Manual of Agent-Based Tsunami Evacuation Modelling Using MATSim

Version 2.0

Ario Muhammad[1,2], Raffaele De Risi[1], Flavia De Luca[1], Nobuhito Mori[3], Tomohiro Yasuda[4] and Katsuichiro Goda[5,6]

1 Department of Civil Engineering, University of Bristol, Bristol, UK.

2 Department of Civil Engineering, Narotama University, Surabaya, Indonesia.

3 Disaster Prevention Research Institute, Kyoto University, Kyoto, Japan.

4 Kansai University, Osaka, Japan.

5 Department of Earth Sciences, Western University, London, Canada.

6 Department of Statistical & Actuarial Sciences, Western University, London, Canada.

1. Introduction

Tsunami is one of the deadliest natural hazards in the world resulting in a fatality number of >200,000 people and loss of > 14,000 billion USD in the last 20 years (Boen, 2005; Rossetto et al., 2007). One of the most essential issues in reducing the tsunami risk in terms of fatality is to develop effective tsunami evacuation plans. In general, current tsunami evacuation scenarios are based on a static evacuation which calculates the shortest route or the quickest time to the evacuation areas (e.g. higher grounds) ignoring a realistic model of evacuees' movement (Taubenböck et al., 2009; Riacnho et al., 2013; Muhammad et al., 2017). An agent-based modelling, which simulates each evacuee action to the evacuation areas, has been recently used for developing realistic tsunami evacuation scenarios (Zarbouti and Marmaras, 2007; Chen and Zhan, 2008). However, rigorous hazard analysis considering the uncertainty of future tsunamigenic events has not been taken into account (Mas et al., 2012, 2015; Wang et al., 2016; Makinoshima et al., 2018). For instance, places >10-m high are designated as tsunami evacuation areas in an ad-hoc manner without an extensive assessment of possible inundation models in a region of interest (Lämmel et al., 2010; Di Maoro et al., 2013).

Moreover, several studies directly adopt a tsunami hazard map (e.g. 2,500-years return period) from Probabilistic Tsunami Hazard Analysis (PTHA) as a deterministic, worst-case scenario to assess the tsunami hazard (Mostafizi et al., 2017). Adopting the deterministic scenario may lead to underestimation of the hazard level and hence, produces an inaccurate tsunami evacuation model (Muhammad et al., 2018). Recently, probabilistic earthquake source models considering the uncertainty and dependency of earthquake source parameters have been successfully used to capture the uncertainty of future tsunami hazards in Mexico, Indonesia and Japan (Goda et al., 2014; Fukutani et al., 2015; De Risi and Goda, 2016; Griffin et al., 2016; Muhammad et al., 2016; Grezio et al., 2017; Mori et al., 2017). It has been further implemented to calculate tsunami risks in terms of economic loss and fatality in those regions (Muhammad et al., 2017; Goda and De Risi, 2018). However, an agent-based evacuation modelling in tandem with the probabilistic tsunami hazard assessment results has not been carried out in those studies. Subsequently, integrating the probabilistic earthquake source models and the agent-based simulation can simulate comprehensive tsunami evacuation plans and capture the the uncertainty of tsunami evacuation models that has not been taken into account in the previous studies (Zarbouti and Marmaras, 2007; Chen and Zhan, 2008; Taubenböck et al., 2009; Riacnho et al., 2013; Muhammad et al., 2017).

In general, two approaches are used in agent-based modelling i.e. microscopic and macroscopic models (Krauß et al., 1997; Kerner et al., 2012). The macroscopic approach simulates a movement of a crowd composed of multi-agents by adopting the liquid flow model, whereas the microscopic method generates a preferred movement of each agent (Lämmel et al., 2010; Kerner et al., 2012). Therefore, the microscopic model can produce a more realistic agent movement than the macroscopic one (Fellendorf et al., 2001; Lämmel et al., 2010). Several models have been developed using the microscopic methods including Cellular Automata (CA), Molecular Dynamic (MD), and Dynamic Traffic Assignment (DTA; Nagel, and Schreckenberg, 1992; Helbing and Tilch, 1998; Maerivoet and De Moor, 2005). The CA converts large areas into a cell resulting in a large computational time, whereas the MD needs extensive social interaction data among the agents (Lämmel et al., 2010). On the other hand, the DTA model defines time-dependent trips on routes that can be modified due to specific sudden events (e.g. tsunami hazard; Lämmel et al., 2010; Horni et al., 2016). Therefore, the DTA approach is the most suitable candidate to develop tsunami evacuation models (Lämmel et al., 2010).

Numerous softwares have been developed by adopting the DTA model for evacuation, such as MITSIM (Jha et al., 2004), PARAMICS (Chen and Zhan, 2004), VISSIM (Han et al., 2007), and MATSim (Balmer et al., 2009). Generally, the queueing system of traffic flow simulation in those packages considers all streets as areas rather than a link leading to a high computational cost (Lammel et al., 2009; Horni et al., 2016). However, MATSim provides an efficient and complex adaptive system that can improve the accuracy and computational speed for tsunami evacuation purposes (Balmer et al., 2009; Lammel et al., 2009). For instance, the MATSim code can be modified to adopt a simplified queuing system which models the streets into nodes and links that can speed up the simulation time (Balmer et al., 2009; Lammel et al., 2009). Moreover, the movement plans of each agent in MATSim can be updated in each iteration and modified in a pre-defined time basis (e.g. 5 minutes) and hence, this package is reliable for developing tsunami evacuation scenarios (Balmer et al., 2009; Lammel et al., 2009; Goetz and Zipf, 2012; Horni et al., 2016).

Subsequently, a new framework of tsunami evacuation modelling that incorporates the probabilistic earthquake source models and the agent-based movement is developed. This manual aims to provide detailed procedures on running the agent-based tsunami evacuation modelling using MATSim incorporating the probabilistic earthquake source modelling. First, an introduction to MATSim is presented to understand the basic concepts of simulating the agent mobility in MATSim. Second, more specific features of MATSim are discussed to provide detail information about the coding/data to run the simulation. Third, two case studies are mentioned: (1) a simple case study to test the MATSim package and (2) a more complex study case from the development of tsunami evacuation plans in Padang using probabilistic earthquake source models. Fourth, a MATLAB codes section is shown to illustrate how the pre and post-processing simulation files are produced. Finally, conclusions are drawn.

2. Introduction to MATSim

MATSim software is based on a microscopic traffic modelling and allows performing a simulation of traffic flows and congestion produced from the flow. It uses schedule of each agent and a synthetic individual decision to define the flow and hence; thus it can be categorized as an "agent-based". The MATSim also adopts a complex adaptive system, so called co-evolutionary algorithm. With this algorithm, each agent iteratively simulates their activities (e.g. normally for one-day simulation) and further competes with other agents for space-time slots on the existing route. It is currently implemented in Java (Balmer et al., 2009; Horni et al., 2016).

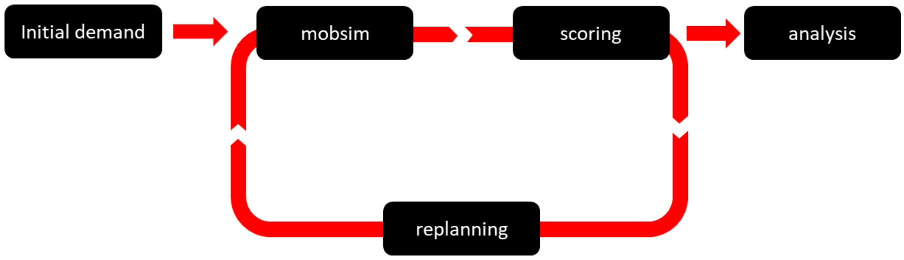

Figure 1 Simulation procedures in MATSim (Horni et al., 2016).

Figure 1 shows the simulation procedure in MATSim. First, initial demand is introduced into the system providing activities of each agent. Inside the schedules, information on the time, mode, and destination choices need to be provided. Based on such planned activities, mobility simulation (mobsim) models the agents' movement to their destination using the existing network/routes. The routes used by the agents are then scored which can be interpreted as an econometric utility. The detail on scoring can be found in Axhausen et al. (2016). After running the first mobility simulation based on the initial demand, iteration of a new mobility simulation is implemented and re-scored for each trial. The best score (i.e. the highest one) is finally used to define the best route (Horni et al., 2016).

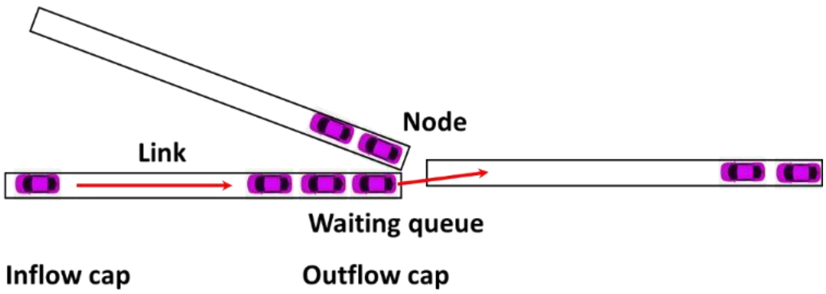

Figure 2 Model of traffic flow in MATSim (Horni et al., 2016)

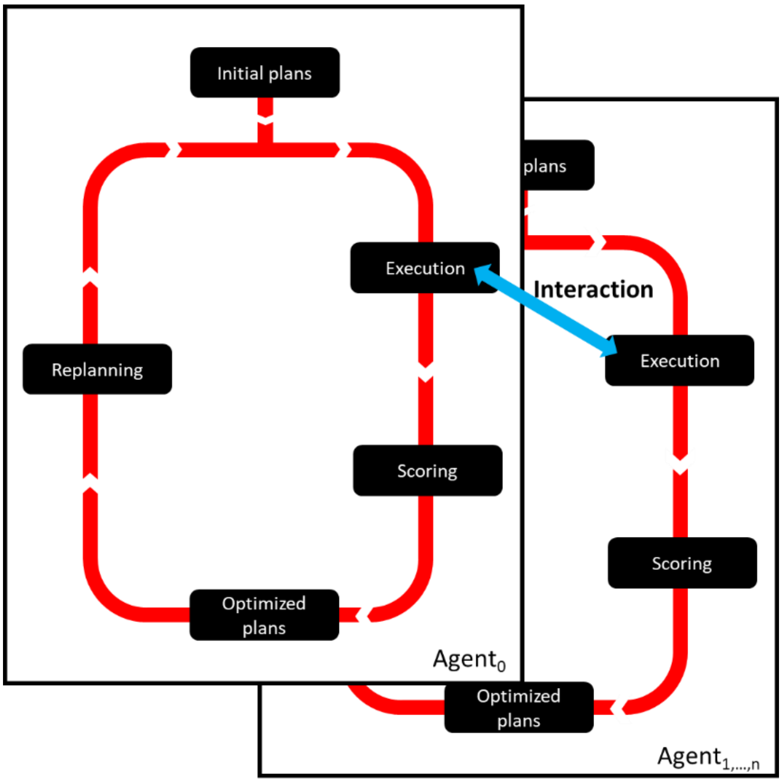

Figure 3 Co-evolutionary algorithm used in MATSim (Horni et al., 2016).

To run the mobility simulation, the existing networks/routes need to be defined as link (i.e. road segment) and node (i.e. and end-point of the link; see Figure 2), whereas a mode used for the agent (e.g. car) is also specified. Car is a default mode in MATSim. The traffic flow model for the mobsim implemented in MATSim is a computationally efficient queue-based approach. Figure 2 illustrates the traffic flow model in MATSim. Each transportation mode (e.g. car) coming from the inflow cap directly goes to the next link when the link is free (e.g. below a minimum route capacity). If an intersection exists, the mode within the link will be put at the tail of the waiting queue and the mode coming from that intersection link will stay at the end of the link near the intersection until all the modes at the free link have moved into the new link. Such an approach may not represent realistic scenarios. However, it is very efficient and can reduce computational time. Moreover, MATSim also adopts the co-evolutionary algorithm to model the interaction among the agents (Figure 3). In this algorithm, optimization is performed iteratively based on the score produced from the interaction of each agent following their schedules.

3. Agent-based simulation using MATSim

The MATSim package is available for download at http://MATSim.org. All documents, updates and extended works that use MATSim are provided at this website. To run MATSim on a PC, one needs to install the Java Standard Edition (Java SE) to compile the MATSim package. A graphical user interface will appear when the installation screen of MATSim is open as presented in Figure 4. The 'choose' in Figure 4 is used to open the input file for the simulation. The 'run' button will be shown after inputing the input file and is used to run the simulation. A default format of the file used to run MATSim is an Extensible Markup Language (XML) file. Subsequently, all the simulation data (e.g. plans and networks) must be converted into XML file before running in MATSim.

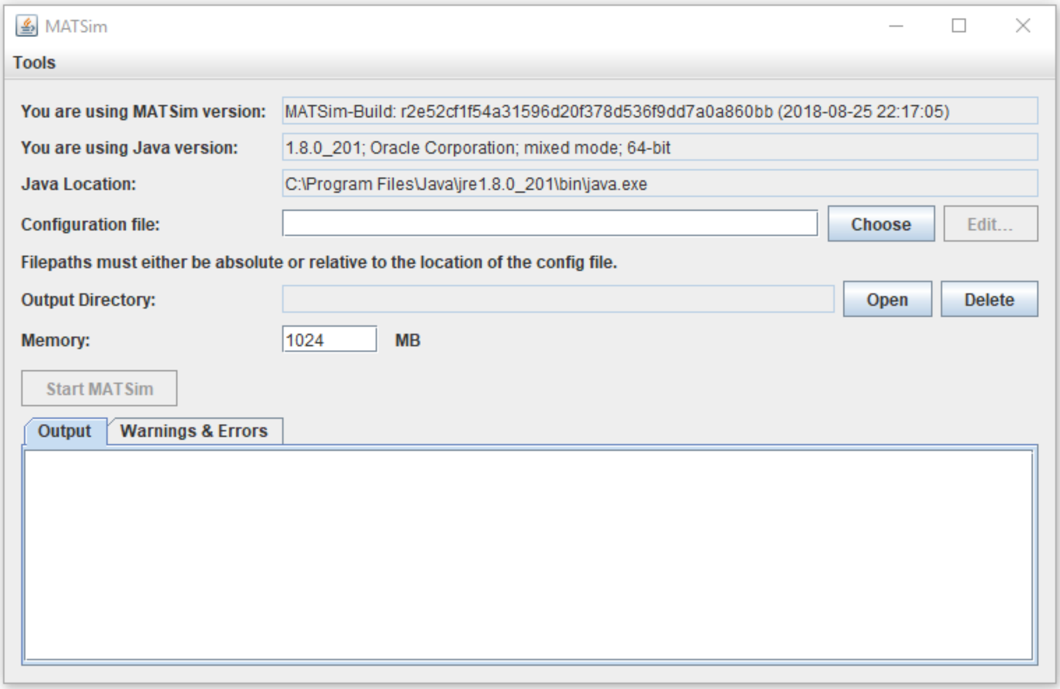

Figure 4 Installation screen of MATSim.

3.1. Coordinate system and units

The Universal Transverse Mercator (UTM) is a default coordinate system in MATSim. Any spherical coordinate systems (e.g. The World Geodetic System 84 (WGS84)) is not recommended for the simulation since MATSim needs to calculate the distance between points to execute the agents' plan. Moreover, two essential parameters for the simulation: distance and time are set up following the used coordinate system. For instance, if the UTM is adopted, the distance parameter (e.g. length of the link) is in **meter**. On the other hand, the unit for time is in **second**.

3.2. Input data

Three files are required to run a simulation in MATSim: (1) config.xml, (2) network.xml, and (3) plans.xml. The configuration file (config) contains lists of setting to define the traffic simulation. Inside the configuration file, two essential data: network and population/plans are connected to the two other XML files (i.e. network.xml and plans.xml). The network.xml is an XML file containing all the network information in a region of interest. It explains the detail of nodes and links (see Figure 2) used by the agent. Coordinates of nodes and links are written in the network file. Moreover, the free-speed of mode and the free capacity of links need to be specified in the links section within the network file (see Figure 6). The population file is in general saved as a "plans" since it informs the required travel plan for each agent. As presented in Figure 7, the plans/population file characterizes the travel requirement of the agent represented by the ID. It provides name, coordinates link, duration, and end time of each activity. Either the coordinate or link is normally used to define where the activity takes place, whereas the duration or end time represents how long the activity will take. Moreover, an extra xml file (i.e. vehicles.xml) may be provided to set up the mode's property used by the agents. However, such extra file can be ignored if the 'car' (i.e. defult mode) is considered.

```
<?xml version="1.0"?>
<!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v2.dtd">
- <config>
  - <module name="network">
        <param name="inputNetworkFile" value="network.xml"/>
    </module>
  - <module name="plans">
        <param name="inputPlansFile" value="plans.xml"/>
    </module>
  - <module name="controler">
        <param name="firstIteration" value="0"/>
        <param name="lastIteration" value="10"/>
    </module>
  - <module name="planCalcScore">
        <param name="learningRate" value="1.0"/>
        <param name="BrainExpBeta" value="1.0"/>
        <param name="lateArrival" value="-18"/>
        <param name="earlyDeparture" value="-18"/>
        <param name="performing" value="+6"/>
        <param name="traveling" value="-6"/>
        <param name="waiting" value="0"/>
        <param name="activityType_0" value="home"/>
            <!-- home -->
        <param name="activityPriority_0" value="1"/>
        <param name="activityTypicalDuration_0" value="12:00:00"/>
        <param name="activityMinimalDuration_0" value="08:00:00"/>
        <param name="activityType_1" value="work"/>
            <!-- work -->
        <param name="activityPriority_1" value="1"/>
        <param name="activityTypicalDuration_1" value="08:00:00"/>
        <param name="activityMinimalDuration_1" value="06:00:00"/>
        <param name="activityOpeningTime_1" value="07:00:00"/>
        <param name="activityLatestStartTime_1" value="09:00:00"/>
        <param name="activityEarliestEndTime_1" value=""/>
        <param name="activityClosingTime_1" value="18:00:00"/>
    </module>
  - <module name="strategy">
        <param name="maxAgentPlanMemorySize" value="5"/>
            <!-- 0 means unlimited -->
        <param name="ModuleProbability_1" value="0.9"/>
        <param name="Module_1" value="BestScore"/>
        <param name="ModuleProbability_2" value="0.1"/>
        <param name="Module_2" value="ReRoute"/>
    </module>
</config>
```

Figure 5. Example of configuration file.

```
  <?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE network SYSTEM "http://matsim.org/files/dtd/network_v1.dtd">
- <network>
  - <nodes>
      <node id="1" y="0.0" x="0.0"/>
      <node id="2" y="0.0" x="2500.0"/>
      <node id="3" y="0.0" x="5000.0"/>
    </nodes>
  - <links capperiod="10:00:00">
      <link id="1" from="1" to="2" length="2500" freespeed="22" capacity="2000" permlanes="1" modes="car"/>
      <link id="2" from="2" to="3" length="2500" freespeed="22" capacity="2000" permlanes="1" modes="car"/>
      <link id="3" from="3" to="1" length="1980" freespeed="22" capacity="2000" permlanes="1" modes="car"/>
    </links>
  </network>
```


Figure 6. Example of a network file.


```
  <?xml version="1.0"?>
  <!DOCTYPE plans SYSTEM "http://www.matsim.org/files/dtd/plans_v4.dtd">
- <plans>
  - <person id="1">
    - <plan>
        <act end_time="06:00" link="1" y="0" x="0" type="home"/>
      - <leg mode="car">
          <route> </route>
        </leg>
        <act link="2" y="0" x="5000" type="work" dur="00:05"/>
      - <leg mode="car">
          <route> </route>
        </leg>
        <act link="2" y="0" x="5000" type="work" dur="08:00"/>
      - <leg mode="car">
          <route> </route>
        </leg>
        <act link="3" y="0" x="0" type="home"/>
      </plan>
    </person>
  </plans>
```


Figure 7. Example of a plans file.

Additional parameters need to be defined in the config file including the scoring and strategy. The scoring shown by the planCalcScore section in Figure 55 contains the parameters to calculate the score of each plan. The following lists current MATim default values of scoring parameters:

$$\beta_m = 1\ utils/monetary\ unit$$

$$\beta_{dur} = 6\ utils/h$$

$$\beta_{trav,mode(q)} = -6\ utils/h$$

$$\beta_{wait} = 0\ utils/h$$

$$\beta_{short.dur} = 0\ utils/h$$

$$\beta_{late.ar} = -18\ utils/h$$

$$\beta_{early.dp} = 0\ utils/h$$

β_m is a marginal utility. It is normally set to a positive value (e.g. 1.0 by default) since having more money generally increases utility. $\beta_{dur}$is a factor to represent the duration utility. It can be set up from a minus (-) to a positive (+) value. The default value is +6 because performing longer activity escalates utility. $\beta_{trav,mode(q)}$ defines a marginal utility of traveling using a specific mode (e.g. car or public transport (pt)). $\beta_{trav,mode(q)}$ indicates that while traveling, the agent may not perform any other activity resulting in decreasing the duration and hence, is normally set to a negative value. β_{wait} indicates a marginal utility of waiting time spent, whilst $\beta_{short.dur}$ is a marginal utility of the shortest possible activity and is generally set up to 0. A zero value means that the waiting time and the shortest activity have no effect on the performed activity. $\beta_{late.ar}$ indicates the marginal utility of late-arrival defaulting by a minus value (-18) because it slows the duration of the agent activity. On the other hand, a marginal utility of early departure ($\beta_{early.dp}$) is normally set to zero exhibiting no influence on the activity. Moreover, the strategy section on the configuration file shows the change plans on each iteration. Subsequently, the users may define their updated plan strategies within this section e.g. changing from a normal route of a specific activity to the evacuation links. The maxAgentPlanMemorySize written below the strategy in Figure 5 is a default in MATSim to control the memory used during the simulation, i.e. the plan with the lowest score is deleted within such a strategy (Balmer et al., 2009; Horni et al., 2016).

To run the simulation, the input files (i.e. 'config.xml', 'network.xml', and 'plans.xml') are saved at the same specified folder by the user. The configuration file ('config.xml') is used as the only input in a graphical user interface of MATSim (see Figure 4) because it connects the user's task and MATSim to run the simulation. After running the simulation, the output folder is then stored at the same folder where all the input xml files are located.

3.3. Output data

After running the simulation, MATSim creates an output folder containing the following information/files:

1. **Log File**: a complete log file containing all the simulation details
2. **Warnings and Errors Log File**: warnings and errors that occurred during the simulation
3. **Score Statistics**: average best, worst, executed and the overall average **score** of each agent's activity in every iteration; this information is saved in txt and image file formats
4. **Leg Travel Distance Statistics**: leg travel distance statistics
5. **Stopwatch**: simulation time for each iteration

6. output_events.xml: the final plan of each agent after the simulation.

The most important file to extract the simulation results is 'output_events.xml' because it contains all the information of each agent's arrival time at each node/link that is useful to calculate the evacuation time. Other results, such as network, plan, and configuration information, are also provided in the output folder.

Another folder named **ITER** is also generated to provide complete information in each iteration and contains the following files:

1. **Events**: every action taken by the agent for each iteration
2. **Leg Histogram**: the number of agent's activity per time unit in every iteration
3. **Trip Duration**: duration of activity for each iteration

4. Example of MATSim simulation

A simple case study is run to test the agent-based modelling using MATSim and to understand how the MATSim simulation works. All simulation files are indexed with a prefix 'SC' refers to **S**imple **C**ase study. Only **one agent** is used to perform mobility simulation. Figure 8 illustrates such a case study. 15 nodes (blue circles) are adopted to represent the location/destination of each plan corresponding to certain coordinates. The coordinates of nodes are presented above the blue circles. The nodes are connected with the links (green lines) indicating the street/route. All streets have one lane and hence, a total of 23 links are drawn. In case the travel of the agents is in two-way, the links need to be provided for both directions between the same nodes.

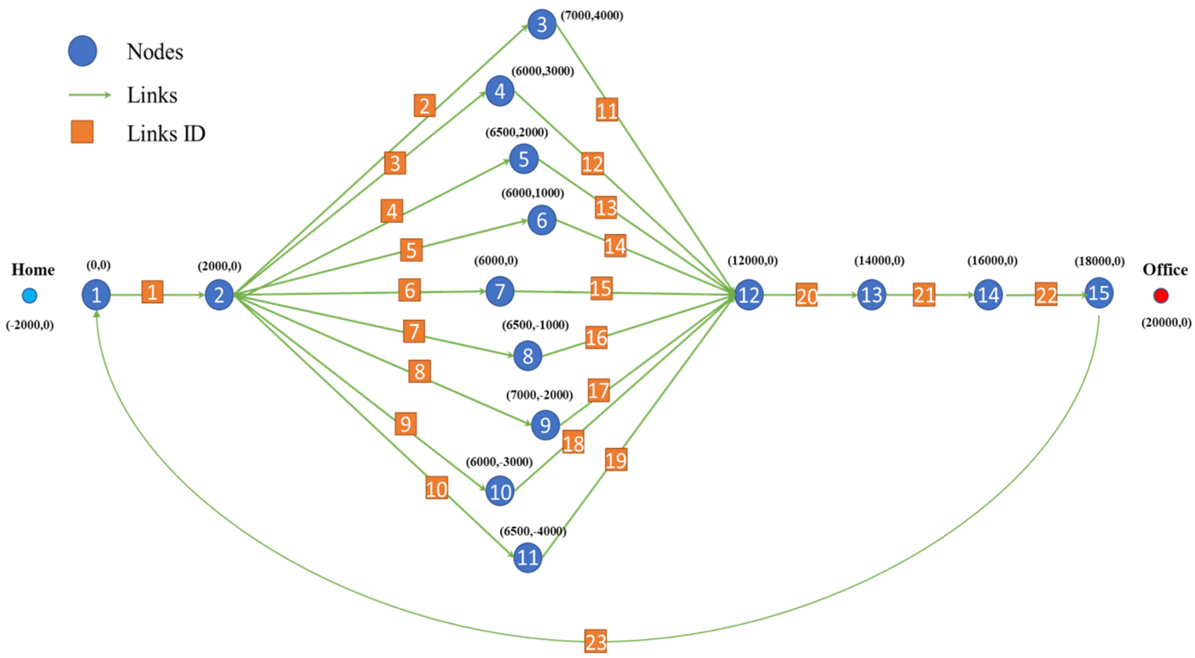

Figure 8. Illustration of a case study network.

Following the network information, a configuration file is generated based on the agent plans and the network data. A plan of the agent is then constructed to run the mobility simulation. In this case, four plans are designed as follows:

1. **Home1** represents the agent's activity at home (light blue in Figure 8) until he/she moves to the car. The location is at the coordinate of $x = -2{,}000$ and $y = 0$ and the end time of this activity is 08.00 am.
2. **Work1** represents the agent's mobility from home to work place at the coordinate of $x = 20{,}000$ and $y = 0$ with the initial plan duration of 10 minutes.
3. **Work2** represents the end of work1 and indicates the starting event of mobility from work to home.
4. **Home2** represents the last activity of the agent from the car arriving at home to the agent's house.

```xml
  <?xml version="1.0"?>
  <!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v2.dtd">
- <config>
  - <module name="network">
        <param name="inputNetworkFile" value="network.xml"/>
    </module>
  - <module name="plans">
        <param name="inputPlansFile" value="plans1.xml"/>
    </module>
  - <module name="controler">
        <param name="firstIteration" value="0"/>
        <param name="lastIteration" value="10"/>
    </module>
  - <module name="planCalcScore">
        <param name="learningRate" value="1.0"/>
        <param name="BrainExpBeta" value="2.0"/>
        <param name="lateArrival" value="-18"/>
        <param name="earlyDeparture" value="-0"/>
        <param name="performing" value="+6"/>
        <param name="traveling" value="-6"/>
        <param name="waiting" value="-0"/>
        <param name="activityType_0" value="h"/>
          <!-- home -->
        <param name="activityPriority_0" value="1"/>
        <param name="activityTypicalDuration_0" value="12:00:00"/>
        <param name="activityMinimalDuration_0" value="08:00:00"/>
        <param name="activityType_1" value="w"/>
          <!-- work -->
        <param name="activityPriority_1" value="1"/>
        <param name="activityTypicalDuration_1" value="08:00:00"/>
        <param name="activityMinimalDuration_1" value="06:00:00"/>
        <param name="activityOpeningTime_1" value="07:00:00"/>
        <param name="activityLatestStartTime_1" value="09:00:00"/>
        <param name="activityEarliestEndTime_1" value=""/>
        <param name="activityClosingTime_1" value="18:00:00"/>
    </module>
  - <module name="strategy">
        <param name="maxAgentPlanMemorySize" value="5"/>
          <!-- 0 means unlimited -->
        <param name="ModuleProbability_1" value="0.9"/>
        <param name="Module_1" value="BestScore"/>
    </module>
  </config>
```

Figure 9. Configuration file used in the simple case study.

Subsequently, a configuration, network, and plans files in XML format are generated based on the above constraints to run the mobility simulation as presented in Figure 9 to Figure 11, respectively. The configuration file shows that network.xml and plans.xml are essential inputs to run the simulation, whilst the score parameters and mobility strategy are defined in the sections after the network and plans command. The name of the network and plans files are highlighted with the magenta color in Figure 9. Those names need to be exactly the same as the saved file of both the network and plans in the simulation folder. The number of iterations is set to 10 and it can be increased based on the user requirement. Balmer et al. (2009) and Horni et al. (2016) recommend the minimum number of 100 iterations to obtain stable scoring results. However, it can be as many as 250 iterations when the case is more complex. Moreover, several data regarding the activity duration is specified at the configuration files including the minimum and typical duration for **work** activity.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE network SYSTEM "http://www.matsim.org/files/dtd/network_v1.dtd">
- <network>
  - <nodes>
      <node y="0" x="0" id="1"/>
      <node y="0" x="2000" id="2"/>
      <node y="4000" x="7000" id="3"/>
      <node y="3000" x="6000" id="4"/>
      <node y="2000" x="6500" id="5"/>
      <node y="1000" x="7000" id="6"/>
      <node y="0" x="6000" id="7"/>
      <node y="-1000" x="6500" id="8"/>
      <node y="-2000" x="7000" id="9"/>
      <node y="-3000" x="6000" id="10"/>
      <node y="-4000" x="6500" id="11"/>
      <node y="0" x="12000" id="12"/>
      <node y="0" x="14000" id="13"/>
      <node y="0" x="16000" id="14"/>
      <node y="0" x="18000" id="15"/>
    </nodes>
  - <links>
      <link id="1" permlanes="1" freespeed="27.78" capacity="1500" length="2000.00" to="2" from="1"/>
      <link id="2" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="3" from="2"/>
      <link id="3" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="4" from="2"/>
      <link id="4" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="5" from="2"/>
      <link id="5" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="6" from="2"/>
      <link id="6" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="7" from="2"/>
      <link id="7" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="8" from="2"/>
      <link id="8" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="9" from="2"/>
      <link id="9" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="10" from="2"/>
      <link id="10" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="11" from="2"/>
      <link id="11" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="12" from="3"/>
      <link id="12" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="12" from="4"/>
      <link id="13" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="12" from="5"/>
      <link id="14" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="12" from="6"/>
      <link id="15" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="12" from="7"/>
      <link id="16" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="12" from="8"/>
      <link id="17" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="12" from="9"/>
      <link id="18" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="12" from="10"/>
      <link id="19" permlanes="1" freespeed="27.78" capacity="8000" length="10000.00" to="12" from="11"/>
      <link id="20" permlanes="1" freespeed="27.78" capacity="1500" length="2000.00" to="13" from="12"/>
      <link id="21" permlanes="1" freespeed="27.78" capacity="1500" length="2000.00" to="14" from="13"/>
      <link id="22" permlanes="1" freespeed="27.78" capacity="1500" length="2000.00" to="15" from="14"/>
      <link id="23" permlanes="1" freespeed="27.78" capacity="10000" length="25000.00" to="1" from="15"/>
    </links>
  </network>
```

Figure 10. Network file used in the simple case study.

```
<?xml version="1.0"?>
<!DOCTYPE plans SYSTEM "http://www.matsim.org/files/dtd/plans_v4.dtd">
- <plans>
  - <person id="1">
    - <plan>
        <act end_time="08:00" link="1" y="0" x="-2000" type="h"/>
      - <leg mode="car">
          <route>2 7 12</route>
        </leg>
        <act link="20" y="0" x="20000" type="w" dur="00:10"/>
      - <leg mode="car">
          <route> </route>
        </leg>
        <act link="20" y="0" x="20000" type="w" dur="08:00"/>
      - <leg mode="car">
          <route>13 14 15 1</route>
        </leg>
        <act link="1" y="0" x="-2000" type="h"/>
      </plan>
    </person>
  </plans>
```

Figure 11. Plans file used in the simple case study.

Using those three required files (i.e. 'SC_configuration.xml', 'SC_network.xml', and 'SC_plans.xml'), the following steps are carried out to run the MATSim simulation:

1. Open the MATSim package as shown in Figure 12.
2. Click the 'choose' bar to open the configuration file (see red arrow in Figure 12). The output folder name and location are automatically shown when the configuration file is chosen (see red rectangle areas in Figure 12). The output folder name and location will not be displayed in the output directory line if the configuration file is incorrect.
3. Press the Start MATSim button to run the simulation and wait until the simulation is finished.

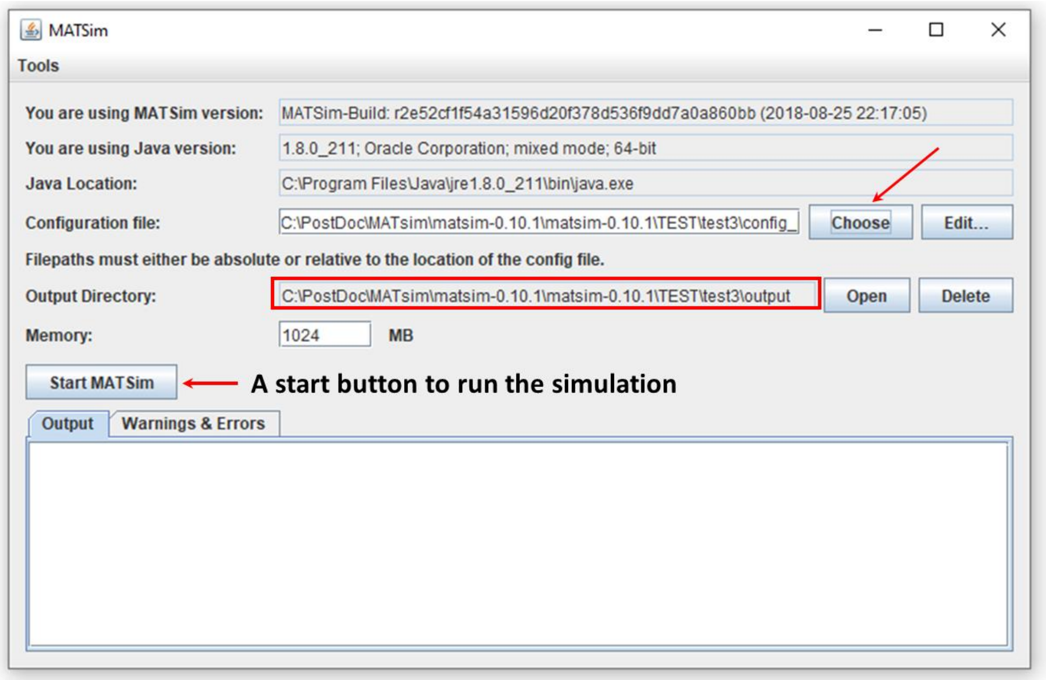

Figure 12. Opening the configuration file.

The results of this case study simulation are presented in the following:

1. Executed agent's plans and routes are shown in the XML file of 'SC_output_events.xml' that can be opend. This is the main output file because it contains the routes, plan, and time of agent's movement.
2. Mode choice in each iteration is shown in Figure 13 where the only mode is car (green line). The 'pt' mode in the bottom of Figure 13 refers to public transport.
3. The calculated score in every trial is presented in Figure 14. The results are constant because the input of initial plan is the best route and further used for the following iterations.
4. The computational time of each iteration is exhibited in Figure 15. The figure indicates that the average computational time for one iteration is about 1.1 seconds.

5. Travel distance calculated in each trial is drawn in Figure 16. The result shows that a total of ~17.5 km journey is carried out by the agent in every iteration.

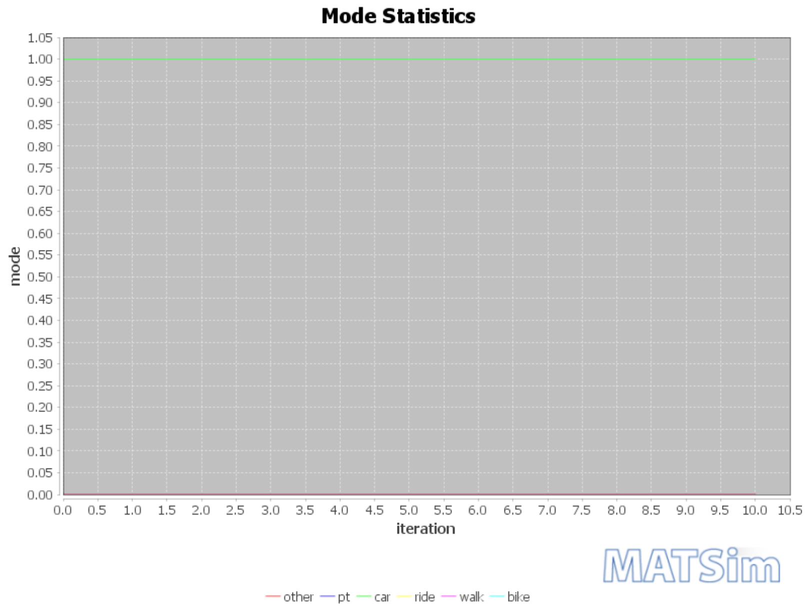

Figure 13. Mode choice.

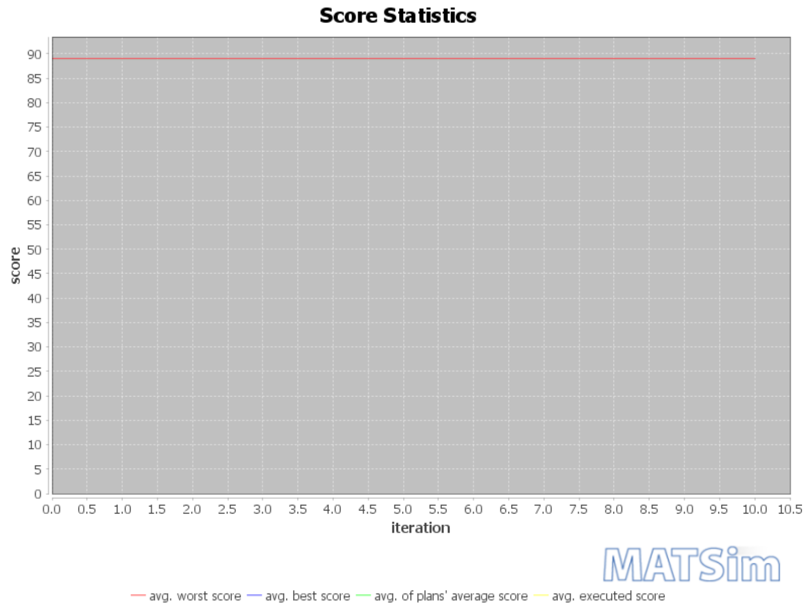

Figure 14. Calculated score.

Moreover, detail of each iteration process including trip duration, executed plans, and number of activities conducting in a specific time bin in PNG format are provided in the dataset directory. The trip duration is essential for developing tsunami evacuation scenarios. In this study case, **the trip duration is ~15 minutes**.

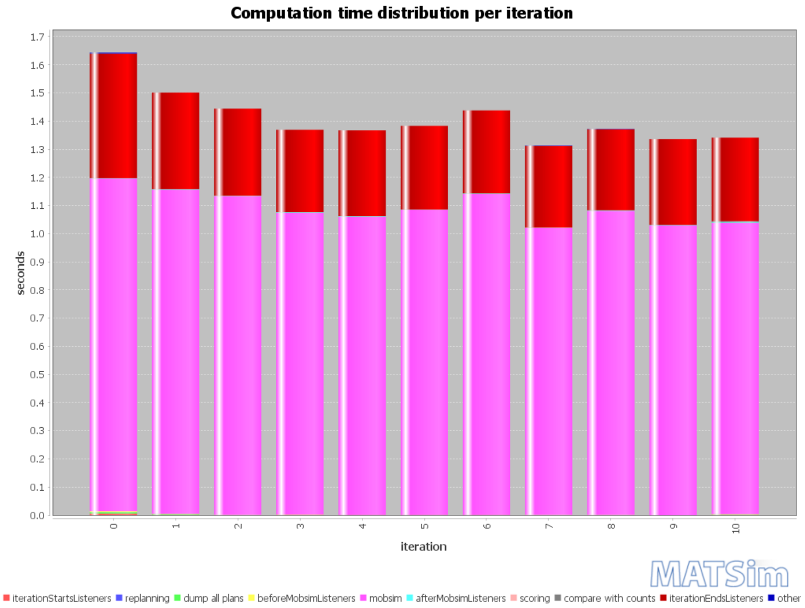

Figure 15. Simulation time in each iteration.

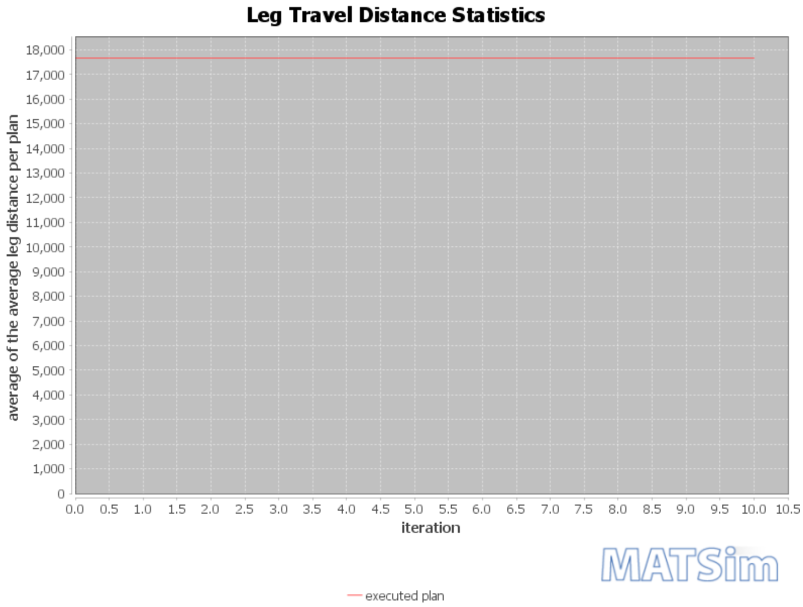

Figure 16. Traveled distance.

6. Tsunami evacuation modelling using MATSim

Following the simple case study, a tsunami evacuation modelling using MATSim is presented in this section. All simulation files are indexed with a prefix of 'EP' refers to **E**vacuation **C**ase in Padang. First, the framework of tsunami evacuation modelling is shown to introduce the concepts of evacuation modelling in MATSim. Second, a case study in one of the most tsunami-prone regions, Padang, Indonesia, is discussed. Finally, the MATLAB codes are used to develop the xml files for running the MATSim simulation and to carry out post-processing.

5.1. Simulation framework

Figure 17 shows a framework of agent-based tsunami evacuation modelling using MATSim. Several essential information is needed to generate suitable network and plan files for the simulation which are the following:

1. Geographic Information System (GIS) data including:
    - Street network and infrastructure (e.g. bridges)
    - Building footprints
    - Existing signs leading to safe places
    - Safe places
    - Digital Elevation Model (DEM)
2. Socio-economic data
    - Population distribution as a function of time (morning, afternoon, night); potentially population per building (spatio-temporal distribution of the population)
3. Earthquake-tsunami hazard maps
    - Status of bridges after earthquake action
    - Inundation scenarios (inundation results and evacuation simulations need to be synchronized)

More specifically, the network information, such as road and bridges as well as DEM, are important to develop the network file, whilst the building footprint, population data, tsunami hazard map, and tsunami evacuation shelters (TES; if existed) are essential to define the agent's plan.

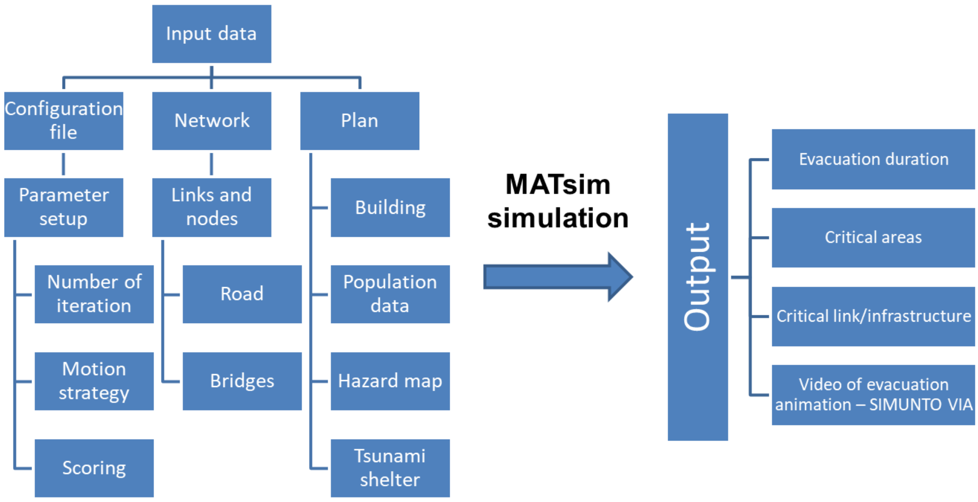

Figure 17. A framework of agent-based tsunami evacuation modelling using MATSim.

To generate the input files, first, the parameters need to be set up within the configuration files. The first parameter is the number of iteration which depends on the number of agents and the complexity of the network. A larger number of nodes and links in more complex networks tends lead to a higher iteration number. The second parameter is movement strategy, which defines the motion plan of each agent in every iteration and is in general defined with the two following motion plans (Horni et al., 2016):

1. ReRoute (10%): new plans are generated in each iteration with new evacuation routes based on the information of experienced travel times from the last run (i.e. immediate past).
2. BestScore (90%): new plans are generated and scored in each iteration. The final plan is a plan with the highest score that is determined when the average scores of all agents' plan are stable. The score representing the utility (U) is a function distance and time (Lämmel et al., 2009, 2010), as shown in Eq. 1.

$$U_i = (\beta_{tr} \times t_{tr})_i + (\beta_{dis} \times d)_i \quad (1)$$

where β is the marginal utility parameters for time to travel (t_{tr}) and distance (d).

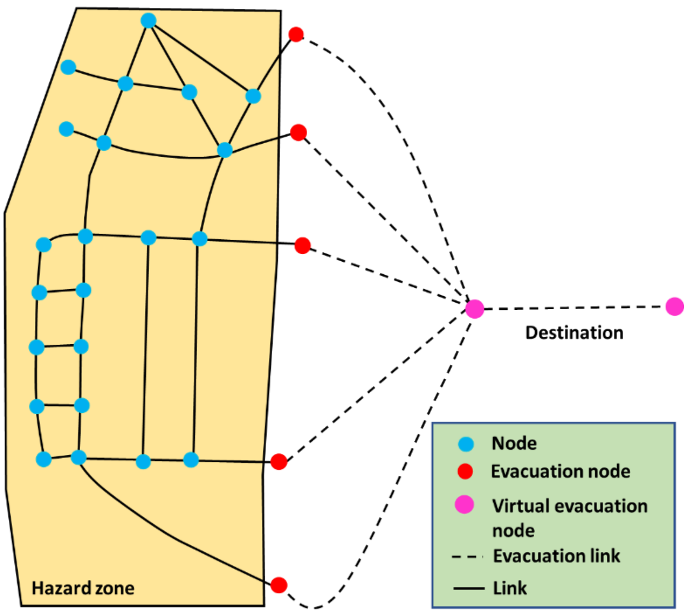

Figure 18. Network setup in tsunami evacuation modelling using MATSim.

Second, the network file is created. To generate the network file for the simulation, links and nodes that represent the networks in a region of interest need to be defined. In the tsunami evacuation modelling using MATSim, two types of nodes are developed: general node and evacuation node. The general nodes (i.e. blue dot in Figure 18) are nodes that connect the links before evacuation areas and are located within the hazard zones (i.e. yellow areas in

Figure 18). On the other hand, the evacuation nodes (i.e. red dot in Figure 18) are the final destination for evacuation located outside the hazard zone and are linked to the existing networks (i.e. links). Those evacuation nodes are further connected with the virtual nodes and links (see magenta dot and black-dash-line, respectively in Figure 18). The length of the virtual links is set up to zero showing that the final destination of each agent is actually at the evacuation nodes (red dot).

For evacuation modelling, the network is developed with the following restrictions. First, maximum flow speed is restricted depending on the mode of the agents. For instance, the maximum flow velocity for pedestrian and motorbike modes in a evacuation modelling are typically set to 1.66 m/s and 5.66 m/s, respectively (Di Mauro et al., 2013; Muhammad et al., 2017). Such speeds are further adopted for the agent-based tsunami evacuation modelling in Padang because it corresponds to the average tsunami evacuation speed for each mode regionally (Affan et al., 2012; Di Mauro et al., 2013; Muhammad et al., 2017; Kemal and Putra, 2018). Second, the maximum flow capacity (FC) is calculated using the following equation:

$$FC = w \times C_{max} \tag{2}$$

where w is the width of the link and C_{max} is the maximum flow capacity per unit width that is determined based on the mode used for evacuation simulation. For instance, the maximum flow capacity for the pedestrian is 1.3 persons/m/s. Such a parameter is determined to approximate Weidmann's fundamental diagram (Weidmann, 1993) that defines the relationship between density and velocity (Lämmel, et al., 2009). Subsequently, for the motorbike, the maximum flow velocity is set up to 4 persons/m/s considering the flow speed of motorbike is about 3 times larger than the pedestrian. Such a limitation restrics the agents to leave the link if the maximum flow capacity is reached. Finally, the link capacity in persons/m^2, i.e. storage capacity (SC), is defined to limit agents on the link and is calculated based on the areas (A) and the maximum density per unit areas (D_{max}) (see Equation 3; Lämmel, et al., 2009):

$$SC = A \times D_{max} \tag{3}$$

Consequently, if a link is filled with agents up to the maximum storage capacity, the evacuee is not removed (i.e. killed during the simulation) but forced to spill back leads to congestion (Lämmel, et al., 2009). D_{max} for pedestrian and motorbike are 5.4 persons/m^2 and 0.4 persons/m^2 (Lämmel, et al., 2009). In additon, the capacity of final evacuation links needs to be defined as infinite in order to gather all agents that successfully evacuate to the safe zones.

On the other hand, the free speed of each link can be modified by considering regional geographic information, such as the slope of the link . Tobler's hiking function can be used to model the change of speed due to the slope of the link as presented in the following equations (Tobler, 1993):

$$W = 6e^{-3.5\left|\frac{dh}{dx}+0.05\right|} \quad (4)$$

$$\frac{dh}{dx} = S = tan\theta \quad (5)$$

where W is the free speed of the link (m/s), dh is elevation difference, dx is a horizontal distance, S is a slope, and θ is the angle of slope.

Finally, the plans file is developed. The plans of individual agents contain the following information: (1) location or coordinates, (2) departure time, (3) initial link closest to the agent's location, and (4) final destination node and link. The location of each agent can be determined based on the population data and the building footprints. Consideration of distribution time (i.e. starting time of evacuation) is essential to define the agent's location. For instance, the agent may start to evacuate from an office or public areas (e.g. market) when the daytime is considered. In contrast, most of the agents may start to evacuate from their houses if the night-time is of concern. Therefore, both population data and building information are important to define how many agents need to evacuate and their locations. The departure time may be determined based on day-time or the night-time consideration. The final destination (evacuation areas) may be determined based on the hazard maps, i.e. the non-inundated areas are designated as evacuation areas. The end of the closest roads connecting to the evacuation areas are used as the evacuation nodes and are further linked to the virtual evacuation nodes and link (red dot in Figure 18). Those virtual evacuation nodes and links are used as the final destinations in developing the agent's plans. Noted that, in MATSim, the final plans of each agent need to be set up to go back to the starting point and hence, an extra plan after going to the final destination (i.e. go back to the starting location) needs to be added in the agents' plan.

Using those input files, the agent-based tsunami evacuation modelling is performed using MATSim. The simulation results can be used to extract essential information for developing an effective mitigation plan including (1) evacuation duration, (2) critical areas where the agents are mostly affected by the tsunami, and (3) critical infrastructures, such as road and bridges that may be subjected to heavy congestion. In addition, additional software (i.e.

Simunto VIA) can be adopted to visualize the agent-based tsunami evacuation simulation in animation.

5.2. Realistic case study

As a realistic case study, a region of Teluk Bayar in Padang (Figure 19), Indonesia is selected. This area is located 100 km away from the Mentawai section of the Sunda Subduction zone predicted to generate a maximum of ~M_w 9.0 in this century (Nalbant et al., 2005; Borrero et al., 2006; Natawidjaja et al., 2006; Daly et al., 2019). Two evacuation routes have been suggested by the local authorities: Air Manis Street and Bypass without defining specific building for TES (see the red dots with white-dashed circles in Figure 19B). The population data (i.e. ~9,000 people) are taken from Statistics Indonesia (BPS, 2018) and started to evacuated at 09.00 am.

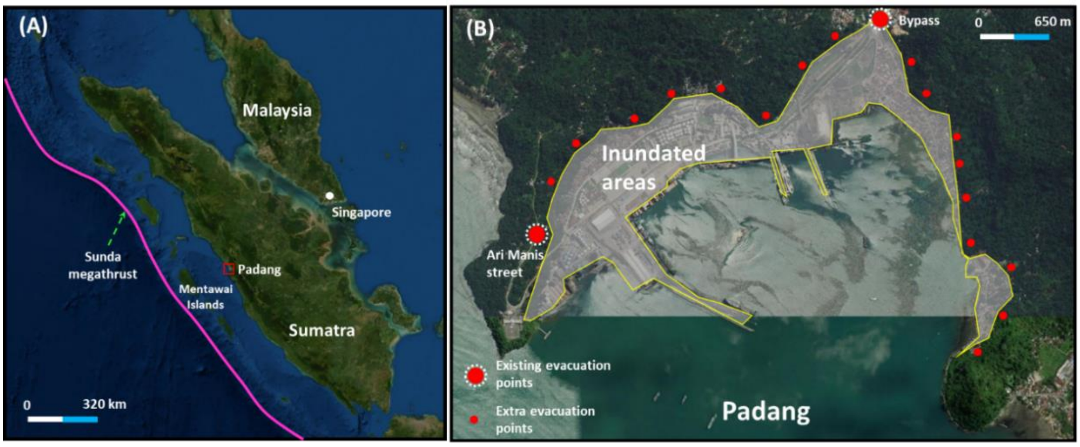

Figure 19. Study areas: (A) Mentawai-Sunda subduction zone and (B) Padang region.

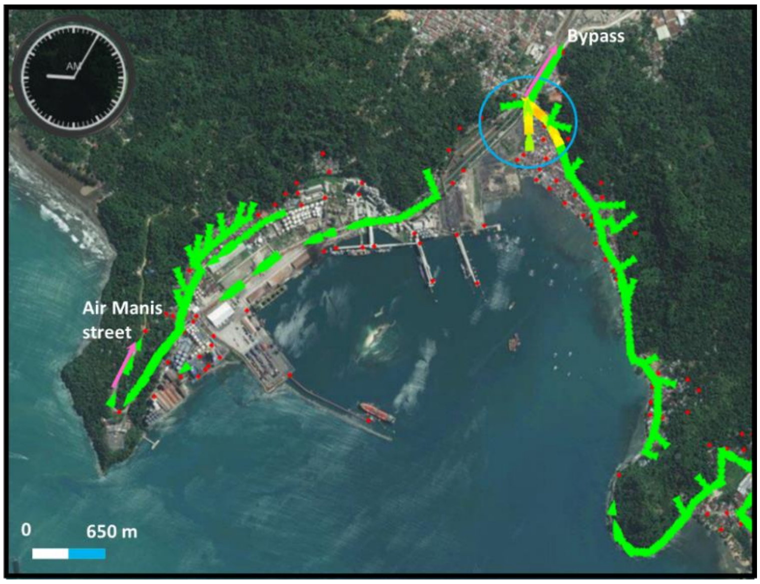

Figure 20. Congested areas from the 2P existing case simulation.

Moreover, with a total population of ~9,000 people, the two evacuation points might be insufficient to gather all the evacuee. Therefore, extra evacuation points are further opened as additional routes for evacuation (see the red dots in Figure 19B). Subsequently, two cases of tsunami evacuation models are presented here: (1) 2-point tsunami evacuation case (stationary case) to the existing routes and (2) 2-point tsunami evacuation case rerouting to the extra evacuation points time-dependently (time-variant case). The stationary case is indexed with a prefix of 'EP_SC', whilst the time-variant case is prefixed with 'EP_TV'. Moreover, all files containing only the prefix of only 'EP' belong to those two cases (i.e. such files are needed to run the 'EP_SC' and 'EP_TV' cases). For both case, the two existing evacuation points are opened at the beginning of evacuation (i.e. 09.00 am), whilst the opening time of the extra evacuation points are at 09.05 am (see Figure 20). In other words, agents are first assumed to follow the local authorities' suggestion then moved to the extra evacuation points if the congestion occurrs during the evacuation. The extra evacuation points are defined as nodes located outside the inundation areas and connected to the existing links (see the red dots in Figure 19B). To define the opening time of the extra evacuation points, the stationary case is firstly running. The congestion location and time produced from the benchmark simulation are further used for the time-variant case as shown in Figure 20. This time-variant case is essential to illustrate how time-dependent evacuation modelling is developed in MATSim.

5.3. MATLAB codes

Two sets of MATLAB codes are developed to run the tsunami evacuation modelling using MATSim for two purposes: (1) input data development and (2) post-processing analysis. The codes for input data development aim to produce XML files for configuration, plans, networks, and vehicles set up. Moreover, the codes for post-processing are used to extract the results of MATSim simulation and to calculate the risk in terms of affected people (i.e. number of people affected by the tsunami).

5.3.1. MATLAB codes for input data

To generate XML files for input data, the analyst needs to prepare essential data about network and agents. Table 1 lists required data for nodes, links, and agents to develop the network and plans XML files.

Table 1. Required information to develop the network and plans XML files.

Network parameter	Data

Nodes	• ID of nodes • Coordinates of each node in the Universal Transverse Mercator (UTM) system (in longitude and latitude directions).
Links	• ID of links • Number of lanes • Flow speed in m/s • Width in m • ID of entry node indicating where the link starts. • ID of exit node indicating where the link ends. • Length of the link in m • Free speed of the link in m/s • Maximum flow capacity in $persons/m/s$ (the period of 1 second can be changed based on the analyst purpose, e.g. it can be set to one hour which shows the number of agents in one hour) • Storage capacity in $persons/m^2$.
Agents	• ID of each agent • Coordinates of each agent in the UTM system • ID of the start link which is defined as the closest link to the agent • ID of the evacuation node taken from the virtual evacuation nodes. • Coordinates of virtual evacuation nodes in the UTM system.

a. Network file

The main program for developing network files is 'XML_network.m'. An extra MATLAB file 'links_and_nodes.mat' is needed to run the 'XML_network.m'. The extra file store all data related to links and nodes data in Teluk Bayur, Padang areas. Three sections of codes are written to generate the network files. The first section is to define the xml file name: network and to connect with the MATSim system that links to a specific website: http://www.matsim.org/files/dtd/network_v1.dtd. The simulation cannot be run without establishing such a link. Technically, in MATLAB, such a process is used to define the Document Object Model (DOM) node object and main root element (i.e. network) as shown in the following:

```
docNode = com.mathworks.xml.XMLUtils.createDocument('network');
domImpl = docNode.getImplementation();
doctype = domImpl.createDocumentType('network', 'SYSTEM',
'http://www.matsim.org/files/dtd/network_v1.dtd');
docNode.appendChild(doctype);
```

The second section is to define the properties of nodes. `'nodes'` is initially defined as the second layer of root belongs to `'network'` as the main root element:

```
root = docNode.createElement('nodes');
docNode.getDocumentElement.appendChild(root);
```

ID and coordinates (X and Y) of each node are further written which belong to the second root (`'nodes'`). To write those properties, an element of `'node'` representing each node is written. Subsequently, the attributes of each node: ID and X and Y coordinates are then coded.

```
child1 = docNode.createElement('node');
child1.setAttribute('id',sprintf('%i',i));
root.appendChild(child1);
temp1 = nodes(i,2);
child1.setAttribute('x',sprintf('%i',temp1));
root.appendChild(child1);
temp1 = nodes(i,3);
child1.setAttribute('y',sprintf('%i',temp1));
root.appendChild(child1);
```

The section of links is finally generated. `'links'`is also written as the second level of root below the main root element. Three parameters are also set up in the properties of the link including the capperiod, effectivecellsize, and effectivelanewidth. The capperiod is the period of the maximum flow capacity within the link, whilst the effectivecellsize and effectivelanewidth are to define the size of the mode (in $meter$). In the following example, the motorbike mode is used for illustration with the mode size of 2.5 m (length) and 1.0 m (width).

```
root1 = docNode.createElement('links');
docNode.getDocumentElement.appendChild(root1);
    root1.setAttribute('capperiod',sprintf('01:00:00'))
    root1.setAttribute('effectivecellsize',sprintf('2.5'));
    root1.setAttribute('effectivelanewidth',sprintf('1.0'));
```

The properties of each link are further coded in `'links'` including ID (`'id'`), type of mode (`'modes'`), number of lanes (`'permlanes'`), capacity (`'capacity'`), freespeed of the link (`'freespeed'`), length of the link (`'length'`), ID of entry node (`'from'`), and ID of exit node (`'to'`). The `'car'` mode is written in the following code since it is a default mode of MATSim. However, changing the size of the mode can be implemented through the vehicles setup.

```
child1 = docNode.createElement('link');
    child1.setAttribute('id',sprintf('%i',i));
    root1.appendChild(child1);
    child1.setAttribute('modes','car');
    root1.appendChild(child1);
    temp1 = links(i,2);
    child1.setAttribute('permlanes',sprintf('%i',temp1));
    root1.appendChild(child1);
    temp1 = links(i,10);
    child1.setAttribute('capacity',sprintf('%i',temp1));
    root1.appendChild(child1);
    temp1 = links(i,3);
    child1.setAttribute('freespeed',sprintf('%i',temp1));
    root1.appendChild(child1);
    temp1 = links(i,9);
    child1.setAttribute('length',sprintf('%i',temp1));
    root1.appendChild(child1);
    temp1 = links(i,5);
    child1.setAttribute('from',sprintf('%i',temp1));
    root1.appendChild(child1);
     temp1 = links(i,6);
    child1.setAttribute('to',sprintf('%i',temp1));
    root1.appendChild(child1);
```

b. Plans file

The main program for developing network files is 'XML_plans.m'. Two extra MATLAB files , i.e. including 'links_and_nodes.mat', 'Agents_attributes.mat', and 'index_SafeAgents.mat' are necessary to run the 'XML_plans.m'. The 'Agents_attributes.mat' records the coordinates of all agents, whilst 'index_SafeAgents.mat' contains the index of safe agents that are located outside inundated zone. To develop the plans file, the main root element (`'plans'`) and the website link to MATSim (http://www.matsim.org/files/dtd/plans_v4.dtd) are firstly introduced.

```
docNode = com.mathworks.xml.XMLUtils.createDocument('plans');
domImpl = docNode.getImplementation();
doctype = domImpl.createDocumentType('plans', 'SYSTEM',
'http://www.matsim.org/files/dtd/plans_v4.dtd');
docNode.appendChild(doctype);
```

`'person'` is the only second level of root to represent the agent.

```
root = docNode.createElement('person');
docNode.getDocumentElement.appendChild(root);
root.setAttribute('id',sprintf('%i',i));
    child1 = docNode.createElement('plan');
    root.appendChild(child1);
```

Plans of each agent are then defined below the `'person'` and consist of two main actions: the agent moves from home (`'type','h'`) to the evacuation point (work; `'type','w'`) and then come back to their home (`'type','h'`). The last plan is used because the default plans setup of MATSim requeires the agent to go back on their initial location. Two type of attributes need to be written below the plan including the action (`'act'`) and the leg mode (`'leg'`). In the attribute of `'act'`, the information of action name (e.g. `'end_time'`), the initial link to implement such an action (`'link',sprintf('%i',LINK_H)`), the coordinates of the actions (`'x'` and `'y'`), and the type of the action (e.g. `'type','h'`) are coded.

```
%%%%%%%%%%%%% ACT MODE 1

child2 = docNode.createElement('act');
child1.appendChild(child2);
child2.setAttribute('end_time',tempTime);
child1.appendChild(child2);
child2.setAttribute('link',sprintf('%i',LINK_H));
child1.appendChild(child2);
child2.setAttribute('x',sprintf('%i',tempx));
child1.appendChild(child2);
child2.setAttribute('y',sprintf('%i',tempy));
child1.appendChild(child2);

child2.setAttribute('type','h');
child1.appendChild(child2);
```

```
%%%%%%%%%%%%% LEG MODE 1

child2 = docNode.createElement('leg');
child1.appendChild(child2);
child2.setAttribute('mode','car');
child1.appendChild(child2);
child3 = docNode.createElement('route');
child2.appendChild(child3);
routes = docNode.createTextNode(sprintf('%i',ROUTE));
child3.appendChild(routes);
```

c. Vehicles file

An extra input file is also prepared to perform the simulation: vehicle setup. The 'index_SafeAgents.mat' is also used to exclude the safe agents located outside inundated areas. The vehicle file is essential to change the mode parameters because the default mode in MATSim is car with size of 7.5 m in length and 3.5 m in width. The main root element is initially written along with its attributes that explain the website linked to the MATSim package as shown in the following:

```
docNode =
com.mathworks.xml.XMLUtils.createDocument('vehicleDefinitions');
toc = docNode.getDocumentElement;
toc.setAttribute('xmlns','http://www.matsim.org/files/dtd');
toc.setAttribute('xmlns:xsi', 'http://www.w3.org/2001/XMLSchema-
instance');
toc.setAttribute('xsi:schemaLocation','http://www.matsim.org/file
s/dtd
http://www.matsim.org/files/dtd/vehicleDefinitions_v1.0.xsd');
```

The first second-level of root: vehicle type (`'vehicleType'`) is then coded to create a default vehicle type that is used for all agents. The attributes of each vehicle including the capacity (number of seats: `'capacity'`), length (`'length'`), width (`'width'`), and maximum velocity (`'maximumVelocity'`) are also written under the vehicle type.

```
root1 = docNode.createElement('vehicleType');
docNode.getDocumentElement.appendChild(root1);
    root1.setAttribute('id',sprintf('defaultVehicleType'))
     docNode.getDocumentElement.appendChild(root1);
      child1 = docNode.createElement('capacity');
      root1.appendChild(child1);
            child2 = docNode.createElement('seats');
            child2.setAttribute('persons',sprintf('1'))
            child1.appendChild(child2);
      child1 = docNode.createElement('length');
      child1.setAttribute('meter',sprintf('2.5'))
      root1.appendChild(child1);
      child1 = docNode.createElement('width');
      child1.setAttribute('meter',sprintf('1.0'))
      root1.appendChild(child1);
```

```
child1 = docNode.createElement('maximumVelocity');
child1.setAttribute('meterPerSecond',sprintf('5.66'))
root1.appendChild(child1);
```

Finally, the last second-level of root (`'vehicle'`) is introduced to define the vehicle ID (`'id'`) and their type (`'defaultVehicleType'`).

```
root1 = docNode.createElement('vehicle');
docNode.getDocumentElement.appendChild(root1);
root1.setAttribute('id',sprintf('%i', id))
docNode.getDocumentElement.appendChild(root1);
root1.setAttribute('type',sprintf('defaultVehicleType'))
docNode.getDocumentElement.appendChild(root1);
```

d. Configuration file

The confuiguration file contains the key information for MATSim simulations since it is used to run the evacuation simulation via MATLAB. Two main programs are created for two considered cases: (1) time-varying network model and (2) stationary network model. The file names of the two models are 'XML_config_timeVariant_network.m' and 'XML_config_stationary_network.m', respectively. The configuration file contains all the input parameters and files including the main parameters of evacuation simulations (e.g. number of iteration and scoring) and input files (i.e. newtrok, plans, and vehicles). The main root element and the website information belongs to the MATSim system is firstly written.

```
docNode = com.mathworks.xml.XMLUtils.createDocument('config');
domImpl = docNode.getImplementation();
doctype = domImpl.createDocumentType('config', 'SYSTEM',
'http://www.matsim.org/files/dtd/config_v2.dtd');
docNode.appendChild(doctype);
```

The modules of all input files, i.e. network, plans, and vehicles, are then coded.

```
root = docNode.createElement('module');
    docNode.getDocumentElement.appendChild(root);
    root.setAttribute('name',sprintf('network'));
        child2 = docNode.createElement('param');
        root.appendChild(child2);

child2.setAttribute('name',sprintf('inputNetworkFile'))
            root.appendChild(child2);
            filenetwork = sprintf(['MOTORBIKE_network_SCE3_M'
num2str(MAG)  '_Model_' num2str(STOCH_MODEL(MODEL)) '.xml']);
            child2.setAttribute('value',filenetwork);
            root.appendChild(child2);
    root = docNode.createElement('module');
    docNode.getDocumentElement.appendChild(root);
    root.setAttribute('name',sprintf('plans'));
        child2 = docNode.createElement('param');
        root.appendChild(child2);
```

```
                    child2.setAttribute('name',sprintf('inputPlansFile'))
                    root.appendChild(child2);
                    fileplan = sprintf(['MOTORBIKE_plans_SCE3_M'
num2str(MAG)  '_Model_' num2str(STOCH_MODEL(MODEL)) '.xml']);
                    child2.setAttribute('value',fileplan);
                    root.appendChild(child2);
   root = docNode.createElement('module');
    docNode.getDocumentElement.appendChild(root);
    root.setAttribute('name',sprintf('vehicles'));
        child2 = docNode.createElement('param');
        root.appendChild(child2);
                    child2.setAttribute('name',sprintf('vehiclesFile'))
                    root.appendChild(child2);
                    filevehicle = sprintf(['MOTORBIKE_vehicleType_SCE3_M'
num2str(MAG)  '_Model_' num2str(STOCH_MODEL(MODEL)) '.xml']);
                    child2.setAttribute('value',filevehicle);
                    root.appendChild(child2);
```

Finally, other modules to define the parameters, such as the number of iteration, flow traffic modelling, and scoring paremeters, are written (see the main program for the details).

To consider the time-variant network model, the analyst needs to create an extra xml file to define the variation of network opening time. In principle, the network can be updated by changing the free speed or the capacity of the link (Horni et al., 2016). Subsequently, the extra xml file containing network changes is connected through the configuration file under the module of `'network'`. For illustration, Figure 21 shows how the time-variant network model is implemented. A prameter name `'timeVariantNetwork'` should be set up to the value of `'true'` and is connected with the file name of `'timeVariant_network.xml'`. Moreover, Figure 22 illustrates an example of network changed file (i.e. `'timeVariant_network.xml'`). The free speed value changes to zero meaning that the considered link (i.e. link number 612, 613, and 614) can not be used anymore starting from 09.10 am (see the time frame in Figure 22). On the other hand, in this manual, the stationary case also needs the extra xml file (i.e. 'stationary_network.xml'). The 'stationary_network.xml' sets up the extra evacuation points to have zero flow velocity before evacuation is started (i.e. before 09.00 am) meaning that those extra evacuation points are not available for evacuation.

```
  <?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v2.dtd"
  PUBLIC "SYSTEM">
- <config>
  - <module name="network">
      <param name="inputNetworkFile"
        value="FOOT_network_SCE3_M90_Model_2.xml"/>
      <param name="timeVariantNetwork" value="true"/>
      <param name="inputChangeEventsFile"
        value="Existing_change.xml"/>
    </module>
```

Figure 21. An example of including time-variant network model within the configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
- <networkChangeEvents
  xsi:schemaLocation="http://www.matsim.org/files/dtd
  http://www.matsim.org/files/dtd/networkChangeEvents.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.matsim.org/files/dtd">
  - <networkChangeEvent startTime="09:10:00">
      <link refId="612"/>
      <link refId="613"/>
      <link refId="614"/>
      <freespeed value="0.0" type="absolute"/>
    </networkChangeEvent>
  </networkChangeEvents>
```

Figure 22. An example of changed network xml file.

Interest reader may try the codes and run the XML file in MATSim. Output files (i.e. 'EP_SC_output_events.xml' and 'EP_TV_output_events.xml') are also included at the shared directory to check the results.

5.3.2. MATLAB codes for pot-processing

Three main programs are developed for post-processing purposes and stored at the shared directory. They can be improved based on the analyst's needs. The three programs are for: (1) extraction of simulation results ('XML_results_extraction.m'), (2) calculation of evacuation time ('EVA_time_calculation.m'), and (3) calculation of affected people ('AFFECTED_people_calculation.m'). To extract the simulation results (XML file), the analyst needs to define the total number of agents (i.e. 'NumAgent') and safe agents (i.e. 'Num_of_safe_agents'). The output of the simulation results is saved in 'RAW_results.mat' containing all the agents' final plans.

Subsequently, using the 'RAW_results.mat' file, the evacuation time can be calculated. Moreover, extra files including 'index_SafeAgents.mat' and 'Agents_attributes.mat' are also needed to calculate the evacuation time of each agent. Before running the code (i.e. 'EVA_time_calculation.m'), the following parameters need to be set up by the analyst:

- 'TotalofAgent': the total number of agents in the simulation
- 'Starting_Time': time refers to the starting time of evacuation (in seconds)
- 'IRT': the initial reaction time of the community; in general, it is between 10-15 minutes (Di Mauro et al., 2013; Muhammad et al., 2017).
- 'Time_Conversion': a conversion parameter from minutes to second (i.e. 60 seconds)

The output is recorded in file 'EVA_time.mat' writing the evacuation time (in minutes) of each agent.

Finally, the number of affected people is calculated. The main data for this calculation is 'RAW_results.mat' file. However, extra thre MATLAB files are needed to calculated the affected people: 'index_SafeAgents.mat', 'Tsunami_Arrival_Time.mat', and 'link_and_nodes.mat'. The 'Tsunami_Arrival_Time.mat' is essential to define the tsunami arrival time at each node, whilst 'link_and_nodes.mat' to extract the attributes of links and nodes (i.e. coordinates in UTM). The user needs to define the following parameters before running the main program:

- 'TotalofAgent': the total number of agents in the simulation
- 'EvaNode': the ID of final evacuation node
- 'IRT': the initial reaction time of the community
- 'Initial_Time': the starting time of evacuation
- 'Time_Conversion': 60 seconds (by default)
- 'Time_Non_Inundated': 6,000 seconds (1,000 minutes). A 6,000 seconds is adopted because it is much longer than the inundation period (120 minutes).

The output of this calculation is recorded in file 'Number_AFFECTED_people.mat' containing the number of affected people.

6. Conclusion and future extensions

An introduction on tsunami evacuation modelling and a testing of MATSim package have been conducted in this study. Two cases: a simple case study and a tsunami evacuation modelling in Padang, Indonesia have been presented in this manual. MATLAB codes to develop input files and post-processing analysis are also described. Future extensions may include more complex cases that are important for developing effective tsunami evacuation plans.

References

Affan M et al. (2012). Tsunami evacuation simulation for disaster awareness education and mitigation planning of Banda Aceh. In Proceeding of the 15th World Conference on Earthquake Engineering, Lisbon, Portugal.

Axhausen, K. W., Nagel, K., & Horni, A. (2016). The Multi-Agent Transport Simulation MATSim.

Balmer, M., Rieser, M., Meister, K., Charypar, D., Lefebvre, N., & Nagel, K. (2009). MATSim-T: Architecture and simulation times. In Multi-agent systems for traffic and transportation engineering (pp. 57-78). IGI Global.

Badan Pusat Statistik. (2018). *Laporan Data Penduduk Kota Padang: Januari.* BPS Sumatera Barat. Padang.

Boen, T. 2005. The northern Sumatra earthquake of March 28, 2005. EERI Special Earthquake Report, August 2005, pp. 2-8.

Borrero J. C, et al. (2006). Tsunami Inundation Modeling for Western Sumatra. *Proceeding of the National Academy of Sciences of the United States of America* 103(52), 19673–19677.

Chalmet, L. G., Francis, R. L., & Saunders, P. B. (1982). Network models for building evacuation. Fire Technology, 18(1), 90-113.

Chen, X., & Zhan, F. B. (2008). Agent-based modelling and simulation of urban evacuation: relative effectiveness of simultaneous and staged evacuation strategies. Journal of the Operational Research Society, 59(1), 25-33.

Daly P, et al. (2019). Archaeological evidence that a late 14th-century tsunami devastated the coast of northern Sumatra and redirected history. *Proceedings of the National Academy of Sciences*, *116*(24), 11679-11686.

De Risi, R., & Goda, K. (2016). Probabilistic earthquake–tsunami multi-hazard analysis: application to the Tohoku region, Japan. Frontiers in Built Environment, 2, 25.

Di Mauro M, et al. (2013). Tsunami risk reduction for densely populated Southeast Asian cities: analysis of vehicular and pedestrian evacuation for the city of Padang, Indonesia, and assessment of interventions. *Natural hazards*, 68(2), 373-404.

Fellendorf, M., & Vortisch, P. (2001). Validation of the microscopic traffic flow model VISSIM in different real-world situations. In Transportation Research Board 80th Annual Meeting.

Fukutani, Y., Suppasri, A., & Imamura, F. (2015). Probabilistic analysis and uncertainty assessment of tsunami wave height using a random source parameter model that targets a Tohoku-type earthquake fault. Probabilistic Environmental Research and Risk Assessment, 29(7), 1763-1779.

Goda, K., & De Risi, R. (2018). Multi-hazard loss estimation for shaking and tsunami using probabilistic rupture sources. International journal of disaster risk reduction, 28, 539-554.

Goetz, M., & Zipf, A. (2012). Using crowdsourced geodata for agent-based indoor evacuation simulations. ISPRS International Journal of Geo-Information, 1(2), 186-208.

Griffin, J. D., Pranantyo, I. R., Kongko, W., Haunan, A., Robiana, R., Miller, V., et al. (2016). "Assessing tsunami hazard using heterogeneous slip models in the Mentawai Islands, Indonesia," in Geohazards in Indonesia: Earth Science for Disaster Risk Reduction, eds P. R. Cummins and I. Meilano (London: Geological Society). Special Publications, 441.

Grezio, A., Babeyko, A., Baptista, M. A., Behrens, J., Costa, A., Davies, G., ... & Harbitz, C. B. (2017). Probabilistic tsunami hazard analysis: Multiple sources and global applications. Reviews of Geophysics, 55(4), 1158-1198.

Han, L. D., Yuan, F., & Urbanik, T. (2007). What is an effective evacuation operation?. Journal of urban planning and development, 133(1), 3-8.

Helbing, D., & Tilch, B. (1998). Generalized force model of traffic dynamics. Physical review E, 58(1), 133.

Horni, A., Nagel, K., & Axhausen, K. W. (Eds.). (2016). The multi-agent transport simulation MATSim (p. 618). London: Ubiquity Press.

Jha, M., Moore, K., & Pashaie, B. (2004). Emergency evacuation planning with microscopic traffic simulation. Transportation Research Record, 1886(1), 40-48.

Kemal B. M, Putra H (2018). An observation of the walking speed of evacuees during a simulated tsunami evacuation in Padang, Indonesia. In IOP Conference Series: Earth and Environmental Science, 140(1), 1-9.

Kerner, B. S., & Klenov, S. L. (2002). A microscopic model for phase transitions in traffic flow. Journal of Physics A: Mathematical and General, 35(3), L31.

Krauß, S., Wagner, P., & Gawron, C. (1997). Metastable states in a microscopic model of traffic flow. Physical Review E, 55(5), 5597.

Lämmel, G., Rieser, M., Nagel, K., Taubenböck, H., Strunz, G., Goseberg, N., & Birkmann, J. (2010). Emergency preparedness in the case of a tsunami—Evacuation analysis and traffic optimization for the Indonesian city of Padang. In Pedestrian and Evacuation Dynamics 2008 (pp. 171-182). Springer, Berlin, Heidelberg.

Lämmel, G., Klüpfel, H., & Nagel, K. (2009). The MATSim network flow model for traffic simulation adapted to large-scale emergency egress and an application to the evacuation of the Indonesian city of Padang in case of a tsunami warning. In Pedestrian Behavior: Models, Data Collection and Applications (pp. 245-265). Emerald Group Publishing Limited.

Makinoshima, F., Imamura, F., & Abe, Y. (2018). Enhancing a tsunami evacuation simulation for a multi-scenario analysis using parallel computing. Simulation Modelling Practice and Theory, 83, 36-50.

Maerivoet, S., & De Moor, B. (2005). Cellular automata models of road traffic. Physics reports, 419(1), 1-64.

Mas, E., Suppasri, A., Imamura, F., & Koshimura, S. (2012). Agent-based simulation of the 2011 great east japan earthquake/tsunami evacuation: An integrated model of tsunami inundation and evacuation. Journal of Natural Disaster Science, 34(1), 41-57.

Mas, Erick, Shunichi Koshimura, Fumihiko Imamura, Anawat Suppasri, Abdul Muhari, and Bruno Adriano. "Recent advances in agent-based tsunami evacuation simulations: Case studies in Indonesia, Thailand, Japan and Peru." Pure and Applied Geophysics 172, no. 12 (2015): 3409-3424.

Mostafizi, A., Wang, H., Cox, D., Cramer, L. A., & Dong, S. (2017). Agent-based tsunami evacuation modeling of unplanned network disruptions for evidence-driven resource allocation and retrofitting strategies. Natural Hazards, 88(3), 1347-1372.

Muhammad, A., Goda, K., Alexander, N. A., Kongko, W., & Muhari, A. (2017). Tsunami evacuation plans for future megathrust earthquakes in Padang, Indonesia, considering probabilistic earthquake scenarios. Natural Hazards and Earth System Sciences, 17(12), 2245.

Muhammad, A., & Goda, K. (2018). Impact of earthquake source complexity and land elevation data resolution on tsunami hazard assessment and fatality estimation. Computers & geosciences, 112, 83-100.

Nagel, K., & Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. Journal de physique I, 2(12), 2221-2229.

Nalbant S S, et al. (2005). Seismology: Earthquake risk on the Sunda trench. *Nature*, 435(7043), 756, doi: 10.1038/nature435756a.

Natawidjaja D H, et al. (2006). Source parameters of the great Sumatran megathrust earthquakes of 1797 and 1833 inferred from coral microatolls. *Journal of Geophysical Research: Solid Earth*, 111(B6), doi:10.1029/2005JB004025.

González-Riancho, P., Aguirre-Ayerbe, I., Aniel-Quiroga, I., Abad, S., González, M., Larreynaga, J., & Medina, R. (2013). Tsunami evacuation modelling as a tool for risk reduction: application to the coastal area of El Salvador. Natural Hazards and Earth System Sciences, 13(12), 3249.

Rossetto, T., Peiris, N., Pomonis, A. et al. The Indian Ocean tsunami of December 26, 2004: observations in Sri Lanka and Thailand, Nat Hazards (2007) 42: 105. https://doi.org/10.1007/s11069-006-9064-3

Taubenböck, H., Goseberg, N., Setiadi, N., Lämmel, G., Moder, F., Oczipka, M., ... & Birkmann, J. (2009). " Last-Mile" preparation for a potential disaster-Interdisciplinary approach towards tsunami early warning and an evacuation information system for the coastal city of Padang, Indonesia. Natural Hazards and Earth System Science 9 (2009), Nr. 4, 9(4), 1509-1528.

Tobler W (1993) Three presentations on geographical analysis and modeling—Nonisotropic geographic modeling. Speculations on the geometry of geography; and global spatial analysis. UCSB. National Center for Geographic Information and Analysis Technical Report 93-1. Available at www.ncgia.ucsb.edu/Publications/Tech_Reports/93/93-1. PDF. Accessed June 1, 2019

Wang, H., Mostafizi, A., Cramer, L. A., Cox, D., & Park, H. (2016). An agent-based model of a multimodal near-field tsunami evacuation: Decision-making and life safety. Transportation Research Part C: Emerging Technologies, 64, 86-100.

Weidmann, U. (1993). Transporttechnik der fußgänger: transporttechnische eigenschaften des fußgängerverkehrs, literaturauswertung. IVT Schriftenreihe, 90.

Zarboutis, N., & Marmaras, N. (2007). Design of formative evacuation plans using agent-based simulation. Safety Science, 45(9), 920-940.