VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

# MULTIKRITERIÁLNÍ GENETICKÉ ALGORITMY V PREDIKCI DOPRAVY

MULTI-OBJECTIVE GENETIC ALGORITHMS IN ROAD TRAFFIC PREDICTION

ROZŠÍŘENÝ ABSTRAKT DISERTAČNÍ PRÁCE
EXTENDED ABSTRACT OF A PHD THESIS


AUTOR PRÁCE Ing. JIŘÍ PETRLÍK
AUTHOR

VEDOUCÍ PRÁCE Prof. Ing. LUKÁŠ SEKANINA, Ph.D.
SUPERVISOR


BRNO 2016

# Contents

# Chapter 1

# Introduction

The understanding of road traffic behavior is a key to effective traffic control, management and organization. This task is becoming more and more important with increasing traffic demands and the number of registered vehicles. To the end of the year 2014, 6 775 877 vehicles were registered in the Czech Republic, which is 136 668 more registered vehicles than in 2013. The road traffic is essential for today economy and modern life. Unfortunately, it has also many negative effects such as traffic accidents or traffic pollution. According to Center of Traffic Research in Brno, the losses caused by traffic accidents were about 53 billions of Czech Crowns in the year 2013 [52].

These basic statistics show the importance of accurate information about the current traffic, which is available for drivers and for people responsible for road administration. To fulfill these demands, it is necessary to be able to measure the traffic flow variables. Traffic sensors like inductance loop detectors, radars, or traffic cameras are usually installed on one place are provide us with the information about the traffic on a given place. The other approach to measure the traffic flow is based on an equipment, which is installed into the chosen vehicles. This equipment is moving with a vehicle and provides the information about the vehicle position. We call the data obtained using this approach as a floating car data. However, with a development in communication and mobile devices, it is even not necessary to install a special device into the vehicle. Most of the drivers own a mobile phone and for mobile operators it is possible to track the position of these phones. Using the information about trajectories of phones, it is possible to get a very comprehensive information about the current state of the traffic.

Measured data can be distributed to the drivers, or can be further processed using traffic modelling and traffic prediction algorithms. The origins of traffic modeling and traffic flow theory dates backed to 1930s, when Bruce D. Greenshields performed observations of the traffic flow and postulated a simple traffic model [17]. Since then, various types of traffic models were proposed. These models typically differ in the aggregation level. The microscopic models work with each single vehicle and model its behavior. Contrary, the macroscopic models work with aggregated data and search for dependencies between the traffic variables.

This thesis is mainly focused on traffic prediction based on machine learning algorithms. The machine learning algorithms are part of computer science, evolved from pattern matching and artificial intelligence. These algorithms are able to adapt their behavior according to used training data. This thesis primarily deals with supervised learning principle in which we have a set of training samples with known values of results. These data are called training data set. Using the training data, the machine learning algorithm is capable of finding dependencies between input values and desired outputs. The trained models are then evaluated using another data and used for prediction. In many previous works, it was shown that the machine learning methods are very successful in short time traffic forecasting and travel times estimation. Soft-computing methods such as neural networks and support vector regression usually have various meta-parameters which should be properly set in order to achieve the best performance. The performance also strongly depends on the selected input variables. The reason is that many soft-computing methods can not work with missing inputs, or when some inputs can contain unimportant noise, which can deteriorate the quality of prediction. Hence automated multiobjective optimization methods are highly requested to simultaneously optimize relevant conflicting design objectives (such as parameters of soft computing models, selection of input data sensors etc.).

In the multi-objective optimization, it is possible to optimize two or more objective functions simultaneously. In the past, it was shown that the optimization method called genetic algorithms are very successful in dealing with multi-objective optimization problems. It is mainly because genetic algorithms internally work with a set of candidate solutions. The quality of these candidate solutions is evaluated in each iteration of the algorithm and the new set of candidate solutions is generated based on the most perspective solutions from the last iteration. Finally, because huge volumes of data are processed, the involved soft computing methods as well as the meta-optimization methods working over these methods have to be carefully implemented in order to minimize the execution time.

On the basis of the previous brief survey of problems relevant for soft computing methods used in road traffic prediction, we have identified an open problem which is almost untouched neither in the literature nor practice:

**An efficient approach is missing which will allow the soft computing methods to be automatically calibrated and utilized with the most suitable data samples in order to maximize the target quality measure.**

The approach proposed to solve this hard problem, which is developed in this thesis, is based on multiobjective evolutionary algorithms. In order to demonstrate the effectiveness of the proposed approach, it will be evaluated on several case studies and compared with relevant methods. In the thesis, we define two main goals:

**Goal 1: To propose a general framework for applying the multiobjective evolutionary optimization paradigm in the context of soft computing methods used in the area of traffic prediction and travel times estimation.**

**Goal 2: To evaluate the proposed framework using selected case studies. The subgoals are defined as follows:**

**Goal 2.1: To propose a new method for estimation of missing values in traffic density maps using a multi-objective genetic algorithms and compare this method with a conventional quadratic programming approach on real world data.**

Many intersections and roads in modern cities are equipped with some kind of measurement device (traffic sensor). Unfortunately, it is very expensive to cover each road and each intersection by these devices. In this thesis, we propose a new method to estimate the values of traffic intensity on the roads which are not covered by traffic sensors. This method is based on a multi-objective genetic algorithms and is capable to provide better estimation than traditional quadratic programming approach for some situations. The main advantage of the new method is that the knowledge of a traffic expert can be incorporated into the process of estimation. Further, we propose a method which combines the quadratic programming approach and multi-objective genetic algorithm.

**Goal 2.2: To propose a new multi-objective method for selection of input data (sensors) for support vector regression, in the task of short time traffic forecasting.**

The machine learning methods such as support vector regression are quite sensitive to the proper selection of input variables. This problem is known as the feature selection. The good feature selection is even more important for the traffic prediction, because a huge amount of data can be unavailable. This can be caused by sensor malfunction, or by many other reasons. And using a sensor, which is very often broken as the input of machine learning method can cause that the traffic prediction system is unavailable. On the other hand, the selection of a very small subset of input variables (sensors) can lead to a very inaccurate prediction. Proposed method is capable to find a proper subset of input variables for support vector regression (SVR). It is based on multi-objective genetic algorithm and especially useful in the scenarios with many missing data. The multi-objective genetic algorithm provides various trade-off solutions between the SVRs requiring many input variables (which provides a very precise prediction) and less precise SVRs utilizing only several input variables. We also propose a method enabling to dynamically switch among these SVRs during the prediction process. The most precise SVRs are then used in the situation, when the complete data are available and the less precise SVRs are employed in the situations when many of input data streams are missing.

**Goal 2.3: To evaluate the method for multi-objective selection of inputs for support vector regression in the task of travel times forecasting.**

The objective is to utilize the method introduced in the previous paragraph for the prediction of travel times. In this case, we will utilize the data from traffic sensors and the data from license plate reading systems as potential inputs. The principle of dynamic switching among the SVRs remains the same. The goal is to provide more accurate prediction, which is available for large portion of time.

**Goal 2.4: To further improve the traffic prediction by simultaneous multi-objective optimization of input variables and meta-parameters of support vector regression.**

The previous approaches have dealt with optimizing only the set of input variables, but the meta-parameters of SVR remained unchanged. However, this is far from the optimum, because for each SVR the optimal settings of the meta-parameters is different. In order to deal with this problem, we will simultaneously optimize the input variables and SVR meta-parameters.

**Goal 2.5: To provide an efficient parallel implementation of proposed methods.**

The optimization methods based on genetic algorithms are usually quite computationally expensive. Modern computers are equipped by hardware, which is capable to split computational effort among many processor cores. However, this requires a parallel algorithm, which is able to work on many processor cores. The objective is to design, implement and evaluate parallel implementations of proposed methods.

# Chapter 2

# Traffic flow modeling

## 2.1 Types of Traffic Data

The most comprehensive type of traffic data is the trajectory data. This data contains the information about trajectories of all vehicles in the area. They are often observed using cameras installed in a helicopter or mounted on the top of buildings. Pictures provided by the cameras are processed by recognition software. Each vehicle is recognized and the information about its trajectory $x_\alpha(t)$ is provided. One camera covers only a small area of a few hundreds square meters at most. As the process of data collecting is very complex many errors can appear. For example, a car can be hidden behind another bigger car. The image recognition software can also make some errors. Because of these problems, the process of collecting reliable trajectory data is very expensive.

The other type of traffic data is called the floating car data (FCD). In this case, selected vehicles are equipped with a system which is able to send information about the actual position of the vehicle. This is provided by GPS navigation, mobile phone or other kind of devices. Using obtained data, the trajectory of selected vehicles can be reconstructed. The selected vehicles can be additionally equipped with other sensors such as radars. These sensors provide the information about distance to the leading vehicle, or its speed. We call this kind of data as the extended floating car data (xFCD). The main difference between the FCD and trajectory data is that FCD data contains the information about only selected vehicles in the area. The information about the current line is not provided in FCD data. The reason is that GPS system is not precise enough to recognize the current line. Sometimes FCD are biased with respect to selected probe vehicles, for example, if the selected vehicles are public transport or taxies. On the other hand, FCD data can contain the information about the state of the vehicle and driver's behavior in the current situation [15, 31, 20].

The third type of data is called the cross-sectional data. This kind of data is provided by stationary sensors such as loop detectors, radars, cameras or infrared sensors placed on a certain location of the road. Provided information depends on the type of sensor. These sensors are transmitting either single vehicle data or data aggregated into time intervals [27].

## 2.2 Microscopic traffic variables

The microscopic traffic data provides the most detailed information about the state of the traffic because they contain the information about each single vehicle. These data are often connected with some place of measurement on which the traffic sensor is located, or the place is interesting from the traffic analysis point of view. The basic information, which practically each measurement technology is able to detect is time $t^0_\alpha$ in which the front of vehicle $\alpha$ passes the given place and time $t^1_\alpha$, in which the rear end passes the given place.

Using this basic microscopic data, it is possible to calculate other microscopic quantities, which are often called the secondary quantities. The length of vehicle $l_\alpha$ can be calculated as a time difference between the time when the front and tale of the vehicle passes the given place multiplied by the vehicle speed. Vehicle type is often detected by means of the length of vehicle. Time headway is a time distance between the front bumpers of successive vehicles. The distance headway is a distance between front bumpers of successive vehicles. Finally, the distance gap is the length between rear and front bumpers of neighboring vehicles. [27]

## 2.3 Macroscopic traffic variables

Microscopic data are often aggregated to time intervals between 20 seconds to 5 minutes, obtained data are the macroscopic data. The aggregation allows for the disc space and processing time reduction.

The traffic flow $Q(x,t)$ is the number of vehicles passing the current place $x$ within a time interval $\Delta t$. The occupancy $O(x,t)$ is a dimensionless variable representing the fraction of time interval during which the given place is occupied by vehicle. The arithmetic mean $V(x,t)$ speed is a mean speed of vehicles passing the given place. The values of traffic flow, occupancy and arithmetic mean speed can be obtained by aggregating variables directly measured by stationary traffic detectors. However, there also exist variables which are measured for the whole road segment. One of the most important ones is traffic density $\rho$. The traffic density is a spatial average of the number of vehicles on a given road segment, where speed V is a spatial average over the whole road segment. [27]

## 2.4 Travel times

Travel time $\tau_{12}$ is a time a vehicle needs to pass a road section $[x_1, x_2]$. The value of travel time depends on the start and end position $(x_1, x_2)$ and time $t$. Another useful variable for traffic-flow optimization and congestion analyses is called total travel time $\tau_{tot}$. It is a cumulative time spent by vehicles in the spatiotemporal region $[x_1, x_2] \times [t_1, t_2]$. [27]

## 2.5 Traffic sensors

Traffic detectors represent the traditional way of collecting traffic data [26, 29, 43]. There are two basic groups of traffic sensors: intrusive sensors and non-intrusive sensors. Intrusive sensors require the modification of the road surface and thus, in general, the installation of these sensors is not trivial. The non-intrusive sensors are installed somewhere near the road and don't require the modification of road surface. The sensors also differ in the utilized technology, quality of provided data and reliability. Depending on technology, these properties are affected by current weather or day time. The most utilized types of sensors are inductive loop detector, piezoelectric detector, magnetometer detector, radar and camera.

## 2.6 Soft-computing methods in traffic flow forecasting

In this chapter we will describe methods, which utilize neural networks or support vector regression to predict the future traffic state. The section is divided into two parts. In the first part, methods for forecasting of the basic traffic flow variables are described. The second part contains a description of methods for travel time estimation.

### 2.6.1 Traffic flow forecasting

Many studies about neural networks and traffic forecasting were proposed in recent two decades. We will deal with those that are most connected to the topic of this thesis. One of the first studise published about the traffic flow forecasting by neural networks was written by Brian Smith in 1994. In his study, he predicted the value of volume in next 15 minutes by backpropagation neural network. The inputs of neural network were the current volume, volume measured 15 minutes ago, historical volume and binary variable, which tells whether the pavement is wet. The results have shown, that the neural network outperforms the historical average and ARIMA model during the peak periods [44]. Another study written in 1997 compares time series methods, such as ARIMA and ATHENA with neural networks. The comparison was done on data from motorway near Beaune in France. However, the results of this study have shown, that traditional time series methods outperforms neural networks [28].

To successfully use neural networks, it is necessary to properly set the structure and various meta-parameters of the network. The current trend is to perform these tasks automatically. For example, genetic algorithms appear to perform well in this task. The evolutionary calibration of the neural network was tested by Vlahogianni et. al.. They proposed a method which can simultaneously optimize network dimension and learning parameters such as the learning rate or step size [53].

The first attempt to predict traffic flow using SVR appears in 2002. The motivation for SVR was its good generalization ability for a limited number of training samples, rapid convergence and capability to avoid local optima during the learning process. The authors focus on prediction of traffic volume. The next value is predicted from a few previously measured values. The method was tested using data from one intersection in Xian city, but the authors do not provide any comparison with other methods [12]. Another method for short term volume prediction is based on Online-SVR, which is a SVR modification capable of learning continuously during the preduction phase [33]. This Online-SVR was tested to predict the traffic volume in typical and atypical traffic conditions. Atypical traffic conditions are holiday traffic and the situation after the traffic collision. The authors compared the results of this method with other methods such as Gaussian maximum likelihood (GML), Holt exponential smoothing and neural networks. The results have shown that the OL-SVR has the second best prediction results after GML under typical traffic conditions and the best prediction results under the atypical traffic conditions [5].

Another approach is a combination of SVR with the chaotic simulated annealing optimization. In this approach, the SVR is utilized to perform the prediction, while simulated annealing optimizes the SVR meta-parameters. The prediction results have shown that this combined method outperforms seasonal autoregressive integrated moving average (SARIMA), Holt-Winters model and bacpropagation neural network [19].

SVR was also combined with a chaotic cloud particle swarm optimization. Similarly to the previous approach, the SVR is utilized to perform the prediction and the chaotic cloud particle swarm optimization optimizes the SVR meta-parameters. The method was evaluated using the traffic flow data from Dalian city [30].

### 2.6.2 Travel times forecasting

The neural networks appear to perform well in the task of travel time estimation. For example, Laurence Rilett proposed a method for forecasting freeway link travel time by multi-layer feed forward neural networks with the back-propagation learning algorithm. He considered four possible configurations of neural network inputs. In the first configuration, only the previously measured travel times on the road segment are used. In other three configurations, various subsets of neighbouring road segments were added among the inputs. The results for prediction 1 or 2 time steps ahead were best for the network, which used only the values measured on the given road segment. However, in longer prediction horizons, the neural networks with inputs from neighbouring sections performed better [37].

Another study utilized a different kind of neural networks called counter propagation network. The authors compared this neural network with traditional back-propagation and reported that the counter propagation neural network is one order of magnitude faster than learning of the backpropagation network and provides the results of the same quality [10].

SVR was used to predict travel times from the current and a few previously measured values. The method was evaluated by publicly available highway data from Intelligent Transportation Web Service Project [59, 60]. The authors compared SVR based method with other two methods. The first of them estimates the travel time from the speed at the enters of the road sections. The second method predicts the future travel time as the mean value of travel time at the same time of day and the same day in the week. The results showed, that SVR based method outperforms the above described methods [57].

## 2.7 Open problems

In this chapter we try to identify the open problems in the area of short term forecasting. A comprehensive summary of the open problems was provided in the article „Short-term traffic forecasting: Where we are and where we're going“ which was written by Eleneni et al [54]. We will focus on those problems that correspond to the topic of this thesis.

**Open problem 1: Arterial and network traffic predictions**

Most short-term traffic forecasting algorithms were built to predict the freeway traffic flow. It is because the traffic prediction for city arterials is a much more complex problem. It is necessary to deal with new problems, such as signalization and traffic lights. The complexity of the problem also arises with the number of intersections and complexity of road network in which many roads may not be covered by measurement devices.

The data driven approaches, such as machine learning algorithms can succeed in this complex environment, where other conventional methods usually fail [38, 46, 13].

**The methods proposed in this thesis are mainly designed to work in complex traffic networks. We used data from city arterials to evaluate the prediction quality of our methods.**

**Open problem 2: Short-term predictions: from volume to travel time**

Most studies dealing with short term traffic forecasting focus on prediction of traffic variables such as volume and occupancy. It is mainly due to the traditional measurement devices such as radars and loop detectors that are able to measure these variables. However, in recent years, many devices capable to measure the travel times were developed.

**As a part of this thesis, we proposed a new method to predict travel times. This method is able to combine the inputs from traffic sensors and modern license plate reading systems.**

**Open problem 3: Combining of the models**

The quality of prediction is usually determined using an error metric such as the root mean squared error (RMSE) or mean absolute error. However, such a comparison is not always fair [25]. It is also necessary to consider other aspects, such as time complexity, adaptability, robustness and requested expertise and skills.

Because it is often hard to decide which model is the best one, it can be very useful to develop methods and heuristics which are capable of combining the results. For example, the approach which combines the backpropagation and RBF neural networks appears to outperform singular predictors on the task of freeway volume production. The Bayes rule was utilized to combine the results of two different neural networks in this approach [62]. Another method uses the neural network to combine the prediction results provided by three different models. These models are moving average, exponential smoothing and ARIMA [48]. The fuzzy logic also appears to work well in combining the model results [45].

**In our approach, we combine multi-objective optimization methods with machine learning algorithms. These methods internally create different models, which are dynamically switched according to the currently available data.**

**Open problem 4: Employng the full potential of artificial intelligence**

In the recent years, algorithms of modern artificial intelligence (AI) are becoming more widely used in the area of traffic systems. At the beginning the AI models were mainly used in the area of data analysis and traffic forecasting. However, it is possible to use them in many other areas such as modeling of driver behavior and employ them in decision making in modern ATIS and ATMS systems [35].

There is also a sceptical view, caused mainly by three reasons. Many results produced by artificial intelligence such predictors based on neural networks can not be interpreted by human and can be considered as a kind of black box. The second reason is that these methods do not guarantee finding an optimal solution. Moreover, many of them do not guarantee finding even a feasible solution. This problem is more visible for evolutionary optimization techniques. The third reason is that the AI methods often have various meta-parameters which must be set properly in order to obtain sufficient results. However, the setting of these parameters is often not trivial and requires a lot of expert knowledge. [6].

**Methods designed in this thesis can calibrate their meta-parameters using multi-objective genetic algorithms or self-adaptation.**

# Chapter 3

# Soft-comupting methods

Soft-computing methods are capable of dealing with uncertainty and problems showing high complexity. These kinds of problems often appear in the area of image processing, signal processing, multi-objective optimization and many others. In this thesis, we mainly utilize two types of soft-computing methods. The first type is machine learning based on neural networks and support vector regression and the second type is a genetic algorithm.

## 3.1 Machine learning algorithms

Machine learning algorithms are capable of adapting their behavior according to the input data. There exist three basic types of machine learning algorithms. In the case of supervised learning, the machine learning model is trained using these samples. Then the model can be used to provide answers for new samples. The second type of machine learning algorithms uses unsupervised learning technique. The input data doesn't contain the correct answers and the goal of the learning process is to find similarities in data and select the correct categories for samples. Finally, in reinforcement learning, the algorithm gets to know when the answer is wrong, but does not get the correct answer. In this thesis, we will focus mainly on the supervised learning techniques.

In the supervised learning scenario, we have given a set of pairs $D = \{(\vec{x}^{(n)}, \vec{y}^{(n)}), n = 1 \dots N\}$. The goal of the learning is to find mathematical dependency between the input $\vec{x}$ and the output $\vec{y}$ in such manner that if a new value $\vec{x^*}$ is given, the calculated value $\vec{y^*}$ is shows as small error as possible. The new tuple $(\vec{x^*}, \vec{y^*})$ may not be in the set $D$, but has to be generated by the same process as the members of $D$. In the case the value of $\vec{y}$ belongs to one of a few discrete values, the process is called classification. In the case the value of $\vec{y}$ is continuous, the process is called regression.

The supervised learning process has three phases. In the phase one, the first part $D_{train}$ is used to train the machine learning model. Then it is necessary to validate the quality of the trained model using set $D_{test}$ (phase 2). If the model provides results of sufficient quality then it is ready for the use (phase 3) [34, 50].

## 3.2 Support vector machine and support vector regression

The support vector machine (SVM) is a very popular soft computing method for solving classification tasks [3]. It was successfully used in the area of computer vision [22, 41], handwriting recognition [61], bioinformatics [40], economy [23] and many others. The basic variant of SVM is capable of solving only linearly separable problems. The algorithm becomes more powerful when the so-called kernels are introduced. Kernels are special functions capable of transforming input data into a more dimensional space. This transformation allows SVM to successfully solve non-linear problems, because in these more dimensional spaces data can often be separated.

The tasks discussed in this thesis are mostly regression tasks and SVM algorithm is not able to solve them. Fortunately there exists a modification of SVM: support vector regression (SVR) which is able to solve regression tasks. SVR appears to be very useful in forecasting of time-series such as [49, 4, 56].

However, SVM and SVR have various meta-parameters, such as the kernel function, regularisation parameter and other parameters related to currently used kernel. It is necessary to set up these meta-parameters properly in order to obtain the best classification or regression quality [14]. In the past, several conventional optimization techniques and guidelines were proposed to solve this problem [32]. Some of these approaches are based on the principles of evolutionary algorithms [21, 55, 56].

For example, the HGA-SVR is a method for kernel function selection and parameter optimization in SVR. This method is based on genetic algorithm which simultaneously optimizes the type of kernel and meta-parameters of SVR. In case of HGA-SVR, each chromosome consists of the integer part and real valued part. The integer part has one value, which defines the kernel type: linear kernel (0), polynomial kernel (1) or RBF kernel (2). The real valued part of chromosome specifies the values of SVR meta-parameters.

This method was originally designed to predict maximal daily electric load. It was tested on the daily electricity loading problem announced at 'World Competition within the EUNITE network'. The EUNITE is abbreviation for 'European Network on Intelligent Technologies for Smart Adaptive Systems'. HGA-SVR method provided better generalization capability and a lower prediction error than other approaches based on neural networks or traditional SVR [56].

## 3.3 Multi-objective optimization

In the area of single objective optimization, the quality of a candidate solution is defined using one objective function $f$. The goal of the single objective optimization is to find a solution with the minimal or maximal value of a given function $f$. However, many real world optimization problems can't be described using only one objective function and it is necessary to use more objective functions $f_1, \ldots, f_m$, each of which has to be minimized or maximized. In this case, we speak about a multi-objective optimization. In many cases, objective functions are conflicting, i.e. improving one objective means worsening the other one. More formally, the multiobjective optimization problem (MOOP) is defined as follows [7]:

$$
\begin{aligned}
\text{minimize: } & f_m(\vec{x}), && m = 1, 2, ..., M \\
\text{subject to: } & x_i^{(L)} \leq x_i \leq x_i^{(U)}, && m = 1, 2, ..., M \\
& g_j(\vec{x}) \geq 0, && j = 1, 2, ..., J \\
& h_k(\vec{x}) = 0, && k = 1, 2, ..., K.
\end{aligned}
\tag{3.1}
$$

A candidate solution $\vec{x}$ is a vector of $n$ decision variables $\vec{x} = (x_1, \ldots, x_n)$. Functions $f_1, \ldots, f_m$ represent objective functions. Each component $x_i$ of vector $\vec{x}$ must be within the range $x_i^{(L)}$ and $x_i^{(U)}$. We call values $x_i^{(L)}$ and $x_i^{(U)}$ as variable bounds. The functions $g_i$ and $h_i$ are called constraint functions, where $g_j$ define inequality constraints and $h_k$ define equality constraints. Solution $\vec{x}$ is feasible if it satisfies all constraints and is within defined variable bounds. Otherwise, $\vec{x}$ is an infeasible solution [7].

One of the most important difficulties in the multiobjective optimization is how to compare the quality of candidate solutions. In the single objective optimization domain, the situation is quite straightforward. The solution $\vec{a}$ is better than solution $\vec{b}$ if the value of $f(\vec{a})$ is better than $f(\vec{b})$. We will denote the situation in which the value of function $f$ for solution $\vec{a}$ is better than for $\vec{b}$ as $f(\vec{a}) \lhd f(\vec{b})$. In the case of minimization of function $f$, the statement $f(\vec{a}) \lhd f(\vec{b})$ would mean that $f(\vec{a}) < f(\vec{b})$. In the case of maximization, the same statement would mean that $f(\vec{a}) > f(\vec{b})$. Moreover, we will denote the situation in which the value of function $f$ for solution $\vec{a}$ is not worse than for $\vec{b}$ as $f(\vec{a}) \not\rhd f(\vec{b})$. This means that $f(\vec{a}) \leq f(\vec{b})$ in case of minimization and $f(\vec{a}) \geq f(\vec{b})$ in case of maximization.

However, in the multiobjective optimization the situation is much more complicated. The main problem is that we have a vector with two or more values of objective functions for each candidate solution. Because of it, the Pareto dominance relation was established [7].

**Definition 1** *A solution $\vec{a}$ is said to dominate the other solution $\vec{b}$ ($\vec{a} \preceq \vec{b}$), if both following conditions are met:*

1. *The solution $\vec{a}$ is not worse than $\vec{b}$ in all objectives ($f_i(\vec{a}) \not\rhd f_i(\vec{b}), \forall i = 1, \ldots, M$).*
2. *The solution $\vec{a}$ is better than $\vec{b}$ in at least one objective* ($\exists i \in \{1, \ldots, M\} : f_i(\vec{a}) \lhd f_i(\vec{b})$).

Figure 3.1 shows an example of Pareto dominated and non-dominated solutions. The depicted scenario expects minimization of two objective functions $f_1$ and $f_2$. We can say that solution $A$ dominates $C$. The first condition is satisfied because $f_1(A) < f_1(C)$ and $f_1(A) = f_2(C)$. The second condition is satisfied, because $f_1(A) < f_1(C)$. The most interesting subset of solutions consists of solutions which are not dominated by any other solution in the set (filled circles). We call these solutions non-dominated solutions.

## 3.4 Multi-objective genetic algorithms

Genetic algorithm (GA) is a very popular method for solving optimization problems. The first version of GA was proposed by John Holland in seventies. GA is inspired by evolutionary and genetic processes of nature. GA is an iterative algorithm, which internally works with a set of candidate solutions. A new set of new candidate solutions replaces the previous set in each iteration. The new set of candidate solutions is produced using selection, crossover and mutation operators. In the area of GA, we call the iteration as generation and the set of candidate solutions as population. The items in the population are called individuals or chromosomes.

GA works as follows.At the beginning the initial population is created. Fitness values are calculated for each individual in the population. Then the termination criterion is evaluated. If the termination condition is satisfied, the algorithm ends. Otherwise, the selection operator is performed. This operator selects perspective candidate solutions for the crossover and mutations. The crossover operator creates a new solution by using parts

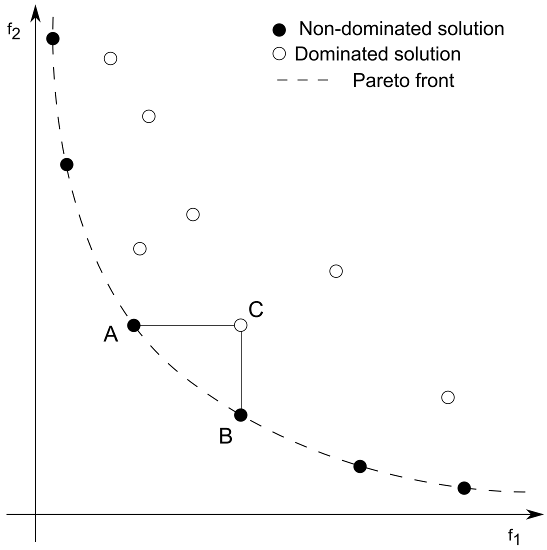


Figure 3.1: Examples of dominated and non-dominated solutions.

of two or more previous solutions. The mutation operator creates a new chromosome by a small modification of the parent chromosome [42].

Unfortunately, the original genetic algorithm can solve only single objective problems. To overcome this disadvantage, many modifications of GA which are capable to solve the multi-objective problems were proposed. For example, the fast and elitist multi-objective genetic algorithm NSGAII splits the entire population into many non-dominated layers in each generation [8]. The solutions into the new generation are selected according to which non-dominated layer they belong to. A new approach for non-dominated sorting is utilized in the algorithm. This approach has the complexity $O(MN^2)$ instead of $O(MN^3)$, where $M$ represents the number of objectives and $N$ represents the population size. The algorithm also utilizes a mechanism for preserving of the diverse set of candidate solutions. This mechanism doesn't need any sharing parameter like previous version of NSGA. One of the important advantages of the NSGAII is that it can easily deal with constraint optimization problems.

In multi-objective optimization different solutions often appear, which have the same values of objective functions. In prediction tasks these solutions can, for example, represent models with the same quality, but with different input sensors. Obtaining many solutions of this type can be very useful, because we can use the proper solution according to the situation for a particular set of sensors and without any loss of quality.

To find multi-modal solutions, it is necessary to use modified versions of multi-objective genetic algorithms. In the past, a modified version of NSGAII for solving of multi-modal problems was proposed [9]. The difference between the standard NSGAII and modified version is in the accommodation of solutions into the new population.

# Chapter 4

# Traffic data

The traffic data are essential for understanding of the current traffic situation and traffic behavior. These data are obtained by methods described in Chapter 2 and sent by protocols such as Datex II or Alert-C into the traffic data centers, where they are stored, processed and further analyzed. In this chapter, we will discuss the data, which will be used for verification of our new methods for traffic prediction. Two groups of data are considered. The first group contains the data from the city of Seattle and we will use them to verify our methods for data imputation, traffic forecasting and travel time prediction. The second group contains the data from Prague and we will use them to verify our method for estimation of missing values in traffic density maps.

## 4.1 Research Data Exchange Project

The Research Data Exchange [1] is a project developed to share traffic data to researchers, application developers, and others. Provided data are well-documented and freely available to the public. The data are divided into many datasets according to the place of measurements or data source. Many different sources of measurements are available, for example traffic detectors or probe vehicles.

In particular, Seattle Sensys Dataset containing the data from sensors in the downtown of Seattle and Arterial Travel Time dataset containing travel time data will be discussed in this thesis.

The Seattle Sensys Data was collected from traffic sensors. This data contains information about the traffic flow, occupancy and mean speed on selected intersections (see the map in Figure 4.1). In this case, sensors are placed at 23 intersections in the city. The data was measured from May 1 to October 31 and aggregated to 1 minute intervals.

**Observation 4.1: The data obtained by traffic sensors contains many missing values. The analysis has also shown that there are many correlations between sensors. It is very desirable to create a method capable to approximate the missing values using the data obtained by other sensors in the area.**

The second type of data are the travel times of vehicles measured by a license plate recognition system. The data are distributed in two tables in *Arterial Travel Times* dataset. The first table provides locations of cameras, camera ID, GPS coordinates and primary and secondary streets. The second table contains the travel times of individual vehicles matched by the camera system (IDs of camera A and B, time stamps and travel time).

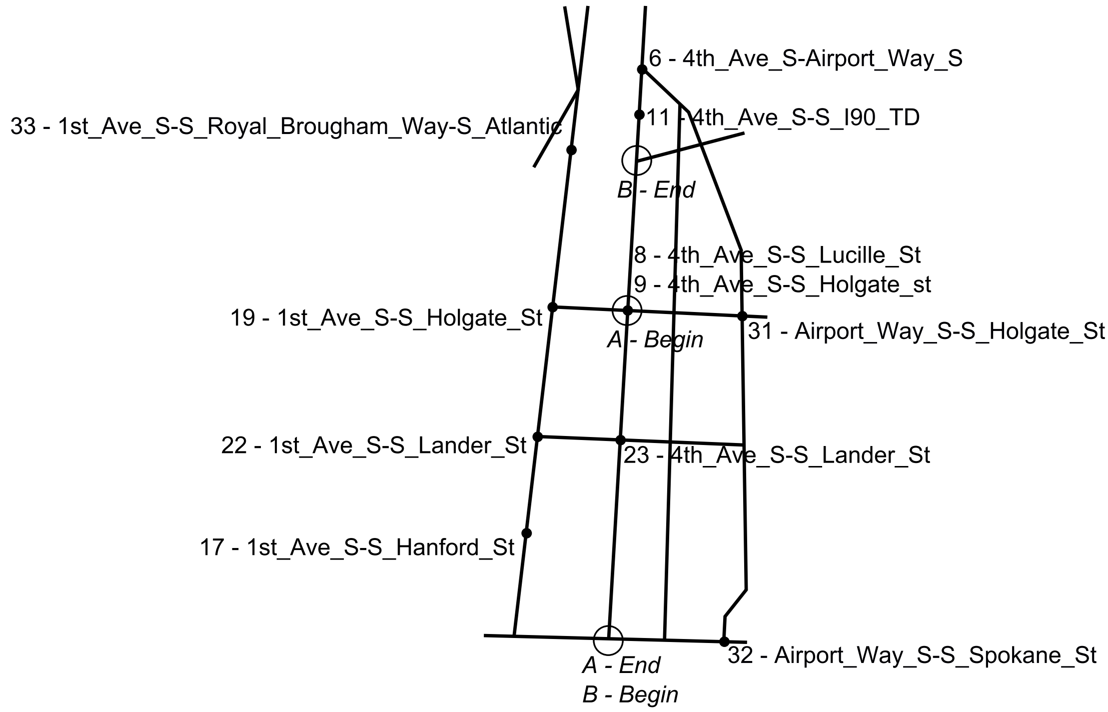


Figure 4.1: Map of the centre of the area. Sensors are marked by filled circles. Measured travel time road segments are marked by empty circles.

In the data, 97 different combination of begin and end places are recorded. This means that there exists at least 97 possible trajectories of vehicles for which it is reasonable to measure travel times. However, most of these trajectories are not interesting for data analyses, because of the low number of vehicles recorder for them. We tried to analyse trajectories taken by more than 15000 vehicles per month.

**Observation 4.2: The travel times data provided by the license plate reading system are strongly dependent on the daytime. The number of matched vehicles decreases at night and early in the morning. The travel times are also deteriorated by many outlier values. On the other hand, missing values in the data comming from traffic sensors do not depend on daytime and do not contain too many outliers, but it is harder to use them for travel time prediction. A method is needed capable of utilizing both data types (from license plate reading system and sensors) to provide more robust and precise travel times prediction.**

## 4.2 Prague data

The second source of data for our work was the dataset for Prague. This dataset is not provided by ITS-RDE and we used it mainly to verify our method for estimation of missing values in traffic density maps.

The dataset contains the information about traffic volume in Prague for years 2008 and 2009 (Table 4.1). It is split into two tables. The first table provides the information about intersections (intersection id and GPS coordinates). The second table contains the data for road segments connecting the intersections. The id, beginning and end intersections define each segment. The table provides the value of intensity for segments measured in the year 2008 and 2009.

| | |
|---|---|
| Number of intersections | 126 |
| Border intersections | 28 |
| Inner intersections | 98 |
| Number of road segments | 277 |
| Measured road segments in 2009 | 117 |
| Not measured road segments in 2009 | 160 |

Table 4.1: Prague data - basic description.

We analyzed the values of traffic volume in the year 2008 and 2009. It appears there are a few segments with very high volume (about 50000 vehicles) and many others with values about 10000 vehicles.
**Observation 4.3: An algorithm with specific properties is needed to estimate missing values in traffic networks. In particular, the algorithm should be able to tolerate inaccurate measurements and it should be able to exploit data from previous measurement to improve the quality of the result.**

# Chapter 5

# Estimation of missing values in traffic density maps

The traffic density map (TDM) represents the density of road network traffic as the number of vehicles per a specific time interval. This interval can be given in minutes or hours. Usually, TDMs are used by traffic experts as a base documentation for planning a new infrastructure (long-term) or by drivers for showing a current traffic status (short-term). Such TDMs can be composed automatically – with the aid from standard surveillance technologies (e.g. various data sensors such as loop detectors or traffic cameras). Another approach, which can be used for TDM calculation, is the manual counting on selected road segments. However, counting where people are involved in the process is usually quite inaccurate and also inefficient [36].

## 5.1 Method

In this thesis we propose a new approach for estimation of missing values in traffic density maps, which is based on NSGAII. This enables us to obtain more realistic solutions, because we can consider more aspects in optimization process. In our approach, each candidate solution is defined by a vector of real numbers. Every component of the vector represents a traffic density on one road segment, for which the density is not avalaible. The parameter value should be rounded to have the integer value. In the first generation of GA, components of vectors are initialized to positive randomly generated values. Then a single point crossover and a normally distributed mutation are utilized during the optimization process.

The main reason to utilize NSGAII is that it allows us to use more fitness functions directly. In our case there will be two objective functions. The first is the sum of errors on nodes and the second is the sum of differences to historic values.

In order to maximize performance of the genetic algorithm it is necessary to correctly set various control parameters such as the population size, the probability of crossoever, the probability of mutation etc. In our evolutionary approach we use a self-adaptive method, which enables to encode some control parameters of genetic algorithm in to the chromosome [18], [2] and [16].

In order to obtain the best performance, we propose three variants of evolutionary estimation of missing values in TDM. The first variant is based on the multi-objective genetic algorithm and doesn't use the self-adaptation. The second variant utilizes the self-adaptation. The first two variants start with a randomly generated initial population. The third variant uses quadratic programming approach to generate the initial population. The result of QP approach is transformed into chromosome and this chromosome is copied into the initial population. The third variant also uses self-adaptation.

## 5.2 Experimental results

In order to evaluate the proposed methods we utilized field data from annual manual counting from the city of Prague (counting in year 2008 and 2009). The data cover the central part of the city which is modelled using 126 nodes (28 of them are border nodes) and 277 edges (117 of them without the traffic intensity).

It was shown that two-objective optimization process gives many tradeoff solutions situated on the Parreto front. This is usefull for iterative estimation, because one can choose the best trade-off according to his/her knowledge. Also, it is possible to use the constraints in the same way discussed in the QP approach. The best results results were obtained by combining both methods, when the initial solution is generated by the quadratic programming and then further optimized by GA.

We developed a computer program in Java, which uses an incremental process of traffic density estimation. This process is supposed to be driven by a user – traffic expert. At the begining the user sets the values for measured edges and runs the multiobjective genetic optimization process. There are several optimized solutions at the end of this process. One of them can be chosen and eventually edited. The user can also change importance of the errors on nodes and constraints as mentioned previously. After this editing, the optimization process can be performed again and again. This iterative process continues until a sufficient estimation is reached 5.1.

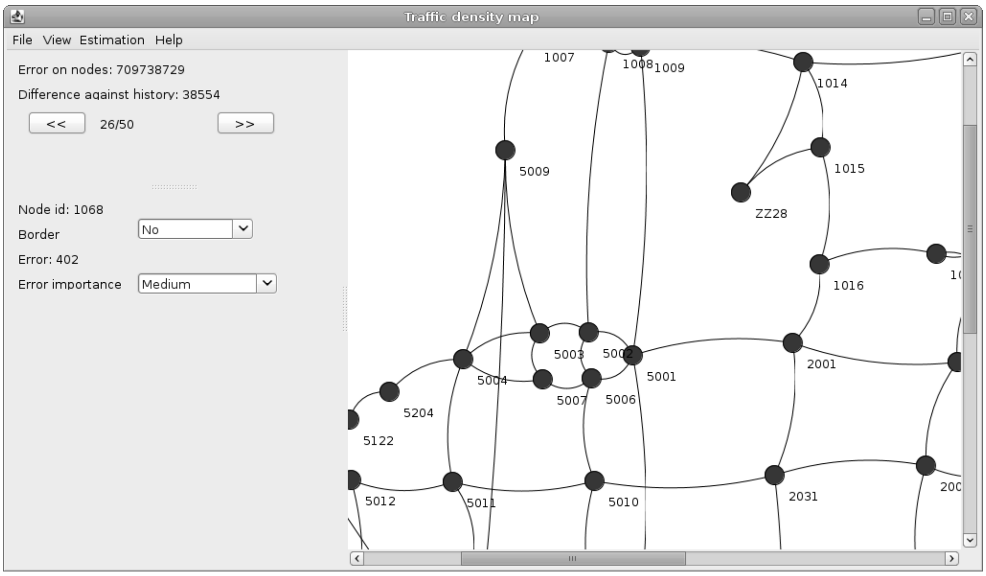


Figure 5.1: A screenshot of application for estimation of missing values in TDM.

# Chapter 6

# Multiobjective Selection of Input Sensors for SVR Applied to Road Traffic Prediction

Modern traffic sensors can measure various road traffic variables such as the traffic flow and average speed. However, some measurements can lead to incorrect data which cannot further be used in subsequent processing tasks such as traffic prediction or intelligent control. In this thesis, we propose a method selecting a subset of input sensors for a support vector regression (SVR) model which is used for traffic prediction. The method is based on a multimodal and multiobjective NSGA-II algorithm. The multiobjective approach allowed us to find a good trade-off between the prediction error and the number of sensors in real-world situations when many traffic data measurements are unavailable.

## 6.1 Method

The proposed method can be used to either predict the traffic flow or estimate missing values for a broken sensor. In the first phase, the SVR model is trained using historical data (the train set) in the supervised learning scenario. Trained SVR model then describes mathematical dependencies among the values of the sensor for which predictions are desired and other sensors in the area. Other historical data, unseen during the learning phase (the test set), are used to validate the resulting model. The multiobjective multimodal NSGA-II algorithm is employed to find the proper subset of input sensors for the SVR model.

Traffic data are usually available as a set of time series $s_1, \ldots, s_n$; one time series for each variable measured by a traffic sensor. In order to train the SVR model, it is necessary to convert these data into training samples (Fig. 6.1). By means of a sliding window, the current value ($s_i^{(0)}$) and a few ($h$) previous values ($s_i^{(-1)}, \ldots, s_i^{(-h)}$) from each series are taken into a training sample. In the case of estimating the current value of a broken sensor (Fig. 6.1, left), the current value $f^{(0)}$ is included into the training sample as a dependent variable. In the case of traffic forecasting in the place of sensor, the future value $f^{(+l)}$ is included into the training sample (Fig. 6.1, right), where $l$ represents the prediction horizon.

We employed the multiobjective multimodal NSGA-II [9] operating over binary strings to select proper input sensors for SVR. Each gene represents one input sensor, where 1 denotes including and 0 excluding of a particular sensor from the input vector fed to SVR (Fig. 6.2).

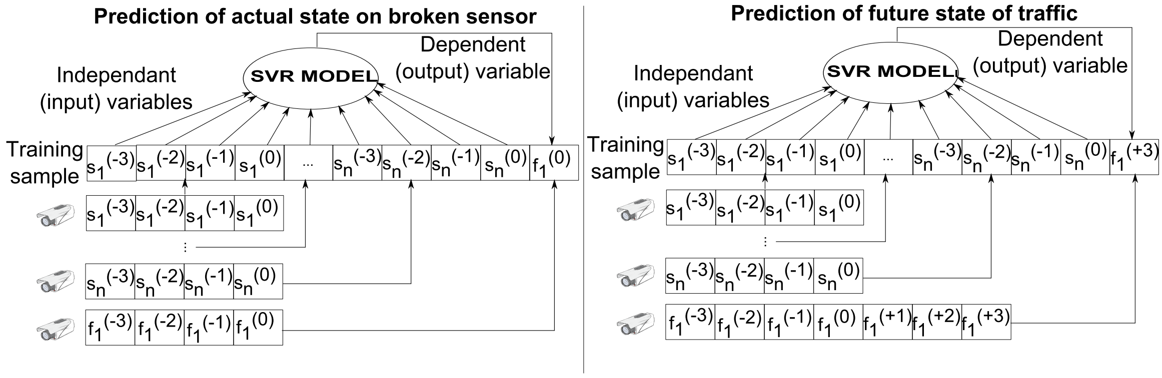


Figure 6.1: Composition of training samples for SVR: prediction of a current value (left) and prediction of a future value (right) of a sensor producing $f$.

Three objectives are considered (all to be minimized) – the number of sensors used as inputs for SVR, the rate of missing samples for prediction and the prediction error. The rate of missing samples is portion of time for which the concrete model can't be used because of missing data. Two well-known error metrics can be used as error objective function: root mean squared error (RMSE) or relative squared error (RSE). All objectives are evaluated using the test set.

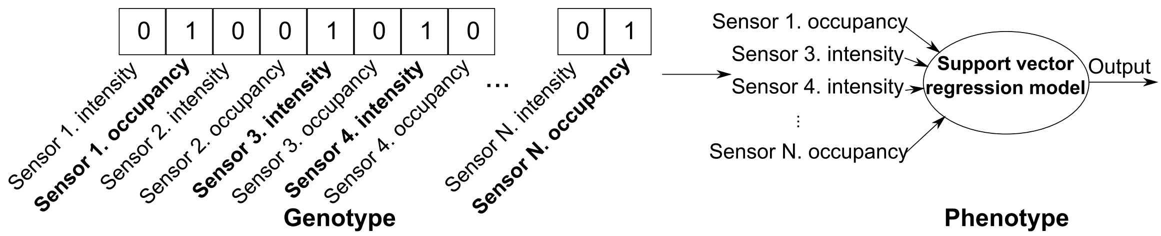


Figure 6.2: Chromosome encoding and a corresponding phenotype (SVR model)

## 6.2 Method Evaluation

The proposed method was evaluated on places 6, 11, 19, 22, and 23 of the Seattle area. For each sensor located on these places, four SVR models were created. The first two SVR models are trained to perform a short-term prediction in the horizon of 15 minutes. One of them uses only the actual values measured on the neighbor detectors in the area and the second one uses the actual values and the values measured on these sensors in previous 15 minutes. The other two SVR models are trained to estimate the actual value on the sensor in the case of a sensor error. And again, one of them uses only the actual values measured on the neighbor detectors in the area and the second one uses the actual values and the values measured on these sensors in previous 15 minutes.

In order to justify the multiobjective approach, we consider a single criterion optimization scenario, in which RMSE is used as the only fitness function. The single-objective GA works with 40 individuals in the population, the probability of crossover is 70%, the probability of mutation is 5%, and 2-individual tournament selection (with elitism) is chosen. Table 6.1 compares NSGA-II with the single objective GA for several places and sensors

| Location | | | Multiobjective approach RMSE for Unavailable ratio: | | | | Best single objective GA result | |
|---|---|---|---|---|---|---|---|---|
| Place | Sensor | Variable | < 10% | < 30% | < 50% | < 70% | RMSE | Unavailable ratio |
| Current values on sensor. | | | | | | | | |
| 11 | 3 | traffic flow | 5.27 | 4.63 | 4.16 | 4.01 | 2.66 | 96.9 |
| 11 | 3 | occupancy | 3.81 | 3.50 | 3.31 | 3.31 | 0.31 | 99.4 |
| 22 | 4 | traffic flow | 5.33 | 4.86 | 4.31 | 4.20 | 1.48 | 99.4 |
| Prediction horizon 15 min. | | | | | | | | |
| 11 | 3 | traffic flow | 5.50 | 4.9 | 4.37 | 4.23 | 2.96 | 97.2 |
| 11 | 3 | occupancy | 4.02 | 3.57 | 3.41 | 3.35 | 0.33 | 99.4 |
| 22 | 4 | traffic flow | 5.51 | 4.89 | 4.56 | 4.35 | 1.84 | 99.0 |
| Current values on sensor, 15 min. history. | | | | | | | | |
| 11 | 3 | traffic flow | 5.20 | 4.58 | 3.91 | 3.34 | 1.15 | 99.4 |
| 11 | 3 | occupancy | 4.04 | 2.72 | 2.18 | 1.50 | 0.19 | 99.4 |
| 22 | 4 | traffic flow | 5.62 | 4.71 | 4.09 | 3.37 | 1.04 | 99.4 |
| Prediction horizon 15 min., 15 min. history | | | | | | | | |
| 11 | 3 | traffic flow | 5.62 | 5.05 | 4.48 | 3.82 | 1.17 | 99.4 |
| 11 | 3 | occupancy | 4.03 | 2.68 | 2.27 | 1.59 | 0.24 | 99.4 |
| 22 | 4 | traffic flow | 5.64 | 4.98 | 4.17 | 3.66 | 1.15 | 99.4 |

Table 6.1: The best RMSE on selected sensors and places for NSGA-II (less than 10%, 30%, 50% and 70% samples unavailable) and a single objective GA

(the best values from 20 independent runs are reported). It can be seen that the single objective GA tends to provide solutions with very small RMSE values; however, it opportunistically exploits the test data containing over 85% missing values (in many cases, over 99%, see the Unavailable ratio column). Such a SVR model will thus be useless in practice, because it will not provide any prediction most of the time. Therefore, the single optimization scenario fails in this task.

Experiments show that the proposed method, in contrast with a common approach reported in the literature, can provide reasonable results even if many samples are unavailable.

# Chapter 7

# Multiobjective Selection of Input Sensors for Travel Times Forecasting Using Support Vector Regression

Travel time information can be used for various purposes, for example, as a source for different transportation analyses or for drivers who are planning their itinerary. Especialy in the situation when there are two or more possible ways to travel and the driver wants to choose one which will take shorter time.

In this thesis, we propose a new method for *travel time prediction* which uses a *support vector regression*. The inputs of our method are data from license plate detection systems and traffic sensors such as induction loops or radars placed in the area. These traffic sensors enable us to measure traffic variables such as traffic volume, road occupancy, average speed and many others [27]. This method is mainly designed to be capable of dealing with missing values in the traffic data. The method is able to create many different SVR models with different input variables. These models are then dynamically switched according to which traffic variables are currently available. Hence the proposed method provides much reliable predictions than traditional license plate detection methods or regression methods, which use the data obtained by a static set of traffic sensors.

In the past, many methods for travel time estimation have been proposed. In this thesis, we will utilize a license plate based approach and a regression based approach.

## 7.1 Known methods for travel times prediction

One of the approaches to measurement and prediction of travel times is based on license plate recognition. The basic principle of this approach is depicted in Figure 7.1. It is necessary to have at least two cameras. The first camera (A) is placed at the begining of the road segment and the second one (B) is placed at the end of the road segment. Each camera is connected to a pattern recognition software, which is able to read numbers on license plates. The license plate number for each vehicle is stored in a database. When the same license plate number is detected by both cameras for a given vehicle then the travel time can be calculated [47, 24].

Based on this data, the information of the expected travel time can be displayed to drivers at the begining of the road segment. The main advantage of this method is that it is a quite straightforward principle. The biggest drawback is that the information displayed to drivers is delayed because it is based on conditions which existed in the current place when the last vehicle leaving the road segment was registered by camera A. Another drawback is that the license plate reading systems are not highly reliable and many vehicles pass the segment undetected.

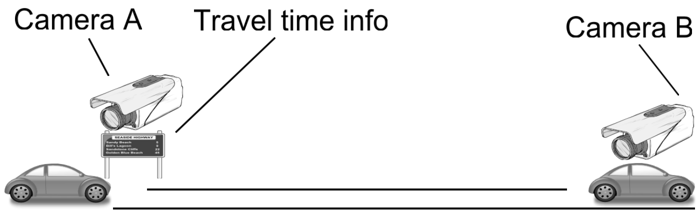


Figure 7.1: The principle of travel times estimation using license plate recognition.

Another approach to predict travel times is based on the data from traffic sensors such as radars, loop detectors, etc. Based on historic experience, the regression approaches predict the expected value of travel times using traditional supervised learning techniques known from the machine learning area. For example, the regression model can be constituted using neural network or support vector regression [58, 11].

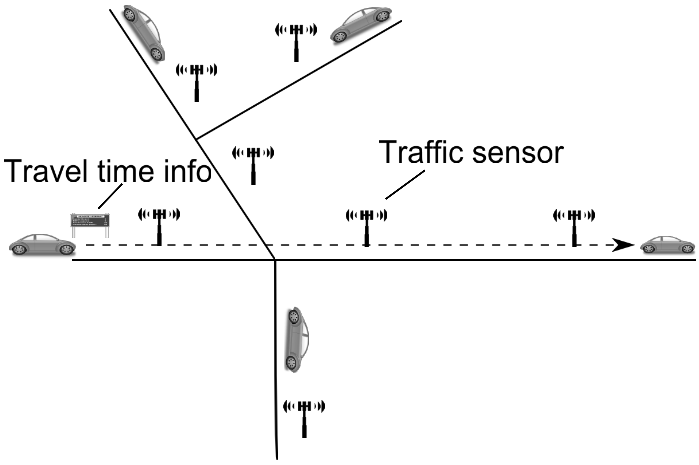


Figure 7.2: The principle of travel times estimation using traffic sensors in the area.

## 7.2 Method

The quality and availability of both described methods for travel times estimation is largely dependent on the quality of the input data. In the license plate based approach, the quality of prediction is largely dependent on the quality of the pattern recognition software for license plate reading. The data produced by sensors such as loop detectors, radars and camera detectors is not absolutely reliable and there are many missing values. Unfortunately, support vector regression and many other regression methods can not deal with missing values. In our approach, we will focus on selection of proper input sensors for travel times prediction.

In order to predict travel times, SVR with a radial basis kernel is exploited. Potential input variables for SVR are the data measured by traffic detectors in the area and travel times estimated by a license plate recognition system. The main goal of the optimization process is to minimize the travel time prediction error. However, as it was mentioned earlier, the data produced by license plate recognition systems or traffic detectors are sometimes unavailable, so it means that if any input value for this model is missing the model will not work. Some of these systems are more reliable than others. Because of it we also decided to minimize the time for which the prediction of the SVR is not available and the number of input variables.

The reason why it is good to have multiple models with different characteristics is that we can switch among them during the real time prediction process according to a given situation. The main factor which is changing over the time is data availability from sensors. The key idea of our approach is to dynamicaly switch among these models according to which data are currently available. The highest-quality model is activated if possible. If the input data for this model are missing, the second best model is taken. If the input data for this model are not available, we will use the next model. We can continue with this process, until we get the model for which the data are available, or we have to stop because the data are unavailable.

## 7.3 Experimental results

The proposed method was evaluated using real world publicly available data from Seattle. We performed our experiments on two trajectories denoted as A and B. These trajectorise shown in the figure 4.1. We compared our new method to a simple license plate recognition method that was implemented. The results in terms of availability and RMSE are summarized in Table 7.1. Which gives median values from 80 runs for each experiment. It can be seen that our method provides better prediction, which is available for a longer period of time.

| **Method** | **Available** | **RMSE** |
|---|---|---|
| Section A | | |
| License plate method | 0.65 | 33.22 |
| Our method (median) | 0.99 | 25.23 |
| Section B | | |
| License plate method | 0.50 | 63.81 |
| Our method (median) | 0.99 | 52.47 |

Table 7.1: Comparission of our algorithm with a simple license plate approach (median values).

# Chapter 8

# Optimization of Meta-parameters and comparison with other methods

In our previous work, we have shown that it is possible to use a multi-objective optimization to find many SVR models for prediction of traffic variables. These SVR models differ in the set of input variables that they utilize. We optimized only the set of input variables, but the meta-parameters of SVR remained unchanged. However, this is far from the optimum, because for each SVR the optimal settings of the meta-parameters is different. In this chapter, we try to improve the prediction results by simultaneous search for the optimal data inputs for SVR and the optimal meta-parameters in one run of the multiobjective-genetic algorithm. Moreover, we will propose a parallel implementation of this method capable to significantly accelerate the optimization process.

## 8.1 Optimization of meta-parameters

In order to obtain high quality predictions, it is necessary to properly select the kernel function and set various parameters. Two types of kernels are supported in our work: linear SVR and SVR with radial basis kernel. For linear SVR it is necessary to optimize only the regularization coefficient $C$. The radial kernel requires, in addition to $C$, to optimize the kernel parameter $\gamma$.

The method is based on a multimodal and multiobjective NSGA-II algorithm. The whole chromosome is divided into two parts. The first part contains the information about used input variables. Each gene represents one potential input variable. The binary part of the chromosome also contains one additional bit, which defines the type of kernel (0 – the linear kernel; 1 – the radial kernel). We use the uniform crossover and bit flip mutations to modify this part of the chromosome. The second part of the chromosome consists of real values, which are devoted to SVR meta-parameters. The first real value defines the regularization coefficient in the case that linear SVR is used. In this case the value of regularisation coefficient is equal to $2^{C_{linear}}$.

The second parameter is the value of regularization coefficient in the case that radial kernel is used. In this case the regularisation coefficient is equal to $2^{C_{radial}}$. Finally, the third real value is for gamma parameter ($\gamma = 2^{gamma}$). We use SBX crossover and normally distributed mutations to modify the real valued part of the chromosome. The schema of the whole chromosome is depicted in Figure 8.1.

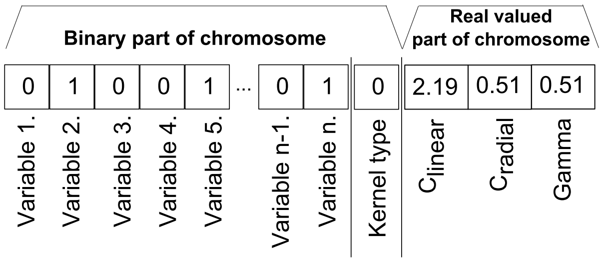


Figure 8.1: Chromosome scheme.

We evaluated our new method for simultaneous optimization of SVR inputs and meta-parameters on three different tasks. It was data imputation task, short term traffic prediction task, and travel time prediction task. We utilized the model switching described in previous Chapter to obtain the prediction results. The experiments shown that the mean improvement for data imputation is 2.29 % and for short time traffic forecasting with 15 minutes prediction horizon is 26.35 %. The method was compared with method, which optimizes only model inputs. New method provides only a small improvement for travel times estimation (the mean is 0.32 %).

### 8.1.1 Parallel implementation

There exist many approaches to parallelize genetic algorithms. The basic approach is known as the master-slave model. In this case, the master process performs all operators like selection, crossover and mutation. The only part of the algorithm which is conducted in parallel, is evaluation of the quality of candidate solutions. In our work we utilized this principle. Each computational unit, which is one processor core in our case, computes the values of fitness functions for some portion of the population.

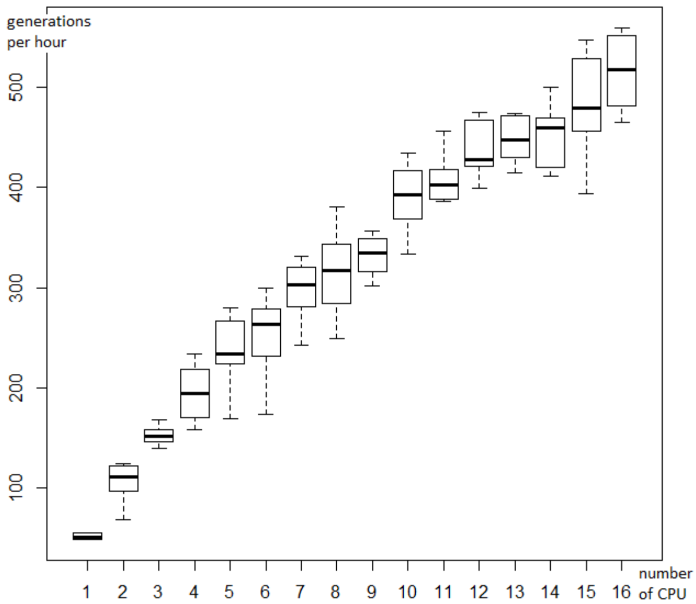


Figure 8.2: Speedup for short time traffic forecasting using parallel version of our method.

In order to show the speedup provided by parallel implementation, we performed 10 independent runs of our method for short time traffic prediction using a different number of processor cores. We measured the number of GA iterations (generations) performed in 1 hour. The results are shown in Figure 8.2.

The Figure shows the number of generations processed during one hour. The results are depicted in the form of box plots, because we have performed 10 independent runs for each number of cores. The population consists of 40 individuals. According to our experiments, it is possible to achieve speedup of almost 10 times when 16 processor cores are employed.

## 8.2 Comparison with other methods

In the area of machine learning, we usually split the available data into two distinct datasets. The first set is used for training and the second for evaluation of the prediction quality. This basic approach works well in simple scenarios. However, sometimes we need to know the prediction quality of the model before its learning is finished. For example, we need it to evaluate the termination condition in iterative algorithms, or we use it in the calibration and optimization of meta-parameters for machine learning algorithm. In this case, we need to split the available data into three groups. The first is utilized for training, the second is used in the termination condition or for calibration and the third is for final evaluation of the obtained model. This approach is called the cross-validation and we will use it in this chapter to compare our methods with other methods [34].

For each of the discused methods, we will split the data into a training set for learning and a test set for evaluation of the objective functions in genetic algorithm. After the model is finally created, the validation set is used to evaluate the prediction quality. The reason we use the validation set for comparison is that these data were not used for training and during the optimization process and they can provide us a fair comparison.

### 8.2.1 Data imputation

The Principal Component Analysis is often used for dimensionality reduction of the input data. However, after some modifications, it can be also utilized to estimate the missing values. In this chapter, we provide the results of the PCA methods for imputation of the missing traffic data and compare them with the proposed method, which utilizes SVR.

We performed the imputation using two variants of PCA method. The first is called Singular Value Decomposition [51] and the second is probabilistic principal component analysis (PPCA) [39]. The results for the imputation of traffic volume are shown in Table 8.1 - four sensors with the biggest mean volume on four places in the centre of Seattle are analysed. We used the data from 1st July to 15th July as the training set, 16th to 31st July as the test set and the data from August as the validation set. The RMSE in Table 8.1 is given for the validation set.

It can be observed that our method provides a better precision for almost every sensors. The exception are sensors number 1 on place 11 and number 8 on place 23, where Singular Value Decomposition and PPCA perform slightly better. For sensor 7 on place 23, the PPCA method also works better than our method. The results for occupancy under the same test scenario are provided in Table 8.2. In this case, our method based on the SVR is in most cases worse than Singular Value Decomposition (6 sensors out of 15) and PPCA (4 sensors out of 15). This might be caused by highrer noise in occupancy data. The rate between standard deviation and mean value is usually higher than for volume.

| Place | Sensor | Our Method | PCA method | | PPCA method | |
|---|---|---|---|---|---|---|
| | | RMSE | RMSE | Improvement | RMSE | Improvement |
| 11 | 1 | 6.70 | 6.46 | −3.72 % | 6.49 | −3.24 % |
| 11 | 3 | 6.50 | 6.93 | **6.20 %** | 6.70 | **2.99 %** |
| 11 | 13 | 6.94 | 6.94 | 0.00 % | 7.02 | **1.14 %** |
| 11 | 17 | 5.20 | 5.30 | **1.89 %** | 5.30 | **1.89 %** |
| 19 | 3 | 5.98 | 7.91 | **24.40 %** | 7.53 | **20.58 %** |
| 19 | 4 | 6.14 | 7.30 | **15.89 %** | 7.23 | **15.08 %** |
| 19 | 7 | 5.54 | 7.87 | **29.61 %** | 7.78 | **28.79 %** |
| 19 | 8 | 4.93 | 5.93 | **16.86 %** | 5.82 | **15.29 %** |
| 22 | 1 | 4.20 | 5.12 | **17.97 %** | 4.90 | **14.29 %** |
| 22 | 4 | 5.44 | 6.73 | **19.17 %** | 6.64 | **18.07 %** |
| 22 | 7 | 4.91 | 5.40 | **9.07 %** | 5.40 | **9.07 %** |
| 22 | 8 | 4.35 | 5.42 | **19.74 %** | 5.18 | **16.02 %** |
| 23 | 6 | 4.43 | 4.89 | **9.41 %** | 4.84 | **8.47 %** |
| 23 | 7 | 11.27 | 11.39 | **1.05 %** | 10.48 | −7.54 % |
| 23 | 8 | 10.95 | 10.86 | −0.83 % | 9.95 | −10.05 % |
| 23 | 9 | 5.07 | 5.98 | **15.22 %** | 5.83 | **13.04 %** |

Table 8.1: Results of the volume imputation compared with PCA and PPCA method (Seattle Sensys data).

| Place | Sensor | Our Method | PCA method | | PPCA method | |
|---|---|---|---|---|---|---|
| | | RMSE | RMSE | Improvement | RMSE | Improvement |
| 11 | 1 | 7.28 | 7.72 | **5.70 %** | 7.70 | **5.45 %** |
| 11 | 3 | 6.52 | 6.60 | **1.21 %** | 6.63 | **1.66 %** |
| 11 | 4 | 8.41 | 8.48 | **0.83 %** | 8.43 | **0.24 %** |
| 11 | 17 | 7.04 | 6.74 | −4.45 % | 6.77 | −3.99 % |
| 19 | 3 | 3.66 | 4.17 | **12.23 %** | 4.17 | **12.23 %** |
| 19 | 4 | 4.72 | 4.67 | −1.07 % | 4.68 | −0.85 % |
| 19 | 8 | 10.44 | 9.13 | −14.35 % | 8.97 | −16.39 % |
| 22 | 1 | 4.26 | 4.12 | −3.40 % | 4.11 | −3.65 % |
| 22 | 3 | 5.14 | 4.52 | −13.72 % | 4.52 | −13.72 % |
| 22 | 6 | 3.92 | 3.87 | −1.29 % | 3.87 | −1.29 % |
| 22 | 8 | 4.15 | 3.95 | −5.06 % | 3.96 | −4.80 % |
| 23 | 1 | 12.55 | 12.59 | **0.32 %** | 12.52 | −0.24 % |
| 23 | 2 | 10.27 | 10.25 | −0.20 % | 10.16 | −1.08 % |
| 23 | 3 | 11.61 | 11.45 | −1.40 % | 11.20 | −3.66 % |
| 23 | 7 | 21.35 | 20.89 | −2.20 % | 20.57 | −3.79 % |

Table 8.2: Results of the occupancy imputation compared with PCA and PPCA method.

### 8.2.2 Short Term Traffic Forecasting

To justify the proposed approach for traffic forecasting which is based on multi-objective genetic algorithms, we will compare it with the method utilizing only SVR. It is almost impossible to utilize SVR directly for traffic forecasting, because there are many missing values in the real world data. This is problematic because the SVR needs all inputs available. Hence, we have to fill these missing input data somehow. In our experiments, we utilized two simple techniques. The first technique fills the missing values by zero and the second one by the mean value of the given variable (computed from the previous samples).

| Place | Sensor | Our Method | Imputation by zero | | Imputation by mean | |
|---|---|---|---|---|---|---|
| | | RMSE | RMSE | Improvement | RMSE | Improvement |
| 11 | 1 | 7.50 | 8.01 | **6.37 %** | 8.08 | **7.18 %** |
| 11 | 3 | 8.12 | 8.47 | **4.13 %** | 8.78 | **7.52 %** |
| 11 | 13 | 7.88 | 9.10 | **13.41 %** | 9.85 | **20.00 %** |
| 11 | 17 | 6.81 | 9.36 | **27.24 %** | 10.75 | **36.65 %** |
| 19 | 3 | 7.76 | 10.35 | **25.02 %** | 11.46 | **32.29 %** |
| 19 | 4 | 7.83 | 10.98 | **28.69 %** | 10.65 | **26.48 %** |
| 19 | 7 | 7.78 | 9.86 | **21.10 %** | 11.34 | **31.39 %** |
| 19 | 8 | 6.64 | 8.71 | **23.77 %** | 8.02 | **17.21 %** |
| 22 | 1 | 6.23 | 6.35 | **1.89 %** | 5.72 | -8.92 % |
| 22 | 4 | 7.59 | 9.88 | **23.18 %** | 9.27 | **18.12 %** |
| 22 | 7 | 6.97 | 6.81 | -2.35 % | 7.30 | **4.52 %** |
| 22 | 8 | 6.46 | 7.59 | **14.89 %** | 8.04 | **19.65 %** |
| 23 | 6 | 6.04 | 6.95 | **13.09 %** | 6.01 | -0.50 % |
| 23 | 7 | 6.39 | 16.56 | **61.41 %** | 14.96 | **57.29 %** |
| 23 | 8 | 6.65 | 14.86 | **55.25 %** | 13.67 | **51.35 %** |
| 23 | 9 | 6.39 | 10.79 | **40.78 %** | 9.88 | **35.32 %** |

Table 8.3: Results of volume forecasting compared with imputation by the zero and mean.

It is important to note that our method based on the multi-objective GA utilizes many SVRs which differ in the input variables involved and these SVRs can be used according to which input data are currently available. Thus, it is not necessary to directly deal with missing data.

The results for traffic forecasting of traffic volume with the prediction horizon of 15 minutes are shown in Table 8.3. We employed the data sets in the same way as in the previous task.

The results have shown that our method performs better in almost all cases. The approach based on a simple SVR with the missing inputs filled with zeros performs slightly better only for one sensor. If the mean is used, 2 out of 15 cases are predicted more precisely.

We tested the same scenario of short term traffic forecasting for the occupancy. In this case, our method provide better prediction than simple SVR with missing inputs filled by zero as well as the mean for 10 out of 15 sensors. The results are shown in Table 8.4.

| Place | Sensor | Our Method | Imputation by zero | | Imputation by mean | |
|---|---|---|---|---|---|---|
| | | RMSE | RMSE | Improvement | RMSE | Improvement |
| 11 | 1 | 7.10 | 7.77 | **8.62%** | 7.80 | **8.97 %** |
| 11 | 17 | 6.49 | 8.27 | **21.52%** | 8.43 | **23.01 %** |
| 11 | 3 | 5.52 | 6.54 | **15.60%** | 6.59 | **16.24 %** |
| 11 | 4 | 8.76 | 8.74 | - 0.23% | 8.64 | -1.39 % |
| 19 | 3 | 3.80 | 4.65 | **18.28%** | 4.80 | **20.83 %** |
| 19 | 4 | 4.90 | 5.58 | **12.19%** | 5.46 | **10.26 %** |
| 19 | 8 | 9.09 | 12.65 | **28.14%** | 12.38 | **26.58 %** |
| 22 | 1 | 4.82 | 4.59 | - 5.01% | 4.60 | -4.78 % |
| 22 | 3 | 4.90 | 7.08 | **30.79%** | 7.28 | **32.69 %** |
| 22 | 6 | 4.59 | 4.24 | - 8.25% | 4.29 | -6.99 % |
| 22 | 8 | 4.33 | 4.43 | **2.26%** | 4.43 | **2.26 %** |
| 23 | 1 | 14.99 | 13.38 | -12.03% | 13.79 | -8.70 % |
| 23 | 2 | 11.98 | 11.08 | - 8.12% | 10.95 | -9.41 % |
| 23 | 3 | 13.72 | 13.95 | **1.65%** | 14.42 | **4.85 %** |
| 23 | 7 | 20.07 | 21.78 | **7.85%** | 22.05 | **8.98 %** |

Table 8.4: Results of occupancy forecasting compared with imputation by the zero and mean.

### 8.2.3 Travel Times Forecasting

The same approach, the same utilization of data and filling the missing values was implemented for the travel times forecasting. The results summarized in Table 8.5 show that our multi-objective method outperforms the simple SVR method in all test scenarios.

| Begin | End | Our Method | Imputation by zero | | Imputation by mean | |
|---|---|---|---|---|---|---|
| | | RMSE | RMSE | Improvement | RMSE | Improvement |
| Imputation by zero | | | | | | |
| Place 7 | Place 58 | 26.04 | 43.72 | **40.44%** | 31.92 | **18.42%** |
| Place 58 | Place 46 | 53.42 | 57.71 | **7.43%** | 79.74 | **33.01%** |
| Imputation by mean | | | | | | |
| Place 7 | Place 58 | 25.68 | 42.28 | **39.26%** | 34.69 | **25.97%** |
| Place 58 | Place 46 | 52.99 | 63.33 | **16.33%** | 59.78 | **5.61%** |

Table 8.5: Results of travel times forecasting compared with imputation by the zero and mean.

# Chapter 9

# Conclusions

The main objective of this thesis was to improve soft-computing methods for road traffic prediction by utilizing multiobjective evolutionary algorithms. The first goal of the thesis was to create a general framework for traffic prediction and travel times estimation. This framework should internally use a multi-objective optimization (such as evolutionary algorithms) in order to provide good trade-offs between various conflicting objectives the users typically formulate in this domain. Moreover, this framework should be capable of working with real world traffic data, which often contain a huge portion of missing values.

To fulfill this goal, we proposed a prediction framework, which internally utilizes SVR-based prediction. The meta-parameters and inputs of SVR are simultaneously optimized by a multi-objective genetic algorithm. The multimodal-NSGAII algorithm is used to perform this optimization task. We choose the RMSE of prediction, the number of SVR inputs and the portion of time in which the SVR can not be used for prediction (because of missing data) as the objective functions. The multi-objective optimization provided us with many solutions (SVRs), which differ in values of the objective functions. We typically obtained solutions showing very small prediction errors (RMSE), but requiring many input variables, but they often cannot be used because of missing input data. On the other hand, we obtained solutions with a higher prediction error which utilize only a few input variables. We also got many compromise solutions between these two extremes.

The reason why it is good to have multiple models with different characteristics is that we can switch among them during the real time prediction process. The main factor which is changing over the time is the data availability from sensors or camera. The key idea of the approach developed in this thesis is to dynamicaly switch among these models according to which data are currently available. The highest-quality model is activated if possible. If the input data for this model are missing, the second best model is taken. If the input data for this model are not available, we will use the next model. We can continue with this process, until we get the model for which the data are available, or we have to stop because the data are unavailable.

The second goal of this thesis was to evaluate this framework on real world case studies. We compared our prediction framework with the current state of the art methods. In order to provide a fair comparison, we utilized a validation data set containing new data which were used neither during learning nor the optimization process. Results were provided for three different tasks: data imputation, short term traffic forecasting and estimating of travel times.

In the case of data imputation, we compared our framework with Singular Value Decomposition and Probabilistic Principal Component Analysis. The results shown that our method provides better results than these methods in the imputation of volume and provides slightly worse results for the imputation of occupancy. For the short time traffic prediction, we have compared our method with a single SVR. The missing inputs of this SVR were filled by zero or the mean values of the missing variable. In this case, our multi-objective method performed better than simple SVR in almost all cases. The last task was the estimation of travel times. In this case, we also used a simple SVR with the missing input values filled with zero or the mean. Here, our multi-objective method outperformed the single SVR for all test road segments.

## 9.1 Conference Publications

- **Petrlik Jiri, Korcek Pavol, Fucik Otto, Beszedes Marian and Sekanina Lukas. Estimation of traffic density map using evolutionary algorithm. In: Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems. Anchorage: IEEE Intelligent Transportation Systems Society, 2012, pp. 632-637. ISBN 978-1-4673-3062-6**

- Petrlik Jiri and Sekanina Lukas. Multiobjective Evolution of Multiple-Constant Multipliers. In: Proceedings of the 18th International Conference on Soft Computing (MENDEL2012). Brno: Faculty of Mechanical Engineering BUT, 2012, pp. 64-69. ISBN 978-80-214-4540-6.

- Petrlik Jiri and Sekanina Lukas. Multiobjective evolution of approximate multiple constant multipliers. In: IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems 2013. Brno: IEEE Computer Society, 2013, pp. 116-119. ISBN 978-1-4673-6133-0.

- **Petrlik Jiri, Fucik Otto and Sekanina Lukas. Multiobjective Selection of Input Sensors for Travel Times Forecasting Using Support Vector Regression. In: 2014 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems Proceedings. Piscataway: Institute of Electrical and Electronics Engineers, 2014, pp. 14-21. ISBN 978-1-4799-4498-9.**

- **Petrlik Jiri, Fucik Otto and Sekanina Lukas. Multiobjective Selection of Input Sensors for SVR Applied to Road Traffic Prediction. In: Parallel Problem Solving from Nature - PPSN XIII. Heidelberg: Springer Verlag, 2014, pp. 802-811. ISBN 978-3-319-10761-5.**

- **Petrlik Jiri and Sekanina Lukas. Towards Robust and Accurate Traffic Prediction Using Parallel Multiobjective Genetic Algorithms and Support Vector Regression. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems. Los Alamitos: IEEE Computer Society, 2015, pp. 2231-2236. ISBN 978-1-4673-6596-3.**

## 9.2 Software

- Petrlik Jiri: SVM Feature Selection System, software, 2013

## 9.3 Research Projects and Grants

- Natural Computing on Unconventional Platforms, GACR, GAP103/10/1517, 2010-2013, completed
- The IT4Innovations Centre of Excellence, MĹ MT, ED1.1.00/02.0070, 2011-2015, running
- Verification and Optimization of Computer Systems, VUT Brno, FIT-S-12-1, 2012-2014, completed
- Research and development focused on monitoring and management of lorry movement on lower class road network in the Czech Republic, TACR, TA02030841, 2012-2014, completed
- Architecture of parallel and embedded computer systems, VUT Brno, FIT-S-14-2297, 2014-2016, running
- Advanced Methods for Evolutionary Design of Complex Digital Circuits, GACR, GA14-04197S, 2014-2016, running
- Verification of the implementation of continuous traffic load map using modern classification and prediction methods, TACR, TA02030915, 2012-2014, completed

# Bibliography


[1] RDE Home Page. `http://www.its-rde.net/`. Accessed: 2014-10-5.

[2] Thomas Back and Martin Schutz. Intelligent mutation rate control in canonical genetic algorithms. In ZbigniewW. Ras and Maciek Michalewicz, editors, *Foundations of Intelligent Systems*, volume 1079 of *Lecture Notes in Computer Science*, pages 158–167. Springer Berlin Heidelberg, 1996.

[3] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):pages 121–167, 1998.

[4] Lijuan Cao. Support vector machines experts for time series forecasting. *Neurocomputing*, 51:pages 321–339, 2003.

[5] Manoel Castro-Neto, Young-Seon Jeong, Myong-Kee Jeong, and Lee D Han. Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert systems with applications*, 36(3):pages 6164–6173, 2009.

[6] Mashrur Chowdhury and Adel W Sadek. Advantages and limitations of artificial intelligence. *Artificial Intelligence Applications to Critical Transportation Issues*, 6:pages 6–8, 2012.

[7] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001. ISBN: 9780471873396.

[8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):pages 182–197, 2002.

[9] Kalyanmoy Deb and A Raji Reddy. Reliable classification of two-class cancer data using evolutionary algorithms. *BioSystems*, 72(1):pages 111–129, 2003.

[10] Abhijit Dharia and Hojjat Adeli. Neural network model for rapid forecasting of freeway link travel time. *Engineering Applications of Artificial Intelligence*, 16(7):pages 607–613, 2003.

[11] Abhijit Dharia and Hojjat Adeli. Neural network model for rapid forecasting of freeway link travel time. *Engineering Applications of Artificial Intelligence*, 16(7-8):607 – 613, 2003.

[12] Ailine Ding, Xangmo Zhao, and LiCheng Jiao. Traffic flow time series prediction based on statistics learning theory. In *IEEE 5th International Conference on Intelligent Transportation Systems*, pages 727–730. IEEE, 2002.

[13] Lili Du, Srinivas Peeta, and Yong Hoon Kim. An adaptive information fusion model to predict the short-term link travel time distribution in dynamic traffic networks. *Transportation Research Part B: Methodological*, 46(1):pages 235–252, 2012.

[14] Kaibo Duan, S Sathiya Keerthi, and Aun Neow Poo. Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, 51:pages 41–59, 2003.

[15] Jan Fabian Ehmke, Stephan Meisel, and Dirk Christian Mattfeld. Floating car based travel times for city logistics. *Transportation research part C: emerging technologies*, 21(1):pages 338–352, 2012.

[16] A.E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):pages 124–141, Jul 1999.

[17] BD Greenshields, Ws Channing, Hh Miller, et al. A study of traffic capacity. In *Highway research board proceedings*, volume 1935. National Research Council (USA), Highway Research Board, 1935.

[18] R. Hinterding. Gaussian mutation and self-adaption for numeric genetic algorithms. In *IEEE International Conference on Evolutionary Computation.*, volume 1, page 384. IEEE, Nov 1995.

[19] Wei-Chiang Hong. Traffic flow forecasting by seasonal svr with chaotic simulated annealing algorithm. *Neurocomputing*, 74(12):pages 2096–2107, 2011.

[20] Maarten Houbraken, Pieter Audenaert, Didier Colle, Mario Pickavet, Karolien Scheerlinck, Isaak Yperman, and Steven Logghe. Real-time traffic monitoring by fusing floating car data with stationary detector data. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on*, pages 127–131. IEEE, 2015.

[21] Cheng-Lung Huang and Chieh-Jen Wang. A ga-based feature selection and parameters optimizationfor support vector machines. *Expert Systems with applications*, 31(2):pages 231–240, 2006.

[22] Jeffrey Huang, Xuhui Shao, and Harry Wechsler. Face pose discrimination using support vector machines (svm). In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 1, pages 154–156. IEEE, 1998.

[23] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32(10):pages 2513–2522, 2005.

[24] K. Kanayama, Y. Fujikawa, K. Fujimoto, and M. Horino. Development of vehicle-license number recognition system using real-time image processing and its application to travel-time measurement. In *Vehicular Technology Conference, 1991. Gateway to the Future Technology in Motion., 41st IEEE*, pages 798–804. IEEE, May 1991.

[25] MG Karlaftis and EI Vlahogianni. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, 19(3):pages 387–399, 2011.

[26] James H Kell, Iris J Fullerton, and Milton K Mills. *Traffic detector handbook*. 1990. ISBN: 9781420067187.

[27] Arne Kesting. *Traffic Flow Dynamics: Data, Models and Simulation*. Springer, 2012. ISBN: 9783642324604.

[28] Howard R Kirby, Susan M Watson, and Mark S Dougherty. Should we use neural networks or statistical models for short-term motorway traffic forecasting. *International Journal of Forecasting*, 13(1):pages 43–50, 1997.

[29] Lawrence A Klein, Milton K Mills, and David RP Gibson. *Traffic Detector Handbook: -Volume II*. 2006.

[30] Ming-Wei Li, Wei-Chiang Hong, and Hai-Gui Kang. Urban traffic flow forecasting using gauss–svr with cat mapping, cloud model and pso hybrid algorithm. *Neurocomputing*, 99:pages 230–240, 2013.

[31] Q Li, T Zhang, and Y Yu. Using cloud computing to process intensive floating car data for urban traffic surveillance. *International Journal of Geographical Information Science*, 25(8):pages 1303–1322, 2011.

[32] Pao-Tsun Lin, Shun-Feng Su, and Tsu-Tian Lee. Support vector regression performance analysis and systematic parameter selection. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, volume 2, pages 877–882. IEEE, 2005.

[33] Junshui Ma, James Theiler, and Simon Perkins. Accurate on-line support vector regression. *Neural Computation*, 15(11):pages 2683–2703, 2003.

[34] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC Press, 2011. ISBN: 9781420067187.

[35] JC Miles and Andrew J Walker. The potential application of artificial intelligence in transport. In *IEEE Proceedings-Intelligent Transport Systems*, volume 153, pages 183–198. IEEE, 2006.

[36] A. Padiath, L. Vanajakshi, S.C. Subramanian, and H. Manda. Prediction of traffic density for congestion analysis under indian traffic conditions. In *12th International IEEE Conference on Intelligent Transportation Systems.*, pages 1–6, Oct 2009.

[37] Dongjoo Park and Laurence R Rilett. Forecasting freeway link travel times with a multilayer feedforward neural network. *Computer-Aided Civil and Infrastructure Engineering*, 14(5):pages 357–367, 1999.

[38] Fengxiang Qiao, Hai Yang, and William HK Lam. Intelligent simulation and prediction of traffic flow dispersion. *Transportation Research Part B: Methodological*, 35(9):pages 843–863, 2001.

[39] Li Qu, Li Li, Yi Zhang, and Jianming Hu. Ppca-based missing data imputation for traffic flow volume: a systematical approach. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):pages 512–522, 2009.

[40] Sridhar Ramaswamy, Pablo Tamayo, Ryan Rifkin, Sayan Mukherjee, Chen-Hsiang Yeang, Michael Angelo, Christine Ladd, Michael Reich, Eva Latulippe, Jill P Mesirov, et al. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, 98(26):pages 15149–15154, 2001.

[41] Henry Roncancio, André Carmona Hernandes, and Marcelo Becker. Vision-based system for pedestrian recognition using a tuned svm classifier. In *Engineering Applications (WEA), 2012 Workshop on*, pages 1–6. IEEE, 2012.

[42] Grzegorz Rozenberg, Thomas Bck, and Joost N Kok. *Handbook of natural computing*. Springer Publishing Company, Incorporated, 2011. 9783540929116.

[43] Rozvoj modernich dopravnich inteligentnich systemu. www.romodis.cz. `http://www.romodis.cz`.

[44] Brian L Smith and Michael J Demetsky. Short-term traffic flow prediction: neural network approach. *Transportation Research Record*, (1453), 1994.

[45] Antony Stathopoulos, Loukas Dimitriou, and Theodore Tsekeris. Fuzzy modeling approach for combined forecasting of urban traffic flow. *Computer-Aided Civil and Infrastructure Engineering*, 23(7):pages 521–535, 2008.

[46] Shiliang Sun, Rongqing Huang, and Ya Gao. Network-scale traffic modeling and forecasting with graphical lasso and neural networks. *Journal of Transportation Engineering*, 2012.

[47] S. Takaba, T. Morita, T. Hada, T. Usami, and M. Yamaguchi. Estimation and measurement of travel time by vehicle detectors and license plate readers. In *Vehicle Navigation and Information Systems Conference, 1991*, volume 2, pages pages 257–267, Oct 1991.

[48] Man-Chun Tan, SC Wong, Jian-Min Xu, Zhan-Rong Guan, and Peng Zhang. An aggregation approach to short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):pages 60–69, 2009.

[49] Francis EH Tay and Lijuan Cao. Application of support vector machines in financial time series forecasting. *Omega*, 29(4):pages 309–317, 2001.

[50] Luis Torgo. *Data mining with R: learning with case studies*. Chapman & Hall/CRC, 2010. 9781439810187.

[51] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):pages 520–525, 2001.

[52] Ambros Valach, Tecl and Vyskocilova. Vyse ztrat z dopravni nehodovosti na pozemnich komunikacich za rok 2013. `http://www.czrso.cz/clanky/vyse-ztrat-z-dopravni-nehodovosti-na-pozemnich-komunikacich-za-rok-2013/`. Accessed: 2015-10-20.

[53] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. *Transportation Research Part C: Emerging Technologies*, 13(3):pages 211–234, 2005.

[54] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:pages 3–19, 2014.

[55] Chih-Hung Wu, Gwo-Hshiung Tzeng, Yeong-Jia Goo, and Wen-Chang Fang. A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert systems with applications*, 32(2):pages 397–408, 2007.

[56] Chih-Hung Wu, Gwo-Hshiung Tzeng, and Rong-Ho Lin. A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert Systems with Applications*, 36(3):pages 4725–4735, 2009.

[57] Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. Travel-time prediction with support vector regression. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):pages 276–281, 2004.

[58] Chun-Hsin Wu, Jan-Ming Ho, and D.T. Lee. Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems,*, 5(4):pages 276–281, Dec 2004.

[59] Chun-Hsin Wu, Da-Chun Su, Justin Chang, Chia-Chen Wei, Jan-Ming Ho, Kwei-Jay Lin, and DT Lee. An advanced traveler information system with emerging network technologies. In *Proc. 6th Asia-Pacific Conf. Intelligent Transportation Systems Forum*, pages 230–231, 2003.

[60] Chun-Hsin Wu, Da-Chun Su, Justin Chang, Chia-Chen Wei, Kwei-Jay Lin, and Jan-Ming Ho. The design and implementation of intelligent transportation web services. In *IEEE International Conference on E-Commerce*, pages 49–52. IEEE, 2003.

[61] Cleber Zanchettin, Byron Leite Dantas Bezerra, and Washington W Azevedo. A knn-svm hybrid model for cursive handwriting recognition. In *The 2012 International Joint Conference on Neural Networks (IJCNN),*, pages 1–8. IEEE, 2012.

[62] Weizhong Zheng, Der-Horng Lee, and Qixin Shi. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of transportation engineering*, 132(2):pages 114–121, 2006.