Applying MATSim to Estimating Demand for Parking Places at Rest Areas

Paweł Gora
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw

Work within a project
“Parking Places at Rest Areas”

for
General Directorate of National Roads and Motorways in Poland

in cooperation with
Warsaw University of Technology
Motor Transport Institute

MATSim User Meeting 2017
September 11, 2017, Technion Campus, Haifa, Israel

Applying MATSim to Estimating Demand for Parking Places at Rest Areas & Approximating Simulations Using Machine Learning

Paweł Gora
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw

Work within a project
"Parking Places at Rest Areas"

for
General Directorate of National Roads and Motorways in Poland

in cooperation with
Warsaw University of Technology
Motor Transport Institute

MATSim User Meeting 2017
September 11, 2017, Technion Campus, Haifa, Israel

Rest area

“Public facility, located next to a large thoroughfare such as a highway, expressway, or freeway at which drivers and passengers can rest, eat, or refuel without exiting onto secondary roads.“

Professional drivers are obliged by law to rest during long trips, they usually stop at rest areas.

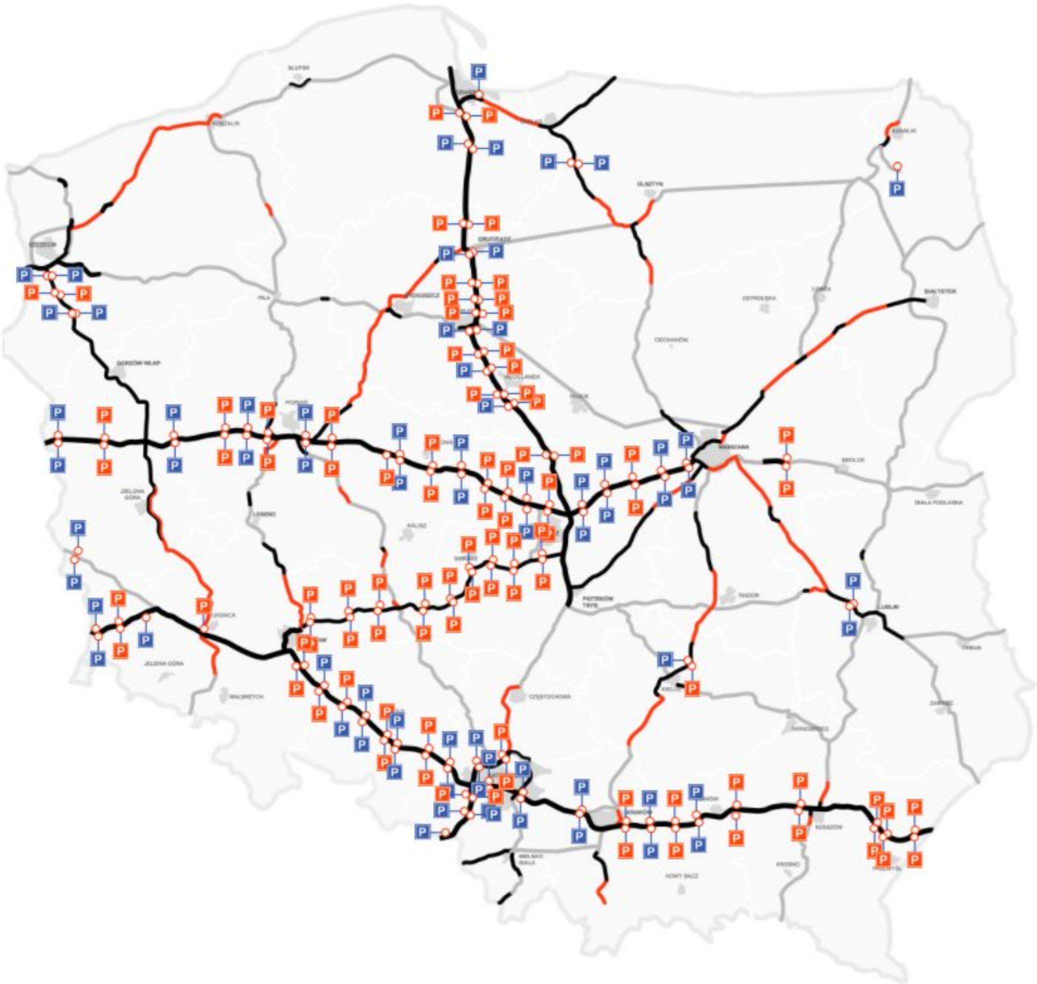
Problem:

How to estimate number of required parking places?

Too many places -> waste of land and money

Too few spaces -> shortages of parking places (which often happens in many countries)

Location of rest areas in Poland



Existing methodologies

N - number of required parking places

ADT - average daily traffic

$$N = f(ADT)$$

Existing methodologies

Poland (1997):

$P = C1 * C2 * C3 * C4 * ADT / 2$

$N = P + 2.5 * \mathrm{sqrt}(P)$ (number of required parking places for 20km)

C1 - percentage of trucks / private cars

C2 - ratio of traffic flow during noon hours (12:00-13:00) to daily traffic

C3 - ratio of cars which park to all cars on road (during noon hours)

C4 - average time of parking

Existing methodologies

Poland (1997):

$P = C1 * C2 * C3 * C4 * ADT / 2$

$N = P + 2.5 * \mathrm{sqrt}(P)$ (number of required parking places for 20km)

C1 - percentage of trucks / private cars

C2 - ratio of traffic flow during noon hours (12:00-13:00) to daily traffic

C3 - ratio of cars which park to all cars on road (during noon hours)

C4 - average time of parking

Modifications with regard to type of rest area, facility, proximity to cities/ borders etc.

Existing methodologies

Poland (1997):

P = C1 * C2 * C3 * C4 * ADT / 2

N = P + 2.5 * sqrt(P)   (number of required parking places for 20km)

C1 - percentage of trucks / private cars

C2 - ratio of traffic flow during noon hours (12:00-13:00) to daily traffic

C3 - ratio of cars which park to all cars on road (during noon hours)

C4 - average time of parking

Modifications with regard to type of rest area, facility, proximity to cities/ borders etc.

Similar analytical methods are used in other countries (Germany, USA, UK).

Problems with existing methodologies

- Need a lot of data from measurements
- Equations are general and correspond to “average cases”
- Don’t take into account network topology
- May be not accurate for atypical areas where driver’s behavior is far from average

Applying MATSim

I am working on his topic with a student from the University of Warsaw, Piotr Szefler.

Goal: run simulations to compute occupation of parkings in time (MOPSim)

(help in validating methodologies or as a methodology for new highways)

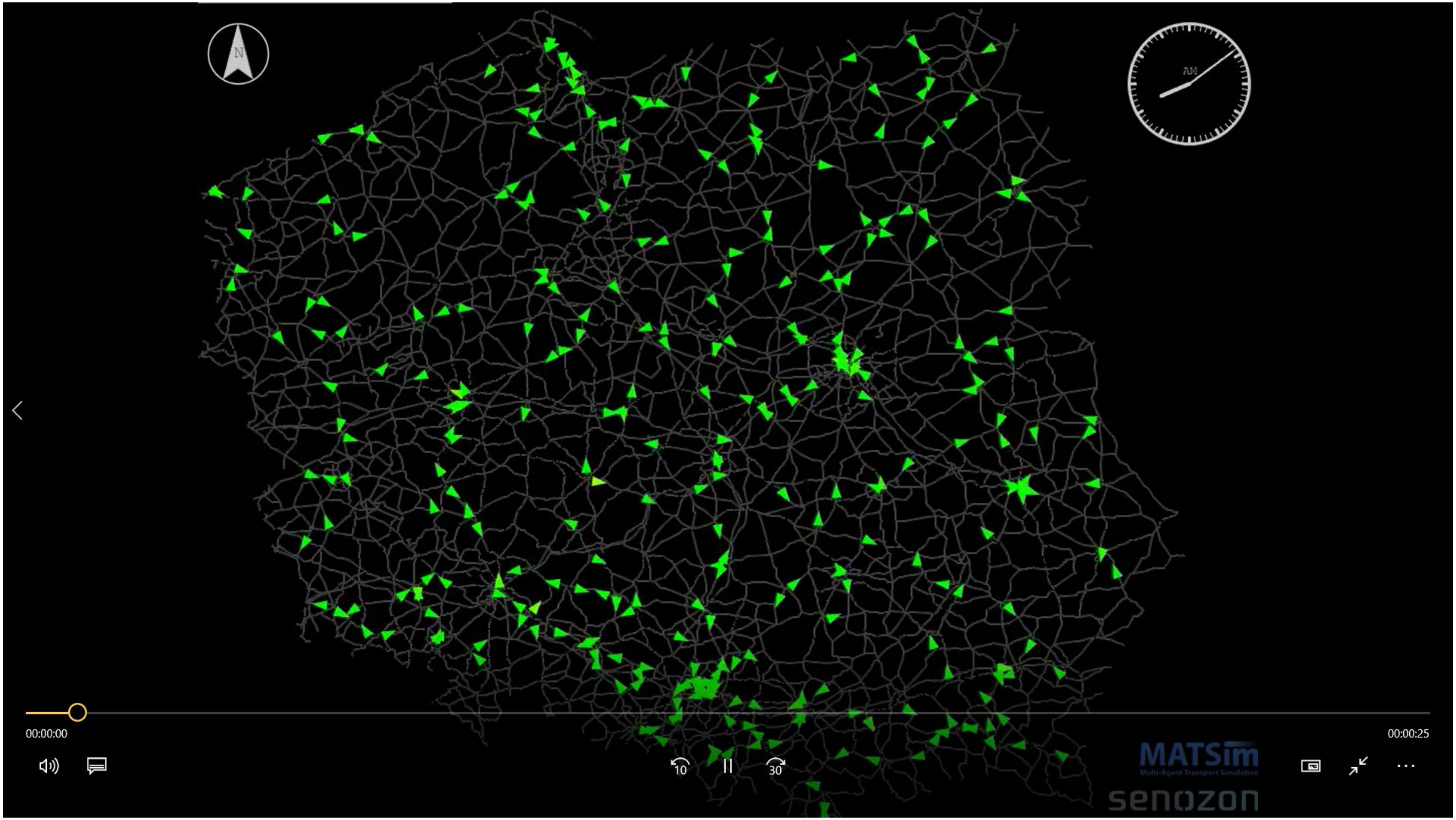
Data:

- Road networks structure -> OpenStreetMap
- Demand: OD matrices (2005, 2015, 2025 (estimated)) with respect to type of vehicle and purpose of trip (OD matrix for Poland: 380 internal regions + 89 border regions)
- Calibrating route assignment: traffic flow measurements
- Surveys and measurements (cameras) at rest areas
- (Potentially) travel times from Google Maps API and CE-Traffic

Applying MATSim

- Road network of Poland (from OpenStreetMap):
  - 3 main types of roads (motorway, trunk, primary from OSM)
  - 110120 nodes
  - 179515 edges
- Based on OD matrix we generated 604452 trips
- We added locations of 281 rest areas in Poland
- We are analyzing data from measurements to model the process of taking decisions regarding stopping at rest areas

Visualization in VIA



Next actions

- After proper calibration we can compute average / expected number of occupied parking places at rest areas in each time frame

Next actions

- After proper calibration we can compute average / expected number of occupied parking places at rest areas in each time frame
- In case of building new roads / rest areas this tool can be used to find optimal road network structure and optimal locations and capacities of rest areas:
  - we can define a metric to evaluate quality of a given setting
  - we can compute value of a metric using simulation
  - we can apply optimization algorithms (e.g., genetic algorithm) to find (sub)optimal setting

Next actions

- Simulations are time-consuming - in the rest area use case we don’t have to worry (those computations will be run “offline” and rarely), but we may think about meta-models
- If the goal of a simulation is to compute value of a given metric (e.g., maximal occupation of parking places) for a given setting of rest areas, then simulation computes the relation:

$$F: X \to Y$$

X - set of settings of rest areas (locations, capacities, types etc)

Y - possible values of a given metric (set of natural / real numbers, e.g., maximal occupancy, cost of building rest areas etc)

Next actions

Perhaps we can compute (or approximate) F using some meta-models, e.g., machine learning algorithms?

Universal Approximation Theorem (1989): for any continuous function F on a compact subset K of $\mathbf{R}^n$, there exists a feedforward neural network, having only a single hidden layer, which uniformly approximates F to within an arbitrary ε>0 on K.

Next actions

Perhaps we can compute (or approximate) F using some meta-models, e.g., machine learning algorithms?

Universal Approximation Theorem (1989): for any continuous function F on a compact subset K of $\mathbf{R}^n$, there exists a feedforward neural network, having only a single hidden layer, which uniformly approximates F to within an arbitrary ε>0 on K.

The theorem doesn’t say anything about number of neurons, size of a training set, training time etc.

Next actions

Perhaps we can compute (or approximate) F using some meta-models, e.g., machine learning algorithms?

Universal Approximation Theorem (1989): for any continuous function F on a compact subset K of $\mathbf{R}^n$, there exists a feedforward neural network, having only a single hidden layer, which uniformly approximates F to within an arbitrary ε>0 on K.

The theorem doesn’t say anything about number of neurons, size of a training set, training time etc.

But it suggests that neural networks are good universal approximators.

Next actions

Perhaps we can compute (or approximate) F using some meta-models, e.g., machine learning algorithms?

Universal Approximation Theorem (1989): for any continuous function F on a compact subset K of $\mathbf{R}^n$, there exists a feedforward neural network, having only a single hidden layer, which uniformly approximates F to within an arbitrary ε>0 on K.

The theorem doesn’t say anything about number of neurons, size of a training set, training time etc.

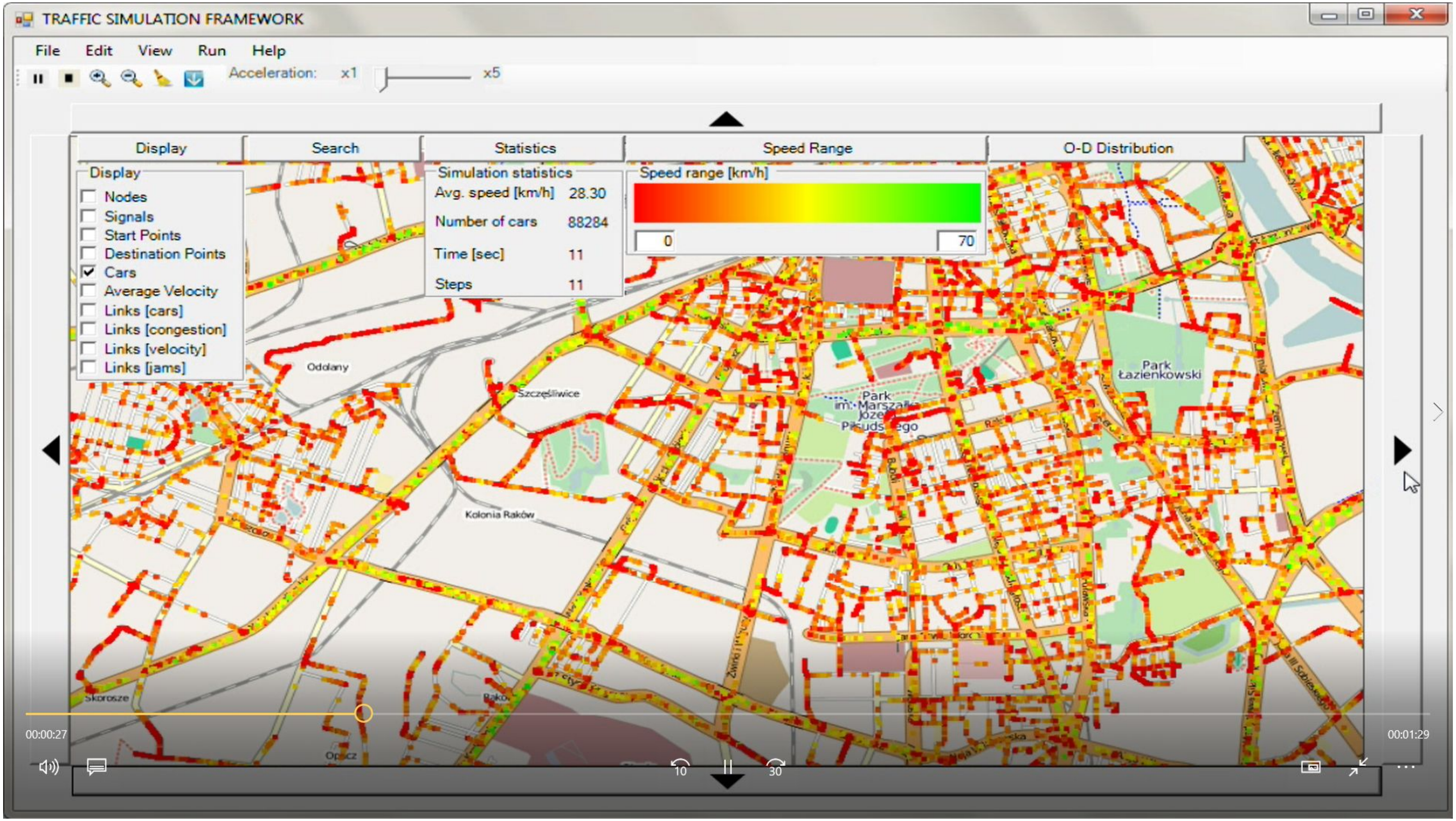
But it suggests that neural networks are good universal approximators.

Nowadays, we know that deep neural networks (with greater number of hidden layers) usually give better performance than neural networks with 1 hidden layer.

Traffic Simulation Framework

- Road network is a directed graph
- Traffic signals are located in some nodes
- Edges are divided into lanes, which are divided into cells
- Motion on edges is according to Na-Sch rules (cellular automaton) with some modification
- Lane changing
- Speed reduction before intersection / turning
- Different profiles of drivers
- Different types of roads (OSM data)

Traffic Simulation Framework



Display

- ☐ Nodes
- ☐ Signals
- ☐ Start Points
- ☐ Destination Points
- ☑ Cars
- ☐ Average Velocity
- ☐ Links [cars]
- ☐ Links [congestion]
- ☐ Links [velocity]
- ☐ Links [jams]

Simulation statistics

Avg. speed [km/h]	28.30
Number of cars	88284
Time [sec]	11
Steps	11

Speed range [km/h]

0 70

Approximating outcomes of traffic simulation using neural network

Using Traffic Simulation Framework I generated training set with more than 117 000 elements:

- input: traffic signal offsets on a small area in Warsaw (15 intersections)
- output: the total waiting time (speed = 0 km/h) during 10 minutes of traffic

I used this set to train neural network to approximate outcomes of simulations.

Results were very good.

TensorTraffic

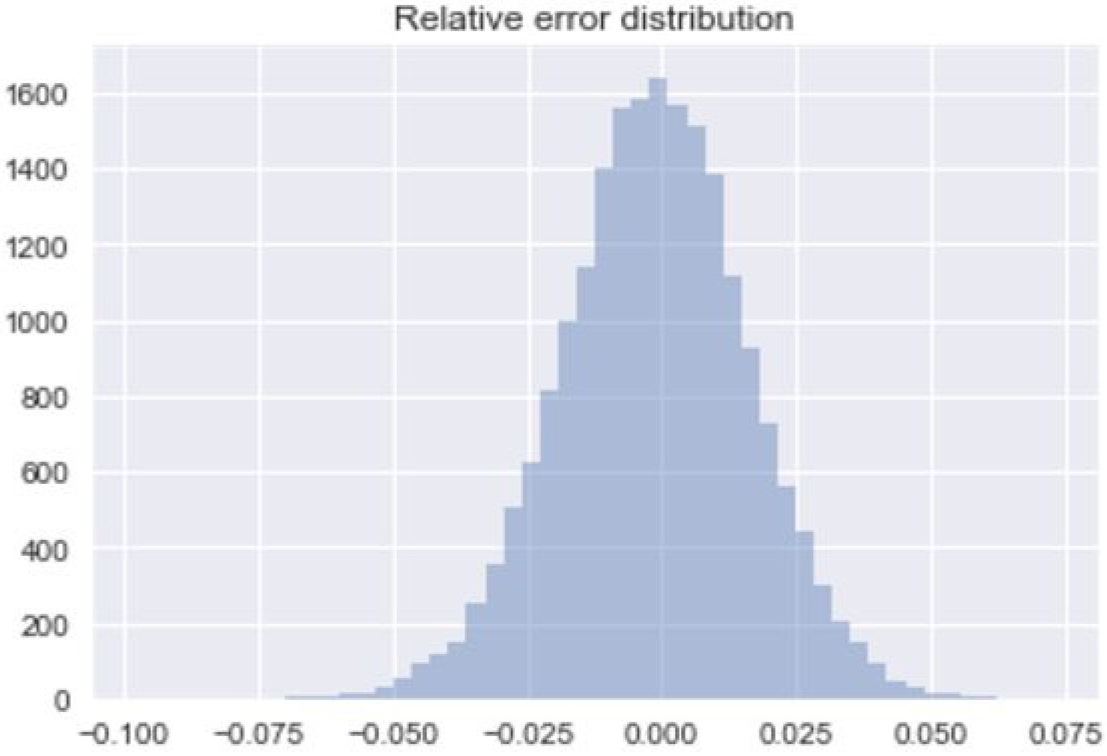
TensorFlow

Joint work with Karol Kurach (Google Research Zurich), Marek Bardoński (Nvidia Zurich) and other scientists (now I am collaborating with over 15 scientists from Poland, UK, Switzerland, Russia, Spain)

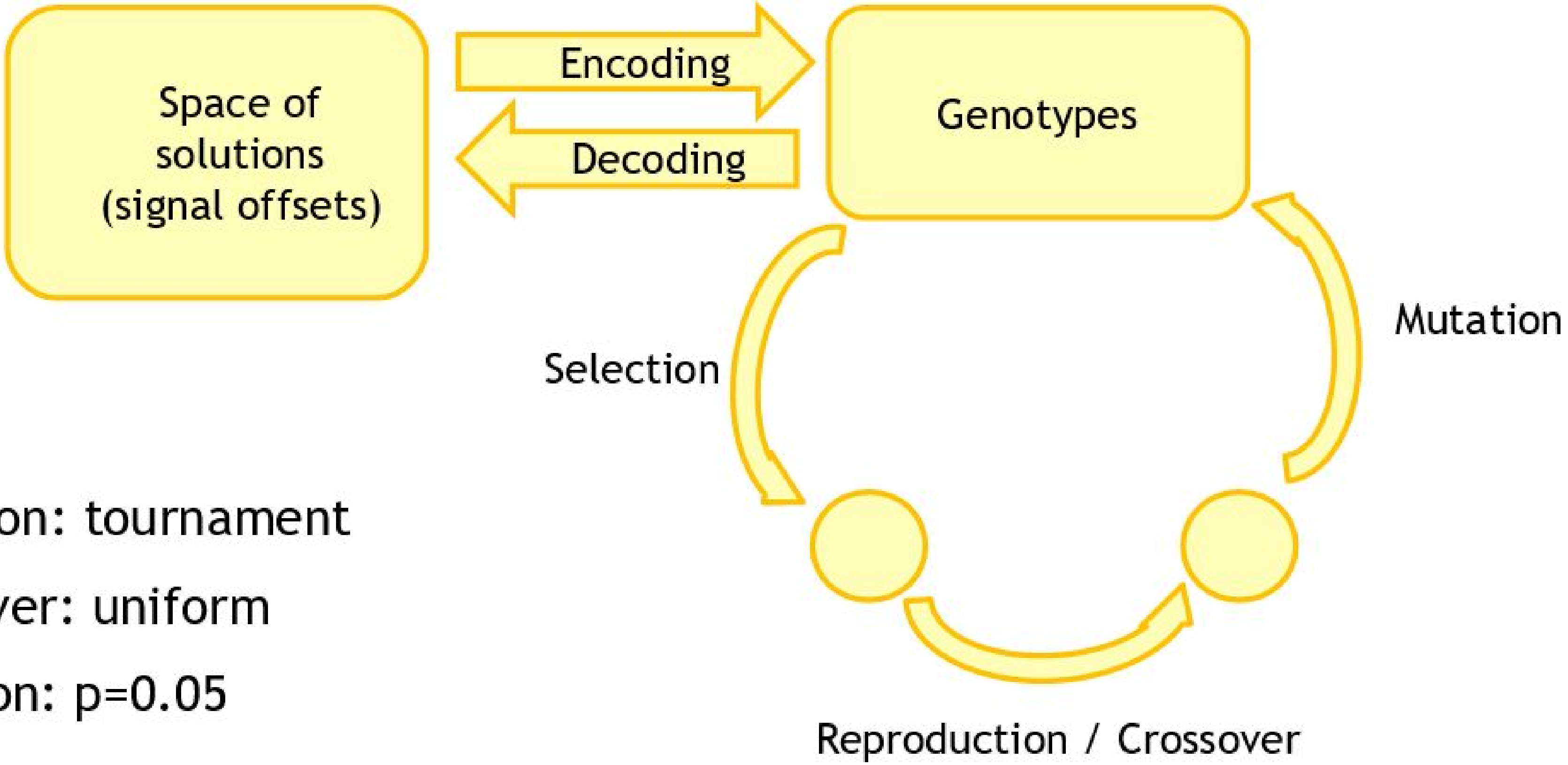
- We tested many NN architectures and training methods
- Training of a single model: ~15 minutes using GPU (Tesla P100)
- Best NN: [100, 100, 100], learning_rate = 0.01, training set: 10240 elements
- Avg error of approximations: 1.2%, max error: 6.8%
- Time of inference: ~0.2 sec (a few orders of magnitude faster than simulation)

TensorTraffic

Distribution of error (test set: 14634 cases):



Genetic algorithm for traffic optimization



Selection: tournament

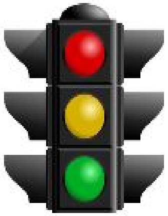
Crossover: uniform

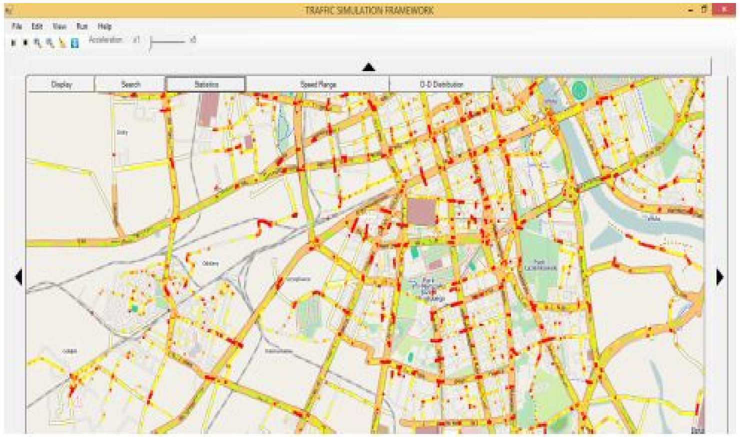
Mutation: p=0.05

(Simulation-based) Traffic optimization

Genetic algorithm which evaluates traffic signal settings using simulation gives good results. But time of computations was too long - it prevented real-time traffic optimization.

15 x



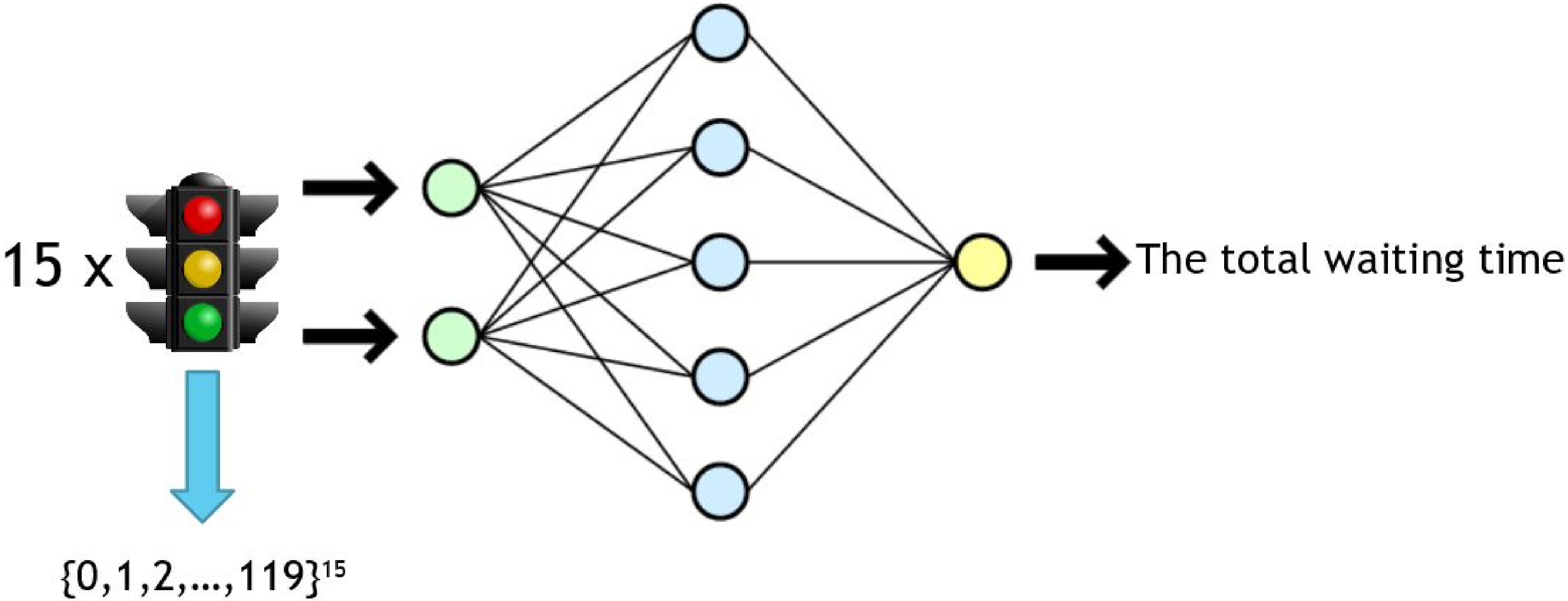


The total waiting time

$\{0,1,2,\ldots,119\}^{15}$

(AI-based) Traffic optimization

Genetic algorithm which evaluates traffic signal settings neural network gives good results and good time of computations.



(AI-based) Traffic optimization

Genetic algorithm:
For 100-element population after 30 iterations we can find genotypes 20-30% better than best genotype in the initial population (time: ~800 seconds, but < 30 seconds with GPU)

Comparison:
For 10 mln (randomly selected) settings, the best is worse by 2% (time of computations: ~600 seconds with GPU)

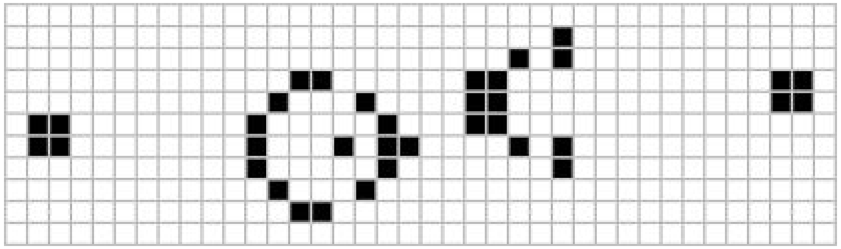
Options for the future research

- Although in the “rest area” topic we may not need to run as many simulations (however, we may consider road network of the whole Europe and not only Poland), it may be very interesting to check whether similar property of being accurately approximated holds for MATSim model

- The intuition is that the more complex the traffic model is, the more difficult it may be to approximate it

Options for the future research

- In case of TSF we don’t approximate real traffic, but only its simplified model (based on cellular automaton)
- Maybe this is a general property of cellular automata ? “Game of life”, “Rule 110” are Turing-complete, so ability to approximate them easily may be very useful



Options for the future research

How about approximating cancer growth?

Analogy:

vehicular traffic → cancer growth (modelling using cellular automata)

signal settings = offsets → cancer treatment settings = dates of radiotherapy

GA (or other metaheuristcs for finding optimal settings) give good results in both cases, time of computations is a common problem as well …

Conclusions

- We are instrumenting MATSim to estimate demand for parking places at rest areas in Poland (potentially - in the whole Europe)
- MOPSim may help in validating our methods and in finding optimal settings for parking places
- To speed up computations, we may use meta-models, e.g., neural networks which proved to give good results in case of the traffic signal setting problem

Conclusions

- Our approach of approximating outcomes of simulation models using machine learning is universal and may found applications beyond transportation area
- Support from Google, Nvidia, Microsoft (Azure Computational Grant)
- I am collaborating with many scientists and invite you to join

Thank you for your attention!

Paweł Gora
p.gora@mimuw.edu.pl
http://www.mimuw.edu.pl/~pawelg

„Logic can get you from A to B, imagination will take you everywhere”
A. Einstein

„Sky is not the limit”

Research in frame of the project "Parking space in rest and service areas (RSA)" financed by NCBiR/GDKKiA as a part of common undertaking "RID", under the contract DZP/RID-I-44/8/NCBR/2016.