583 0.217 d_b_time 9.946 9.921 9.921 9.474	m_distanc 40.594 40.420 39.851 40.420 40.571 40.0571 40.0571 40.0571 40.0571 40.0571 39.764 40.182 39.277	ce 0.371 0.091	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018 14 La: m_waiting_t 10.347 10.340 10.601	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.783 6.875 6.537	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 unboarded 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480 f_b_time 50.450 50.950 52.300 50.900 48.850	m_b_time 25.575 25.516 25.382 25.551 25.416 25.418 0.007 m_b_time 25.217 25.569 25.633 25.633 25.315 24.800	d_b_time 10.178 9.982 9.345 9.287 9.125 9.125 9.583 0.217 d_b_time 9.417 9.946 9.921 9.474 9.078	m_distanc 40.594 40.594 40.591 40.420 40.571 40.0571 40.0571 39.764 40.182 39.277 38.785	cce 0.371 0.091	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.601 10.622	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028	d_decisions 8.947 8.527 7.408 8.710 8.618 8.618 d_decisions 3.868 4.165 4.033 3.794 4.161	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.605	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 51.250 51.250 54.800 5.480 50.450 50.950 52.300 50.900 48.850 50.900	m_b_time 25.575 25.516 25.382 25.551 25.416 25.418 0.007 m_b_time 25.217 25.569 25.633 25.315 24.800 25.307	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217 d_b_time 9.417 9.946 9.921 9.474 9.078 9.078	m_distand 40.594 40.420 39.851 40.420 40.571 40.571 40.571 39.801 39.801 39.764 40.182 39.277 38.785 38	cce	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.622 10.532	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 50.450 50.950 50.950 50.900 48.850 50.690 1.537	m_b_time 25.575 25.516 25.382 25.416 25.416 25.418 0.007 m_b_time 25.217 25.569 25.633 25.315 24.800 25.307 0.110	d_b_time 10.178 9.962 9.345 9.287 9.125 9.583 0.217 0.547 9.946 9.921 9.474 9.078 9.078 9.567 0.135	m_distand 40.594 40.420 39.851 40.420 40.571 40.571 40.571 39.801 39.764 40.182 39.277 38.785 38.785 39.277	cce	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432	13 La: m_waiting_t 9.437 9.259 9.614 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.522 10.532 0.024	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114	unboarded 0.000
Lb_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480 50.950 52.300 50.900 48.850 50.690 1.537	m_b_time 25.575 25.516 25.382 25.416 25.416 25.438 0.007 m_b_time 25.217 25.669 25.633 25.315 24.800 25.307 0.110	d_b_time 10.178 9.982 9.345 9.287 9.125 9.125 0.217 d_b_time 9.417 9.946 9.921 9.474 9.078 9.567 0.135	m_distand 40.594 40.420 39.851 40.420 0.571 40.571 40.571 39.764 40.182 39.277 38.785 39.277	CCE D.371 D.091 CCE 9.562 9.562 D.292	d_distance 15.922 15.506 14.772 14.318 14.579 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.340 10.340 10.360 10.522 10.532 0.024	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.783 6.875 6.537 6.466 6.681 0.030	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028	d_door_distrib. 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 f_b_time 50.450 50.950 52.300 48.850 50.900 48.850 50.690 1.537	m_b_time 25.575 25.516 25.382 25.416 25.416 25.418 0.007 m_b_time 25.217 25.569 25.633 25.315 24.800 25.307 0.110	d_b_time 10.178 9.982 9.345 9.287 9.125 9.125 9.125 0.217 d_b_time 9.417 9.946 9.921 9.474 9.078 9.474 9.078 9.475 0.135	m_distant 40.594 40.420 39.851 40.420 40.571 40.0571 40.0571 40.0571 39.801 39.801 39.764 40.182 39.277 38.785 39.277 38.785	CCE D.371 D.091 CCE 9.562 9.562 D.292	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.0409 0.432	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.601 10.522 10.532 0.024 15 La:	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.466 6.681 0.030 cy 0.4	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.645 5.067 5.589 0.114	unboarded 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 5.480 50.450 50.450 50.900 52.300 50.900 1.537 f_b_time	m_b_time 25.575 25.516 25.382 25.551 25.416 25.418 0.007 m_b_time 25.217 25.569 25.633 25.315 24.800 25.307 0.110 m_b_time	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217 d_b_time 9.417 9.946 9.921 9.474 9.078 9.078 9.078 9.078 9.078	m_distand 40.594 40.420 39.851 40.420 40.571 40.571 40.182 39.801 39.764 40.182 39.277 38.785 38 (C) m_distand	CCE 0.371 0.091 CCE 9.562 0.292 CCE	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.540 10.522 0.024 15 La: m_waiting_t	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030 cy 0.4 d_waiting_t	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib.	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000 0.00000 0.0000 0.0000 0.0000 0.00000 0.00000 0.0
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 50.450 50.950 50.950 50.900 48.850 50.900 48.850 50.690 1.537 f_b_time 74.300	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488 0.007 m_b_time 25.217 25.569 25.633 25.315 24.800 25.315 24.800 25.307 0.110 m_b_time 26.329	d_b_time 10.178 9.962 9.345 9.287 9.125 9.583 0.217 0.217 9.946 9.921 9.417 9.946 9.921 9.474 9.078 9.567 0.135 d_b_time 11.769	m_distand 40.594 40.420 39.851 40.420 40.571 40.571 40.671 39.801 39.764 40.182 39.277 38.785 38.05 39.277 38.785 39.277 38.785 39.277 38.785 39.277 38.785 39.277 38.785 39.277 39.275 40.275 400 400000000000000000000000000000000	CCE 0.371 0.091 CCE 0.292 CCE	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance 17.652	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.522 0.024 15 La: m_waiting_t 10.963	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030 cy 0.4 6.654	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000 0.0
t_b_time 51.900 48.200 47.400 51.250 49.090 5.480 5 0.950 52.300 50.950 52.300 50.900 48.850 50.690 1.537 f_b_time 74.300 71.700	m_b_time 25.575 25.516 25.551 25.416 25.416 25.438 0.007 m_b_time 25.217 25.639 25.315 24.800 25.315 24.800 25.307 0.110 m_b_time 26.329 26.316	d_b_time 10.178 9.982 9.345 9.287 9.125 9.125 0.217 d_b_time 9.417 9.946 9.921 9.474 9.078 9.567 0.135 d_b_time 11.769 11.383	m_distand 40.594 40.420 39.851 40.420 40.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 41 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50 0.575 50505050505050505050	CCe 0.371 0.091 CCe 9.562 0.292 CCe	d_distance 15.922 15.506 14.772 14.318 14.579 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.042 0.432 d_distance 17.652 16.669	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.340 10.522 10.532 0.024 15 La: m_waiting_t 10.835 10.835	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.783 6.875 6.537 6.466 6.681 0.030 y 0.4 d_waiting_t 6.654 7.126	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.689 0.114 d_door_distrib. 6.176 6.344	unboarded 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480 50.450 50.450 50.900 48.850 50.900 48.850 50.900 1.537 f_b_time 74.300 71.700 57.950	m_b_time 25.575 25.516 25.382 25.551 25.416 25.418 0.007 m_b_time 25.217 25.663 25.633 25.315 24.800 25.315 24.800 0.110 m_b_time 26.329 26.316 25.978	d_b_time 10.178 9.982 9.345 9.287 9.125 9.125 9.125 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.125 9.287 9.583 9.567 0.135 d_b_time 9.135 d_b_time 9.1769 11.769 11.769	m_distand 40.594 40.420 39.851 40.420 40.571 40.420 (C m_distand 39.801 39.764 40.182 39.277 38.785 38.785 35 (C m_distand 40.102 41.028 41.072 40.475	CCe 0.371 0.091 CCe 9.562 0.292 CCe	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance 17.652 16.669 15.526	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.522 0.024 15 La: m_waiting_t 10.835 11.283	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3 d_waiting_t 6.742 6.783 6.875 6.466 6.681 0.030 cy 0.4 d_waiting_t 6.654 7.126 6.839	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 4.532	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827 2.930	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 50.450 50.950 50.950 52.300 50.900 48.850 50.900 48.850 50.900 1.537 1.537	m_b_time 25.575 25.516 25.382 25.551 25.416 25.418 0.007 m_b_time 25.217 25.569 25.633 25.315 24.800 25.307 0.110 m_b_time 26.329 26.316 25.978 25.697	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217 0.417 9.946 9.921 9.474 9.947 0.788 0.785 0.135 d_b_time 11.769 11.383 10.746	m_distand 40.594 40.420 39.851 40.420 40.571 40.420 (C) m_distand 39.801 39.764 40.182 39.277 38.785 38.785 38.785 38.785 40.172 40.475 40.070	cce 0.371 0.091 cce 9.562 0.292 cce	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance 17.652 16.669 15.526 14.644	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.547 10.547 10.542 10.532 10.652 10.532 10.532 10.535 11.835 11.283 11.283	y 0.2 d_waiting_t 7.236 6.274 7.057 6.676 6.871 0.155 ey 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030 ey 0.4 d_waiting_t 6.654 7.126 6.839 7.253	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 4.532 5.216	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827 2.930 3.415	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000 0.0000 0.0000 0.0000 0.00000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 50.950 50.950 50.950 52.300 50.900 48.850 50.690 1.537 f_b_time 74.300 71.700 57.950 56.100 74.300	m_b_time 25.575 25.516 25.382 25.551 25.416 25.488 0.007 m_b_time 25.217 25.569 25.633 25.315 24.800 25.307 0.110 m_b_time 26.329 26.316 25.978 25.978 25.607 26.143	d_b_time 10.178 9.982 9.287 9.287 9.125 9.287 9.125 9.417 9.946 9.921 9.474 9.078 9.567 0.135 d_b_time 11.769 11.383 10.776	m_distand 40.594 40.420 39.851 40.420 40.571 40.071 39.801 39.764 40.182 39.277 38.785 38.275 38.785 39.277 38.785 40.122 40.475 40.072	cce 0.371 0.091 cce 0.562 0.292 cce	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance 17.652 16.669 15.526 14.644 16.222	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.452 0.024 10.532 0.024 10.532 0.024 10.532 10.535 10.555	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030 y 0.4 d_waiting_t 6.654 7.126 6.839 7.253 7.194	m_decisions 11.176 9.700 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 4.532 5.216 5.032	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827 2.930 3.415 3.260	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000
L b_time 51.900 48.200 47.400 46.700 51.250 49.090 51.250 50.950 52.300 50.950 52.300 50.950 52.300 50.950 50.900 48.850 50.900 1.537 f _b_time 74.300 71.700 57.950 56.100 74.300	m_b_time 25.575 25.516 25.551 25.416 25.416 25.438 0.007 m_b_time 25.217 25.633 25.315 24.800 25.337 0.110 m_b_time 26.329 26.316 25.978 25.697 26.143 20.227	d_b_time 10.178 9.982 9.345 9.287 9.125 9.125 0.217 d_b_time 9.417 9.946 9.921 9.474 9.078 9.567 0.135 d_b_time 11.769 11.383 10.746 10.153 10.746	m_distand 40.594 40.420 39.851 40.420 40.571 40.571 40.0571 40.0571 39.764 40.182 39.277 38.785 38.785 39.277 40.182 41.022 40.475 40.070 40.842	ce 0.371 0.091 ce 9.562 0.292 ce	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.049 0.432 d_distance 17.652 16.669 15.526 14.644 16.222	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018 14 La: m_waiting_t 10.340 10.340 10.601 10.522 10.532 0.024 15 La: m_waiting_t 10.835 10.835 11.283 11.385 10.900	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.466 6.537 6.466 6.681 0.030 y 0.4 d_waiting_t 6.537 6.466 6.654 7.126 6.839 7.253 7.194	m_decisions 11.176 9.700 9.156 10.492 10.240 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 4.532 5.216 5.032	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.978 2.827 2.930 3.415 3.260	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 f_b_time 50.450 50.950 52.300 48.850 50.900 48.850 50.900 1.537 74.300 71.700 57.950 56.100 74.300 66.870 66.870	m_b_time 25.575 25.516 25.551 25.416 25.418 0.007 m_b_time 25.217 25.669 25.633 25.315 24.800 25.315 24.800 25.315 24.800 25.317 0.110 m_b_time 26.329 26.316 25.978 25.607 26.143 26.075	d_b_time 10.178 9.982 9.287 9.287 9.125 9.125 9.125 9.125 9.125 9.125 9.287 9.127 9.287 9.287 9.287 0.217	m_distand 40.594 40.420 39.851 40.420 40.571 40.420 70 70 70 70 70 70 70 70 70 70 70 70 70	cce 	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance 17.652 16.669 15.526 14.644 16.222 16.143	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.601 10.601 10.522 0.024 15 La: m_waiting_t 10.835 10.835 11.283 11.283 11.285 10.910 11.075	cy 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 cy 0.3 d_waiting_t 6.742 6.783 6.783 6.537 6.466 6.681 0.030 cy 0.4 d_waiting_t 6.854 7.126 6.839 7.253 7.194 7.013	m_decisions 11.176 9.700 9.156 10.492 10.240 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 5.216 5.032 4.831	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827 2.930 3.415 3.260 3.082	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 50.450 50.950 50.950 50.950 50.900 48.850 50.900 48.850 50.900 1.537 74.300 56.100 74.300 66.870 82.325	m_b_time 25.575 25.516 25.382 25.551 25.416 25.418 0.007 m_b_time 25.217 25.569 25.315 25.315 24.800 25.315 24.800 0.110 m_b_time 26.329 26.316 25.978 25.597 26.143 26.075 0.089	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217 9.543 9.946 9.921 9.474 9.946 9.921 9.474 9.978 9.078 9.078 9.078 0.135 d_b_time 11.769 11.383 10.746 10.153 10.776 0.391	m_distand 40.594 40.420 39.851 40.420 40.571 40.420 (0) m_distand 39.801 39.764 40.182 39.275 38.785 38.785 38.785 0 0 0 0 0 40.475 40.070 40.842 40.070 40.842 40.070 40.842	CCe 0.0.371 0.091 CCe 0.097 CCe 0.0697 0.178	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance 17.652 16.669 15.526 14.644 16.222 16.143 1.297	13 La: m_witing_t 9.440 9.437 9.259 9.614 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.621 10.522 10.532 0.024 15 La: m_waiting_t 10.963 11.283 11.285 11.385 1	y 0.2 d_waiting_t 7.236 6.274 7.057 6.676 6.871 0.155 ey 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030 ey 0.4 d_waiting_t 6.654 7.126 6.839 7.253 7.194 7.013 0.066	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 4.532 5.216 5.032 4.831 0.081	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827 2.930 3.415 3.260 3.082 0.060	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038	unboarded 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480 50.450 50.950 52.300 52.300 50.900 48.850 50.690 1.537 f_b_time 74.300 71.700 56.600 74.300 74.300 74.300 74.300 74.300 74.300 74.300 74.300	m_b_time 25.575 25.516 25.382 25.51 25.416 25.488 0.007 m_b_time 25.217 25.633 25.315 24.800 25.315 24.800 25.315 24.800 0.110 m_b_time 26.329 26.329 26.316 25.978 25.978 25.607 26.143 26.075 0.089	d_b_time 10.178 9.982 9.287 9.125 9.125 9.287 9.125 9.247 9.417 9.946 9.921 9.474 9.078 9.567 0.135 d_b_time 11.769 11.383 10.746 10.776 10.965 0.391	m_distand 40.594 40.420 39.851 40.420 0.571 40 39.764 40.182 39.277 38.785 38 6 7 1 1 1 1 1 1 1 1 1 1	CCE 0.371 0.0091 CCE 0.562 0.292 CCE 0.697 0.178	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance 17.652 16.669 15.526 14.644 16.222 16.143 1.297	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.452 10.452 10.532 0.024 10.532 10.532 10.532 10.935 11.283 11.385 10.910 11.075 0.059 16 L =	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030 y 0.4 4.waiting_t 6.654 7.126 6.839 7.253 7.194 7.013 0.066 y 0.5	m_decisions 11.176 9.700 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 4.532 5.216 5.032 4.831 0.081	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827 2.930 3.415 3.260 3.082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082 0.3082	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000 0.00000 0.00000 0.000000 0.00000000
Lb_time 51.900 48.200 47.400 46.700 51.250 49.090 51.250 50.950 52.300 50.950 52.300 50.900 48.850 50.950 52.300 50.950 52.300 50.950 52.300 50.950 50.950 51.950 56.100 74.300 66.870 82.325 f b time	m_b_time 25.575 25.516 25.581 25.416 25.416 25.418 0.007 m_b_time 25.217 25.669 25.633 25.315 24.800 25.337 0.110 m_b_time 26.329 26.316 25.978 25.607 26.143 26.075 0.089 m_b_time	d_b_time 10.178 9.982 9.345 9.287 9.125 9.125 9.287 9.125 9.287 9.125 9.287 9.247 9.947 9.947 9.947 9.947 9.947 9.947 9.474 9.078 9.567 0.135 0.135 0.176 10.365 0.391 d_b_time	m_distand 40.594 40.420 39.851 40.420 40.571 40.0571 40.0571 39.764 40.182 39.277 38.785 39.277 38.785 39.277 38.785 39.277 40.070 41.028 41.072 40.475 40.070 40.842 40.070 40.842 40.070	CCe 0.371 0.091 CCe 0.292 CCe 0.697 0.178	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.049 0.432 d_distance 17.652 16.669 15.526 14.644 16.222 16.143 1.297 d_distance	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.601 10.601 10.632 10.633 10.835 10.835 10.835 10.835 10.835 10.910 11.075 0.058 11.075	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.466 6.466 6.681 0.030 ry 0.4 d_waiting_t 6.654 7.126 6.839 7.253 7.194 7.013 0.066 ry 0.5	m_decisions 11.176 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 4.532 5.216 5.032 4.831 0.081 m_decisions	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.937 3.415 3.260 3.082 0.060	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038 d_door_distrib.	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 5.480 5.0450 50.450 50.900 48.850 50.900 48.850 50.900 48.850 50.900 1.537 7.1.700 57.950 56.100 74.300 74.300 74.300 74.300 74.300 74.300 74.300 74.300 75.955 56.100 74.300 74.300 75.956 57.950 56.100 74.300 76.870 82.325 76.600 76.600 76.870 82.325 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.600 76.60	m_b_time 25.575 25.516 25.551 25.416 25.418 0.007 m_b_time 25.217 25.569 25.633 25.315 24.800 25.315 24.800 25.315 24.800 25.337 0.110 m_b_time 26.329 26.316 25.978 25.607 26.143 26.075 0.089 m_b_time	d_b_time 10.178 9.982 9.345 9.287 9.125 9.125 9.125 9.125 9.125 9.287 9.127 9.474 9.946 9.921 9.474 9.9474 9.921 9.474 9.978 9.567 0.135 0.135 0.135 10.746 10.746 10.765 0.391 d_b_time 11.529 0.391	m_distand 40.594 40.420 39.851 40.420 40.571 40 39.801 39.764 40.182 39.277 38.785 35 35 36 1 1 1 1 1 1 1 1 1 1	CCe 0.371 0.091 CCe 0.562 0.292 CCe 0.697 0.178 CCe	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance 17.652 16.669 15.526 14.644 16.222 16.143 1.297 d_distance 17.071	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.522 0.024 15 La: m_waiting_t 10.835 11.283 10.910 11.075 0.059 16 La: m_waiting_t	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.466 6.681 0.030 y 0.4 d_waiting_t 6.654 7.126 6.839 7.253 7.194 7.013 0.066 y 0.5 d_waiting_t 7.293 7.194	m_decisions 11.176 9.700 9.156 10.492 10.240 10.240 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 4.532 5.216 5.032 4.831 0.081 m_decisions	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827 2.930 3.415 3.260 3.082 0.060 d_decisions 2.557	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038 d_door_distrib. 6.255	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000
f_b_time 51.900 48.200 47.400 51.250 49.090 5.480 f_b_time 50.450 50.950 52.300 50.950 52.300 50.900 48.850 50.900 48.850 50.900 48.850 50.900 48.850 50.900 1.537 f_b_time 74.300 71.700 56.100 74.300 66.870 82.325 f_b_time 76.400 78.900	m_b_time 25.575 25.516 25.382 25.51 25.416 25.488 0.007 m_b_time 25.217 25.569 25.569 25.569 25.633 25.315 24.800 25.315 24.800 25.307 0.110 m_b_time 26.329 26.316 25.978 25.978 25.607 26.143 26.075 0.089 m_b_time 26.403 26.075 0.089	d_b_time 10.178 9.982 9.345 9.287 9.125 9.583 0.217 d_b_time 9.417 9.946 9.921 9.474 9.078 9.078 9.078 9.078 9.078 11.769 11.383 10.776 10.153 10.776 0.391 d_b_time 11.559 0.391	m_distand 40.594 40.420 39.851 40.420 40.571 40.571 40.571 39.764 40.182 39.275 38.785 38.785 38.785 40.182 40.475 40.070 40.842 40.475 40.077 40.842 40.475 40.070 40.842 40.75	CCe 0.697 0.178 CCe 0.697 0.178	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.644 16.222 16.143 1.297 d_distance 17.071 17.100	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.621 10.532 10.632 10.532 10.532 10.532 10.532 10.532 10.532 10.532 10.532 10.535 11.283 11.385 11.3	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 ey 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030 ey 0.4 d_waiting_t 6.654 7.126 6.839 7.253 7.194 7.013 0.066 ey 0.5 d_waiting_t 7.888 7.055 7.057 7.057 7.057 7.057 7.057 7.057 7.253 7.194 7.013 0.066	m_decisions 11.176 9.700 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.931 0.047 m_decisions 4.744 4.632 4.532 5.032 4.831 0.081 m_decisions 4.144 4.000	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827 2.930 3.415 3.260 3.082 0.060 d_decisions 2.567	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038 d_door_distrib. 6.366 6.961	unboarded 0.0000 0.000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480 50.950 52.300 50.950 52.300 50.900 48.850 50.690 1.537 1.537 1.50 50.950	m_b_time 25.575 25.516 25.551 25.416 25.438 0.007 m_b_time 25.217 25.633 25.315 24.800 25.337 0.110 m_b_time 26.329 26.329 26.329 26.329 26.316 25.978 25.607 26.143 26.075 0.089 m_b_time 26.403 26.403 26.403 26.403 26.675 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.21111 0.211111 0.211111 0.21111 0.21	d_b_time 10.178 9.982 9.287 9.125 9.125 9.287 9.125 9.417 9.946 9.417 9.946 9.417 9.946 9.474 9.078 9.567 0.135 d_b_time 11.769 11.383 10.746 10.153 10.746 10.965 0.391 d_b_time 11.550 11.721 10.961 10.965 11.721 10.965 11.721 10.965 10.961 11.750 11.750 11.751 10.965 11.750 11.751 10.965 11.750 11.750 11.750 11.751 10.965 1	m_distand 40.594 40.420 39.851 40.420 40.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.571 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 41 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.575 40 0.075 40 0.075 40 0.075 40 0.070 40 .475 40 .070 40 .842 40 40 .803 41 .0803 41 .0805 41 .0805 4	CCe 0.371 0.091 CCe 0.562 0.292 CCe 0.697 0.178 CCe	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.042 13.651 14.049 0.432 d_distance 17.652 16.669 15.526 14.644 16.222 16.143 1.297 d_distance 17.071 17.109	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.340 10.340 10.601 10.522 10.632 10.632 10.632 10.633 10.635 10.963 11.283 11.385 10.910 11.075 0.059 11.075 0.059 11.283 11.075 0.059 11.075 11.05	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.466 6.537 6.466 6.654 7.126 6.839 7.253 7.194 7.013 0.066 y 0.5 d_waiting_t 7.253 7.194 7.013 0.066	m_decisions 11.176 9.700 9.156 10.492 10.240 10.240 m_decisions 5.824 6.172 5.820 6.028 5.600 6.028 5.600 6.028 5.601 0.047 m_decisions 4.744 4.632 4.532 5.216 5.032 4.831 0.081 m_decisions 4.144 4.000 0.000	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.930 3.415 3.260 3.082 0.060 d_decisions 2.567 2.500	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.689 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038 d_door_distrib. 6.366 6.001	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 51.250 49.090 5.480 50.950 52.300 48.850 50.900 48.850 74.300 71.700 57.950 56.100 74.300 66.870 82.325 f_b_time 76.400 78.300	m_b_time 25.575 25.516 25.551 25.416 25.416 25.417 25.569 25.633 25.315 24.800 25.315 24.800 25.315 24.800 25.317 25.637 26.316 25.978 25.607 26.143 26.075 0.089 m_b_time 26.403 26.675 26.514	d_b_time 10.178 9.982 9.287 9.287 9.125 9.287 9.125 9.287 9.287 9.287 9.217 0.217 0.217 0.217 0.217 0.217 0.325 0.217 0.217 0.217 0.355 0.301 0.766 10.765 0.391 d_b_time 11.550 11.550 11.721 12.451	m_distand 40.594 40.420 39.851 40.420 40.571 40.571 40.070 39.801 39.764 40.182 39.277 38.785 38.785 39.277 38.785 39.277 38.785 39.277 38.785 40.070 40.475 40.070 40.475 40.070 40.842 41.072 40.070 40.842 40.070 40.842 40.070 40.842 40.070 40.842 40.070 40.842 40.070 40.842 40.070 40.842 40.070 40.842 40.070 40.842 40.070 40.842 40.070 40.842 40.070 40.842 40.070	CCe 0.371 0.091 CCe 0.562 0.292 CCe 0.697 0.178 CCe	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.042 13.651 14.042 16.669 15.526 14.644 16.222 16.143 1.297 d_distance 17.071 17.109 18.0006	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.621 10.621 10.522 0.024 10.522 0.024 10.532 0.024 10.532 10.532 0.024 11.283 11.385 10.910 11.285 11.460 11.460 11.545 11.545 11.545 11.281 11.545 11.281 11	y 0.2 d_waiting_t 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.466 6.466 6.466 6.681 0.030 y 0.4 d_waiting_t 6.839 7.253 7.194 7.013 0.066 y 0.5 d_waiting_t 7.888 7.996 7.764	m_decisions 11.176 9.700 9.156 10.492 10.240 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 5.216 5.032 4.831 0.081 m_decisions 4.144 4.000 3.5588	d_decisions 8,947 8,527 7,408 8,710 8,618 8,442 0,359 d_decisions 3,868 4,165 4,033 3,794 4,161 4,004 0,028 d_decisions 2,978 2,827 2,930 3,415 3,260 0,060 d_decisions 2,567 2,500 2,540	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.667 5.687 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038 d_door_distrib. 6.366 6.601 6.485	unboarded 0.0000 0.0000 0.000 0.000 0.0000 0.000 0.000 0.000 0.000 00
f_b_time 51.900 48.200 47.400 51.250 49.990 5.480 50.450 50.950 52.300 50.900 48.80 50.600 1.537 6.690 1.700 57.950 56.100 74.300 74.300 66.870 82.325 f_b_time 76.400 73.450	m_b_time 25.575 25.516 25.551 25.551 25.416 25.418 0.007 m_b_time 25.217 25.569 25.633 25.315 24.800 25.315 24.800 25.315 24.800 25.317 25.631 25.631 25.607 26.316 25.978 25.607 26.143 26.075 0.089 m_b_time 26.403 26.675 26.514 25.916 25.916	d_b_time 10.178 9.982 9.345 9.287 9.125 9.287 9.125 9.287 9.417 9.946 9.921 9.474 9.9474 9.921 9.474 9.978 9.567 0.135 0.135 0.135 10.746 10.746 10.746 10.746 10.745 0.391 d_b_time 11.550 11.721 12.451 10.417	m_distand 40.594 40.420 39.851 40.420 40.571 40 50 39.801 39.764 40.182 39.277 38 .785 38 .785 38 .785 38 .785 39 .277 40 .475 40.475 40.475 40.475 40.475 40.475 40 .472 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 40 .475 4	CCe 0.371 0.091 CCe 0.562 0.292 CCe 0.697 0.178 CCe	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.409 0.432 d_distance 17.652 16.669 15.526 14.644 16.222 16.143 1.297 d_distance 17.071 17.071 17.09 18.006 16.092 16.92	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.749 10.340 10.522 0.024 15 La: m_waiting_t 11.283 10.910 11.075 0.059 11.283 11.285 10.910 11.075 0.059 16 La: m_waiting_t 11.545 11.555 11.555 11.555 11.555 11.555 11.555 11.555 11.555 11.555 11.55	y 0.2 d_waiting_t 7.114 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030 y 0.4 d_waiting_t 6.654 7.126 6.839 7.253 7.194 7.013 0.066 y 0.5 d_waiting_t 7.888 7.996 7.764 7.216	m_decisions 11.176 9.700 9.156 10.492 10.240 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.600 6.028 m_decisions 4.744 4.632 4.532 5.216 5.032 4.831 0.081 m_decisions 4.144 4.000 3.588 3.940 - 0.01	d_decisions 8,947 8,527 7,408 8,710 8,618 8,442 0,359 d_decisions 3,868 4,165 4,033 3,794 4,161 4,004 0,028 d_decisions 2,978 2,827 2,930 3,415 3,260 0,060 d_decisions 2,567 2,540 2,648 0,025 1,005	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038 d_door_distrib. 6.366 6.601 6.485 6.291	unboarded 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000
f_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480 50.450 50.950 50.950 50.950 50.900 48.850 50.900 48.850 50.900 1.537 1.538 1.540 1.540 1.540 1.5400 1.5.750	m_b_time 25.575 25.516 25.382 25.51 25.416 25.488 0.007 m_b_time 25.217 25.569 25.569 25.315 24.800 25.315 24.800 25.307 0.110 m_b_time 26.329 26.316 25.978 26.975 26.974 26.975 26.974 26.975 26.974 26.975 26.974 26.975 26.974 26.975 26.978 26.975 26.9788 26.9788 26.978	d_b_time 10.178 9.982 9.287 9.125 9.287 9.125 9.247 9.417 9.946 9.417 9.921 9.474 9.078 9.567 0.135 0.766 10.153 10.776 10.776 10.995 0.391 d_b_time 11.550 11.721 12.451 10.417 11.086	m_distand 40.594 40.420 39.851 40.420 40.571 40.420 (0 m_distand 39.801 39.764 40.182 39.277 38.785 38.785 38.785 38.785 40.182 40.475 40.070 40.842 40.475 40.070 (0 7 m_distand 41.028 41.028 41.689 41.609 41.400	CCe 0.697 0.178 CCe 0.697 0.178	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.640 15.526 14.644 16.222 16.143 1.297 d_distance 17.071 17.109 18.006 16.092 16.940	13 La: m_waiting_t 9.440 9.437 9.259 9.614 9.542 9.542 9.437 0.018 14 La: m_waiting_t 10.447 10.447 10.447 10.542 0.024 10.522 0.024 10.521 10.522 0.024 15 La: m_waiting_t 10.963 10.835 11.385 10.910 11.450 11.460 11.455 11.221 11.31 11.057	y 0.2 d_waiting_t 7.236 6.274 7.057 6.676 6.871 0.155 y 0.3 d_waiting_t 6.742 6.783 6.875 6.537 6.466 6.681 0.030 y 0.4 d_waiting_t 6.654 7.126 6.839 7.253 7.194 7.013 0.066 y 0.5 d_waiting_t 7.888 7.996 7.764 7.216 7.481	m_decisions 11.176 9.700 9.700 9.156 10.492 10.240 10.153 0.592 m_decisions 5.824 6.172 5.932 5.600 6.028 5.911 0.047 m_decisions 4.744 4.632 4.532 5.216 5.032 4.831 0.081 m_decisions 4.144 4.000 3.588 3.940 3.964	d_decisions 8.947 8.527 7.408 8.710 8.618 8.618 4.033 3.794 4.165 4.033 3.794 4.161 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.827 2.930 3.415 3.260 3.082 0.060 d_decisions 2.567 2.500 2.543 2.540 2.544 2.540	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 5.324 0.018 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.589 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038 d_door_distrib. 6.366 6.601 6.485 5.291 6.659	unboarded 0.0000 0.0000 0.0000 0.000000
1_b_time 51.900 48.200 47.400 46.700 51.250 49.090 5.480 50.950 52.300 50.950 52.300 50.950 52.300 50.950 51.950 50.690 1.537 71.700 57.950 56.100 74.300 66.870 82.325 75.400 75.750 75.750	m_b_time 25.575 25.516 25.551 25.551 25.416 25.438 0.007 m_b_time 25.217 25.633 25.315 24.800 25.335 24.800 25.337 0.110 m_b_time 26.329 26.329 26.329 26.329 26.329 26.316 25.978 25.607 26.372 26.403 26.075 0.089 m_b_time 26.403 26.403 26.403 26.403 26.514 25.916 26.272 26.326 26.272 26.326	d_b_time 10.178 9.982 9.345 9.287 9.125 9.125 0.217 d_b_time 9.417 9.946 9.921 9.474 9.078 9.567 0.135 d_b_time 11.769 11.383 10.746 10.153 10.746 10.153 10.746 11.383 10.746 11.550 11.721 12.451 10.417 11.086 11.086	m_distand 40.594 40.420 39.851 40.420 40.571 40. 39.764 40.182 39.277 38.785 32 6 7 1 1 1 1 1 1 1 1 1 1	ce 0.371 0.091 ce 9.562 0.292 ce 0.697 0.178 ce	d_distance 15.922 15.506 14.772 14.318 14.579 15.019 0.450 d_distance 14.225 14.808 15.317 14.042 13.651 14.042 13.651 14.049 0.432 d_distance 17.652 14.644 16.222 16.143 1.297 d_distance 17.071 17.109 18.006 16.092 16.940 17.044	13 La: m_waiting_t 9.440 9.447 9.259 9.614 9.542 9.542 9.458 0.018 14 La: m_waiting_t 10.447 10.447 10.447 10.447 10.601 10.522 10.532 0.024 10.532 10.532 10.532 10.633 10.633 10.633 10.633 10.633 10.634 10.525 10.963 11.055 11.075 0.059 11.075 11.025 11.281 11.345 11.057 11.295	y 0.2 d_waiting_t 7.236 6.274 7.057 6.676 6.676 y 0.3 d_waiting_t 6.742 6.783 6.875 6.466 6.537 6.466 6.654 7.126 6.839 7.253 7.194 7.013 0.066 y 0.5 d_waiting_t 7.253 7.194 7.013 0.066 y 0.5 7.764 7.216 7.481 7.669	m_decisions 11.176 9.700 9.156 10.492 10.240 10.240 m_decisions 5.824 6.172 5.932 5.600 6.028 5.931 0.047 m_decisions 4.744 4.632 4.532 5.216 5.032 4.831 0.081 m_decisions 4.144 4.000 3.588 3.940 3.964 3.964 3.927	d_decisions 8.947 8.527 7.408 8.710 8.618 8.442 0.359 d_decisions 3.868 4.165 4.033 3.794 4.161 4.004 0.028 d_decisions 2.978 2.978 2.930 3.415 3.260 3.082 0.060 d_decisions 2.567 2.500 2.540 2.667 2.584	d_door_distrib. 5.491 5.350 5.197 5.403 5.178 d_door_distrib. 5.560 5.670 6.004 5.645 5.067 5.689 0.114 d_door_distrib. 6.176 6.344 6.404 5.949 6.404 6.255 0.038 d_door_distrib. 6.366 6.601 6.485 6.291 6.659 6.480	unboarded 0.0000 0.0000 0.0000 0.000000

					17 Laz	y 0.6				
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
80.950	27.492	13.568	42.712	19.859	11.856	8.485	3.848	2.376	7.082	0.000
75.650	26.377	11.105	41.130	16.096	11.391	8.034	3.328	2.148	6.849	0.000
78.650	26.584	11.634	41.571	17.330	11.515	7.155	2.944	1.909	6.744	0.000
75.450	26.383	11.651	41.231	17.617	11.432	8.114	3.112	2.091	6.716	0.000
78.600	27.221	12.742	42.578	19.020	11.467	7.924	3.232	2.116	7.235	0.000
77.860	26.811	12.140	41.844	17.984	11.532	7.942	3.293	2.128	6.925	0.000
5.353	0.264	0.992	0.563	2.179	0.035	0.238	0.117	0.028	0.051	0.000
					18 97	w 0 7				
f h time	m h time	d h time	m distance	d distance	m waiting t	d waiting t	m decisions	d decisions	d door distrib	unboarded
87 500	29 549	17 019	46 921	26 491	12 275	9 477	2 468	1 434	7 895	0.000
81 850	27 953	14 702	43 668	21 437	11 947	8 930	2 408	1.435	7 122	0.000
87.950	29.511	17.278	46.693	26.205	12.286	9.014	2.488	1.435	7.506	0.000
61.350	26.496	10.929	41.420	16.243	11.280	7.344	2.520	1.506	6.898	0.000
83.500	27.479	13.323	42.785	19.958	11.910	8.487	2.692	1.582	7.208	0.000
80.430	28,198	14.650	44,297	22.067	11.940	8,650	2,515	1.478	7.326	0.000
120.506	1.756	7.032	5.896	18,865	0.167	0.657	0.011	0.004	0.149	0.000
120.000	1.750	1.002	5.050	10.000	0.107	0.007	0.011	0.004	0.145	0.000
					19 Laz	y 0.8				
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
81.100	27.616	13.139	43.496	19.736	12.037	8.022	2.288	1.118	7.088	0.000
89.850	29.661	17.734	47.100	27.322	12.357	9.638	2.169	1.173	7.925	1.000
77.900	26.980	12.399	42.298	18.178	11.359	7.763	2.356	1.340	7.261	0.000
00.000	20.442	17 140	44.415	24.001	12.030	9.091	2.344	1.213	7.009	2,000
88.800	29.248	17.148	45.995	25.940	12.412	10.054	2.339	1.343	7.007	2.000
84.860	28.389	15.270	44.661	23.051	12.044	8.913	2.299	1.237	7.460	0.600
26.552	1.235	5.704	3.682	15.596	0.176	0.994	0.006	0.010	0.112	0.800
					20 Laz	y 0.9				
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
88.900	29.239	16.391	45.880	24.778	12.265	9.251	2.040	0.928	7.834	1.000
88.600	27.965	14.899	43.445	21.971	12.222	9.106	1.924	0.854	7.506	0.000
87.450	28.720	15.693	45.306	25.045	11.946	9.467	2.128	1.033	7.525	0.000
89.350	29.334	17.309	46.964	26.601	12.444	9.426	2.061	0.953	7.982	2.000
88.800	29.699	17.064	46.673	26.334	12.883	9.942	2.089	0.998	8.328	2.000
88.620	28.991	16.271	45.654	24.946	12.352	9.438	2.048	0.953	7.835	1.000
0.503	0.452	0.984	1.952	3.389	0.120	0.100	0.006	0.005	0.117	1.000
					21 07	w 1 0				
f h timo	m h timo	d h timo	m distanco	d distanco	m waiting t	d waiting t	m docisions	d docisions	d door distrib	unboardod
1_D_0110	20.472	14 044	16 497	0_013tance	12.566	0_0271	1 744			
88 800	29.368	15.016	46 444	22.618	12.300	9.438	1.744	0.736	9.378	1,000
89 100	29 761	17 506	47 078	26 687	12.570	9 971	1.746	0.807	9 160	6.000
88.650	29.210	15.438	45.801	23.053	12.489	10.115	1.699	0.703	9.078	1.000
89.500	29.674	16.461	47.284	25.336	12.402	9.844	1.756	0.736	9.542	8.000
89,100	29,497	15.873	46.619	24.081	12,484	9.728	1,733	0.738	9,249	3.200
0.144	0.050	1.200	0.343	3.359	0.009	0.129	0.001	0.002	0.041	12,700
•••••	0.000		0.0.10	0.000		0.1.20	0.001	0.002	0.011	
				22 De	cision Step Free	uency 1 per se	cond			
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
77.250	∠b.993	13.002	42.150	19.024	11.557	8.0/2 7.971	3.804	2.428	0./23	0.000
70.050	21.122	12.009	42.338	19.304	11./99	1.0/1	4.000	2.5/1	0.021	0.000
76.700	20.00/	11.240	41.857	16.400	11.432	0.039	0.932 2.076	2.010	0.404	0.000
74 200	26.349	11 230	40 944	16.671	11.818	7 929	3 984	2.050	6.558	0.000
76 040	06 600	10.050	44 600	10 040	44 650	7 070	2.001	0 570	E 604	0.000
76.210	20.083	12.058	41.099	18.010	11.003	7.870	3.939	2.5/2	0.001	0.000
1.369	0.125	0.010	0.376	1./56	0.075	0.059	0.006	0.009	0.029	0.000
				23 Deci	ision Step Freat	iency 100 per s	econd			
f_b time	m_b time	d_b time	m_distance	d_distance	m_waiting t	d_waiting t	m_decisions	d_decisions	d_door distrib	unboarded
78.350	26.704	12.246	41.584	17.869	11.447	7.742	4.080	2.750	6.433	0.000
60.150	25.992	10.378	40.397	14.872	11.121	7.416	4.508	3.040	6.558	0.000
76.450		11.930	41.137	17.345	11.816	7.897	4.100	2.790	6.396	0.000
75.950	26.556		10.010	16.471	10.824	6.843	3.976	2.683	6.793	0.000
78 150	26.556 26.074	11.151	40.819			0 100	0.000	2 466	C 000	0.000
70.150	26.556 26.074 26.824	11.151 13.421	40.819 41.784	19.215	11.238	8.182	3.992	2.400	0.898	
73.810	26.556 26.074 26.824 26.430	11.151 13.421 11.825	40.819 41.784 41.144	19.215 17.154	11.238 11.289	8.182 7.616	3.992 4.131	2.400	6.616	0.000
73.810 59.398	26.556 26.074 26.824 26.430 0.141	11.151 13.421 11.825 1.321	40.819 41.784 41.144 0.317	19.215 17.154 2.617	11.238 11.289 0.137	8.182 7.616 0.263	3.992 4.131 0.047	2.400 2.746 0.043	6.616 0.049	0.000
73.810 59.398	26.556 26.074 26.824 26.430 0.141	11.151 13.421 11.825 1.321	40.819 41.784 41.144 0.317	19.215 17.154 2.617	11.238 11.289 0.137	8.182 7.616 0.263	4.131 0.047	2.746 0.043	6.616 0.049	0.000
73.810 59.398	26.556 26.074 26.824 26.430 0.141	11.151 13.421 11.825 1.321	40.819 41.784 41.144 0.317	19.215 17.154 2.617	11.238 11.289 0.137 24 one decisio	8.182 7.616 0.263 on per agent	4.131 0.047	2.746 0.043	6.616 0.049	0.000
73.810 59.398	26.556 26.074 26.824 26.430 0.141 m_b_time	11.151 13.421 11.825 1.321 d_b_time	40.819 41.784 41.144 0.317 m_distance	19.215 17.154 2.617	11.238 11.289 0.137 24 one decision m_waiting_t	7.616 0.263 0 per agent d_waiting_t	4.131 0.047 m_decisions	2.746 2.746 0.043	6.698 6.616 0.049 d_door_distrib.	0.000 0.000 unboarded
73.810 59.398 f_b_time 82.000	26.556 26.074 26.824 26.430 0.141 m_b_time 30.509 20.702	11.151 13.421 11.825 1.321 d_b_time 17.404	40.819 41.784 41.144 0.317 m_distance 49.319 40.522	19.215 17.154 2.617 d_distance 26.529 27.95	11.238 11.289 0.137 24 one decision m_waiting_t 9.931 10.132	8.182 7.616 0.263 on per agent d_waiting_t 8.841 0.679	4.131 0.047 m_decisions 1.172	2.746 2.746 0.043 d_decisions 0.565	6.698 6.616 0.049 d_door_distrib. 8.116	0.000 0.000 unboarded 0.000
73.810 59.398 f_b_time 82.000 88.600 82.250	26.556 26.074 26.824 26.430 0.141 m_b_time 30.509 30.708 20.145	11.151 13.421 11.825 1.321 d_b_time 17.404 18.028	40.819 41.784 41.144 0.317 m_distance 49.319 49.539	19.215 17.154 2.617 d_distance 26.529 27.956 26.429	11.238 11.289 0.137 24 one decision m_waiting_t 9.931 10.136 10.265	8.182 7.616 0.263 on per agent d_waiting_t 8.841 9.678 9.017	3.392 4.131 0.047 m_decisions 1.172 1.218 1.152	2.746 2.746 0.043 d_decisions 0.565 0.662 0.525	6.698 6.616 0.049 d_door_distrib. 8.116 8.328 9.217	0.000 0.000 0.000 2.000 1.000
73.810 59.398 f_b_time 82.000 88.600 82.250 83.450	26.556 26.074 26.824 26.430 0.141 m_b_time 30.509 30.708 30.145 30.150	11.151 13.421 11.825 1.321 d_b_time 17.404 18.028 16.889 17.300	40.819 41.784 41.144 0.317 m_distance 49.319 49.539 49.022 48.634	19.215 17.154 2.617 d_distance 26.529 27.956 26.438 26.009	11.238 11.289 0.137 24 one decisie m_waiting_t 9.931 10.136 10.365 9.787	7.616 0.263 on per agent d_waiting_t 8.841 9.678 9.017 8.923	4.131 0.047 m_decisions 1.172 1.218 1.157 1.161	2.746 2.746 0.043 d_decisions 0.565 0.662 0.535 0.545	6.698 6.616 0.049 d_door_distrib. 8.116 8.328 8.317 8.079	0.000 0.000 unboarded 0.000 2.000 1.000 1.000
73.810 59.398 f_b_time 82.000 88.600 82.250 83.450 81.000	26.556 26.074 26.824 26.430 0.141 m_b_time 30.509 30.708 30.145 30.150 29.921	11.151 13.421 11.825 1.321 d_b_time 17.404 18.028 16.889 17.390 16.440	40.819 41.784 41.144 0.317 m_distance 49.319 49.539 49.022 48.634 47.882	19.215 17.154 2.617 d_distance 26.529 27.956 26.438 26.009 24.340	11.238 11.289 0.137 24 one decision m_waiting_t 9.931 10.136 10.365 9.787 10.328	7.616 0.263 on per agent d_waiting_t 8.841 9.678 9.017 8.923 9.200	4.131 0.047 m_decisions 1.172 1.218 1.157 1.161 1.173	2.746 0.043 d_decisions 0.565 0.662 0.535 0.545 0.602	6.698 6.616 0.049 d_door_distrib. 8.116 8.328 8.317 8.079 8.166	0.000 0.000 0.000 2.000 1.000 1.000 2.000
73.810 59.398 f_b_time 82.000 88.600 82.250 83.450 81.000	26.556 26.074 26.824 26.430 0.141 m_b_time 30.509 30.708 30.145 30.150 29.921	11.151 13.421 11.825 1.321 d_b_time 17.404 18.028 16.889 17.390 16.440	40.819 41.784 41.144 0.317 m_distance 49.319 49.539 49.022 48.634 47.882	19.215 17.154 2.617 d_distance 26.529 27.956 26.438 26.009 24.340	11.238 11.289 0.137 24 one decision m_waiting_t 9.931 10.136 10.365 9.787 10.328 40.422	7.616 0.263 on per agent d_waiting_t 8.841 9.678 9.017 8.923 9.200	4.131 0.047 m_decisions 1.172 1.218 1.157 1.161 1.173	2.746 2.746 0.043 d_decisions 0.565 0.662 0.535 0.602 0.545 0.602	6.616 0.049 d_door_distrib. 8.116 8.328 8.317 8.079 8.166	0.000 0.000 0.000 2.000 1.000 2.000 1.000 2.000
73.810 59.398 f_b_time 82.000 88.600 82.250 83.450 81.000 83.460	26.556 26.074 26.824 26.430 0.141 m_b_time 30.509 30.708 30.145 30.150 29.921 30.287 0.400	11.151 13.421 11.825 1.321 d_b_time 17.404 18.028 16.889 17.390 16.440 17.230	40.819 41.784 41.144 0.317 m_distance 49.319 49.539 49.022 48.634 47.882 48.879	19.215 17.154 2.617 d_distance 26.529 27.956 26.438 26.009 24.340 26.254 1000	11.238 11.289 0.137 24 one decisie m_waiting_t 9.931 10.136 10.365 9.787 10.328 10.109 0.009	8.182 7.616 0.263 on per agent d_waiting_t 8.841 9.678 9.017 8.923 9.200 9.132	4.131 0.047 m_decisions 1.172 1.218 1.157 1.161 1.173 1.176	2.400 2.746 0.043 d_decisions 0.565 0.662 0.535 0.545 0.602 0.582	6.698 6.616 0.049 d_door_distrib. 8.116 8.328 8.317 8.079 8.166 8.201	0.000 0.000 2.000 1.000 2.000 1.000 2.000 1.200 0.200

					25 ten decisio	ns per agent				
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
76.000	26.302	11.981	40.988	17.318	11.353	7.481	3.712	2.284	6.630	0.000
77.850	26.145	11.162	40.874	16.564	11.147	7.322	4.140	2.513	6.507	0.000
76.700	26.725	12.544	41.817	19.144	11.714	8.408	4.116	2.582	6.644	0.000
76.350	26.282	12.025	41.074	18.014	11.499	7.422	3.680	2.248	6.521	0.000
75.400	26.037	11.076	40.480	16.226	11.370	7.588	3.696	2.412	6.536	0.000
76.460	26.298	11.758	41.047	17.453	11.417	7.644	3.869	2.408	6.568	0.000
0.834	0.069	0.390	0.237	1.372	0.044	0.192	0.056	0.021	0.004	0.000
					06					
the stress	and the Alaman	d to diverse	and the transmission	d distances	26 no pa	tience	and the state of a	at at a state way	al also an all shells	contra a surfaced
T_D_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
76.950	27.250	13.530	42.890	20.035	11.532	8.300	3.790	2.437	0.080	0.000
74.900	25.769	11.279	40.642	10.703	10.942	7.985	3.588	2.471	0.411	0.000
76.400	20.188	10.050	40.953	17.790	10.962	7.604	3.580	2.290	0.000	0.000
72.900	25.918	10.850	40.418	15.980	11.138	7.490	3.524	2.415	6.433	0.000
71.150	23.000	10.230	40.070	13.275	11.009	0.073	3.970	2.000	0.233	0.000
74.460	26.185	11.5/8	40.972	17.491	11.183	7.661	3.693	2.455	6.497	0.000
5.887	0.382	1.555	1.290	3.915	0.064	0.310	0.036	0.015	0.028	0.000
					07 notion	aa hiah				
f h time	m h timo	d h time	m distanco	d distanco	∠/ patien	d waiting t	m decisions	d decisions	d door distrib	unboardod
73 200	26 17/	11 502	10 974	17 778	10 986	7 700	3 712	2 718	6 350	0.000
78 000	27 267	12 03/	42 362	19.080	11 236	7 863	3.956	2.636	6.814	0.000
77 200	26 652	11 862	72.002	17 300	11.200	7 327	4 256	2.000	6 694	0.000
77.300	20.002	12.050	41.722	10.027	10.062	1.321 7.975	4.200	2.112	6 400	0.000
10.450	20.981	10.059	42.3/3	19.93/	10.903	7.070	3.940	2./10	0.499	0.000
ob.900	25.947	10.490	40.072	15.333	11.413	1.276	3.848	2.430	0.389	0.000
72.170	26.604	11.970	41.541	17.904	11.209	7.608	3.942	2.654	6.551	0.000
76.325	0.300	1.134	1.063	3.102	0.053	0.083	0.040	0.018	0.039	0.000
					28 veloci	ty 1m/s				
t_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
88.850	28.580	12.465	26.788	11.782	9.188	6.985	3.073	1.999	6.977	2.000
89.550	28.592	13.394	26.822	12.582	9.186	6.954	2.886	1.672	7.187	4.000
58.100	26.587	10.152	24.804	9.741	9.025	6.243	3.036	1.848	5.541	1.000
84.850	28.027	13.030	26.273	12.352	9.055	6.596	2.884	1.623	6.715	1.000
90.000	28.606	13.261	27.012	12.715	8.508	6.942	3.136	1.975	6.863	0.000
82.270	28.078	12.460	26.340	11.834	8.992	6.744	3.003	1.823	6.657	1.600
186.723	0.755	1.791	0.812	1.497	0.079	0.104	0.013	0.029	0.419	2.300
					29 velocit	/ 15 m/e				
f h timo	m h timo	d h timo	m distanco	d distanco	m waiting t	d waiting t	m docisions	d docisions	d door distrib	unboardod
61 450	26.290	10.255	25.671	12.075	10.956	0_waiting_t	2 472	2 162	6 426	
01.450	20.369	10.255	35.071	12.975	10.630	7.220	3.472	2.102	0.420	0.000
92.750	27.009	12.347	36.530	16 125	11.026	7.223	3.002	2.317	6.644	0.000
70.250	26.244	10.331	35.350	12 225	11.030	7.505	3.490	2.237	6.252	0.000
93.000	20.244	14 152	37.710	19.452	11.007	7.440	3.740	2.433	6.956	0.000
63.000	27.007	14.152	37.710	10.452	11.104	1.212	3.304	2.207	0.000	0.000
77.850	26.925	12.009	36.373	15.594	10.941	7.430	3.581	2.291	6.550	0.000
86.489	0.419	2.872	0.915	5.870	0.042	0.103	0.012	. 0.011	0.052	0.000
					30 velocity 2	5+/-1.5 m/s				
f h time	m h time	d h time	m distance	d distance	m waiting t	d waiting t	m decisions	d decisions	d door distrib	unboarded
10 600	25 302	10.169	71 326	29.401	13 2/15	8 352	3 756	2 351	5 /37	0.000
52 500	25 080	10.306	71 593	29 109	13.069	8.076	3.316	2 119	5 795	0.000
50 700	25.009	10.263	70.649	28 215	12 933	8 459	3 284	1 964	5 542	0.000
48 850	24 705	9 855	70 412	28 313	13 236	8 001	3 440	2 139	5 438	0.000
51,550	25.430	10.858	72.694	30.448	13.075	8.953	3.568	2,490	5.941	0.000
E0.000	_000	10.000	74 005	00.007	10.110	0.000	0.000		E 604	0.000
50.640	25.125	0.420	/1.335	29.097	13.112	8.368	3.4/3	2.213	5.631	0.000
2.144	0.089	0.132	0.809	0.828	0.017	0.143	0.038	0.043	0.051	0.000
					31 20% (Groups				
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
82.350	28.302	14.817	44.192	21.282	11.449	8.050	4.024	2.880	7.280	0.000
80.050	27.545	13.070	42.915	18.627	11.356	8.236	4.032	2.627	7.248	0.000
52.600	25.690	9.225	39.732	13.257	11.078	7.236	3.480	2.232	5.745	0.000
76.150	27.562	13.668	43.370	20.276	11.625	8.075	3.996	2.540	6.925	0.000
73.650	26.751	12.657	42.227	19.491	10.403	7.084	3.668	2.396	6.779	0.000
72.960	27.170	12.687	42.487	18.587	11.182	7.736	3.840	2.535	6.795	0.000
140.906	0.985	4.408	2.882	9.836	0.229	0.285	0.064	0.060	0.390	0.000
					32 50%	Groups				
f h time	m h time	d h time	m distance	d distance	m waiting t	d waiting t	m decisions	d decisions	d door dietrib	unboarded
74 600	27 120	12 151	42 919	18 561	10 287	7 563	3 976	2 653	6 723	0.000
84 100	28 395	14 595	44 280	21 187	11 451	8 154	4 116	2 374	7 352	0.000
78 550	26.816	12 138	41 780	17 761	10.936	7 583	4 408	2 878	6 737	0.000
78 150	28.950	15 381	45 944	23 215	10.800	8 213	3 723	2 416	7 332	1 000
79,350	28,286	14.818	44.989	22.136	10.983	7.983	3.880	2.524	7.326	0.000
78 050	27 012	13 817	43 093	20 572	10 894	7 800	4 021		7 004	0 200
11.001	21.913	2 /10	43.982	20.372	0 174	0.000	4.021	2.009	0.111	0.200
11 600			6.1.1.1		11.1/4	11.11.20	1.101	17.174		1.2.10

					33 insta	nt start				
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
67.550	20.075	10.882	30.507	16.320	9.646	6.824	1.984	1.451	7.378	0.000
49.750	19.462	10.630	29.610	16.556	9.176	6.452	1.948	1.586	6.593	0.000
67.700	19.465	10.943	29.306	16.499	9.778	6.981	1.688	1.082	6.849	0.000
68.450	19.682	10.774	29.693	16.350	9.740	6.572	1.732	1.194	6.898	0.000
69.050	19.610	10.847	29.537	16.106	10.152	6.999	1.520	0.893	6.779	0.000
64.500	19.659	10.815	29.731	16.366	9.698	6.766	1.774	1.241	6.899	0.000
68.353	0.063	0.014	0.209	0.031	0.122	0.060	0.037	0.078	0.085	0.000
					34 late	start				
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
81.500	34.382	12.814	55.023	19.767	12.935	7.769	5.492	3.315	6.514	0.000
65.000	33.605	11.841	53.611	18.862	12.629	7.311	6.440	4.729	6.215	0.000
61.100	33.247	11.204	53.097	18.075	12.462	7.069	6.372	4.621	5.793	0.000
65.300	33.673	11.536	53.379	18.028	12.771	7.928	6.124	3.769	6.176	0.000
62.000	33.453	11.255	53.078	17.820	12.461	7.192	6.296	4.135	6.107	0.000
66.980	33.672	11.730	53.638	18.510	12.652	7.454	6.145	4.114	6.161	0.000
69.237	0.184	0.432	0.648	0.650	0.042	0.140	0.147	0.348	0.067	0.000
					35 small wait	ing area OT				
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
57.800	30.674	10.000	49.166	13.398	12.546	8.633	6.582	3.004	8.822	9.000
52.850	30.795	8.526	49.632	12.557	12.886	8.167	6.828	3.367	8.017	7.000
57.850	30.968	9.849	49.157	13.864	12.595	8.956	6.000	3.081	8.663	8.000
54.350	31.047	9.066	49.275	12.465	12.579	8.370	6.689	3.193	8.341	10.000
56.250	30.571	9.543	48.175	12.652	13.006	8.199	6.370	3.102	8.594	8.000
55.820	30.811	9.397	49.081	12.987	12.722	8.465	6.494	3.149	8.487	8.400
4.802	0.039	0.364	0.294	0.377	0.044	0.110	0.104	0.019	0.099	1.300
					36 small wai	ting area TT				
f_b_time	m_b_time	d_b_time	m_distance	d_distance	m_waiting_t	d_waiting_t	m_decisions	d_decisions	d_door_distrib.	unboarded
78.200	30.670	14.556	48.422	20.967	12.447	9.275	4.804	2.772	8.589	0.000
75.800	31.033	14.920	49.188	21.539	12.267	8.725	5.512	3.328	8.824	0.000
75.950	30.625	14.776	48.042	21.843	12.499	9.647	4.980	2.947	8.483	0.000
78.050	30.591	15.266	48.046	21.808	12.423	9.501	4.564	2.592	8.775	0.000
74.900	30.067	13.890	47.403	20.096	12.242	9.685	4.920	2.777	8.567	0.000
76.580	30.597	14.682	48.220	21.251	12.376	9.367	4.956	2.883	8.647	0.000
2.153	0.119	0.262	0.427	0.540	0.013	0.155	0.122	0.078	0.021	0.000

0.000