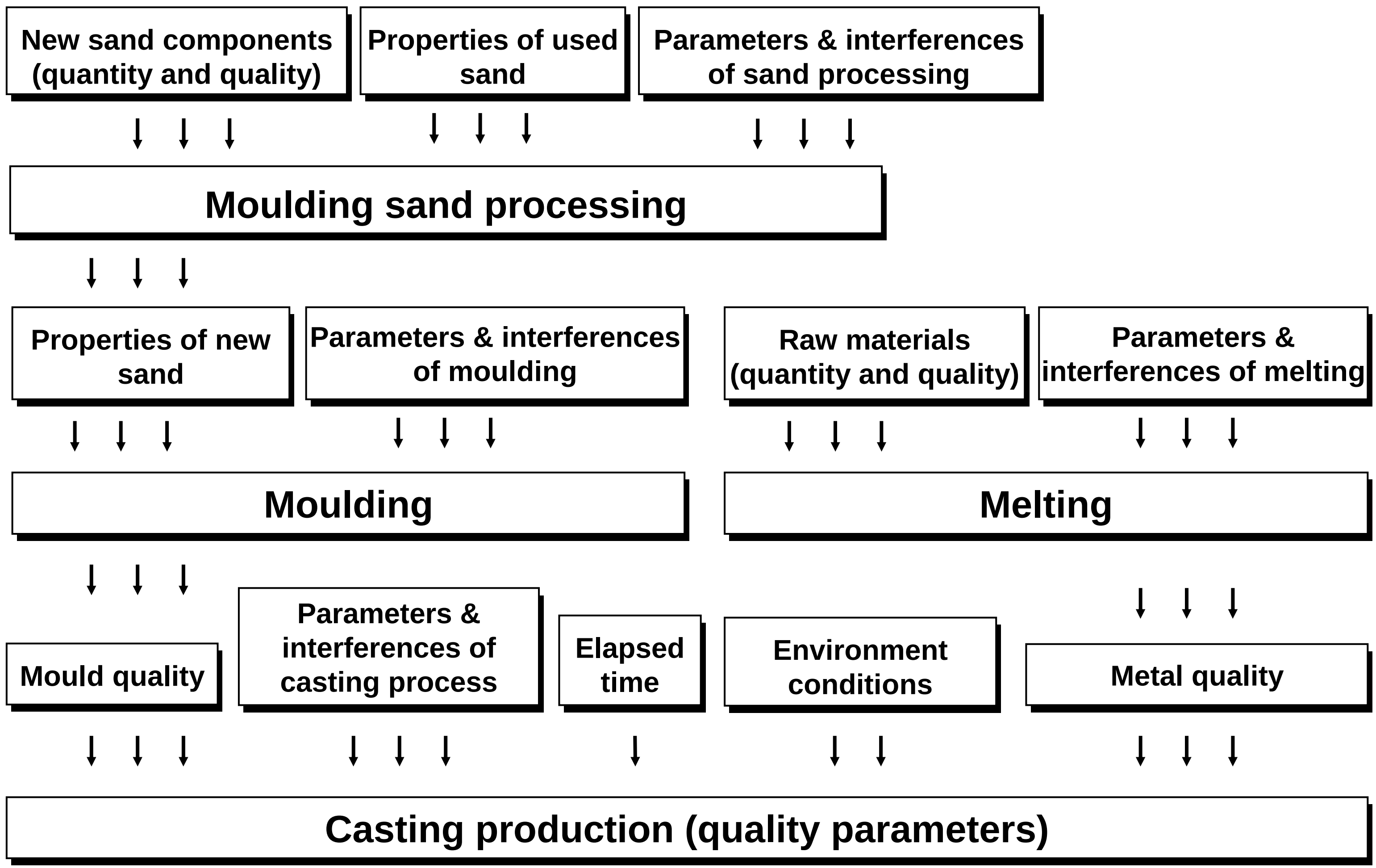
METRO
MEtallurgical TRaining On-line

Computer methods for analysis and control of production processes

Marcin Perzyk
WUT

Education and Culture

Main production processes in a typical foundry



Types of tasks related to production processes, aided by computer techniques

- Prediction of a process run by simulation
- Current monitoring and control of processes
- Data mining (analysis) for detection of hidden regularities in processes

Prediction of a process run by simulation

- Main applications - process design and planning stages (e.g. numerical simulation of metal flow assists pouring system design)
- Also facilitates prediction of consequences of changes in manufacturing process or production management system

Simulation requires development of a process *model*

Current monitoring and control of processes

- Includes statistical process control methods (SPC) and strategies or methodologies (e.g. Six Sigma).
- *SPC* facilitates detection of process irregularities and troubles as well as assessment of the process capabilities.
- SPC methods serve as an assisting tool only for the process irregularities diagnosis which has to be done by the company employees.
- *Methodologies* indicate methods and organisational issues required for design and maintenance of correct processes.

The above issues are presented in a separate lecture

Detection of regularities in production processes

- Enables identification of causes of process troubles (e.g. appearance of defects in final or intermediate products)
- Facilitates indication of optimal or critical process parameters (e.g. combinations of time and temperature for heat treatment)
- Utilises modern methods of data analysis, including data mining

Requires development of a process *model*

Process modelling
Basic concepts

- *Model* is a simplified object which behaves, from the viewpoint of the studied phenomena, as the real one.
- There are two basic *types of models*: physical and mathematical.
- *Physical model* makes use of a physical similarity between the real object and the model which is a type of experimental test. The similarity theory gives the framework for constructing such models.
- *Mathematical model* is a mathematical equation or set of equations or another type relationship describing given phenomenon or process.

Vast growth of computational methods made the mathematical (mainly numerical) modelling dominant

Process modelling
Main types of mathematical models

- **Relationships taking into account the nature of a phenomenon or process** (e.g. physical laws governing the heat transfer between casting and mould). Applied mainly for modelling of the casting formation, i.e. pouring, solidification and cooling-induced stress development.

  Are topics of another lectures within the METRO course.

- **Models of the ‘black box’ type, i.e. ignoring the physical nature of the processes**. Can be applied to all production processes.

  Are a subject of this lecture

Models ignoring the physical nature of the processes

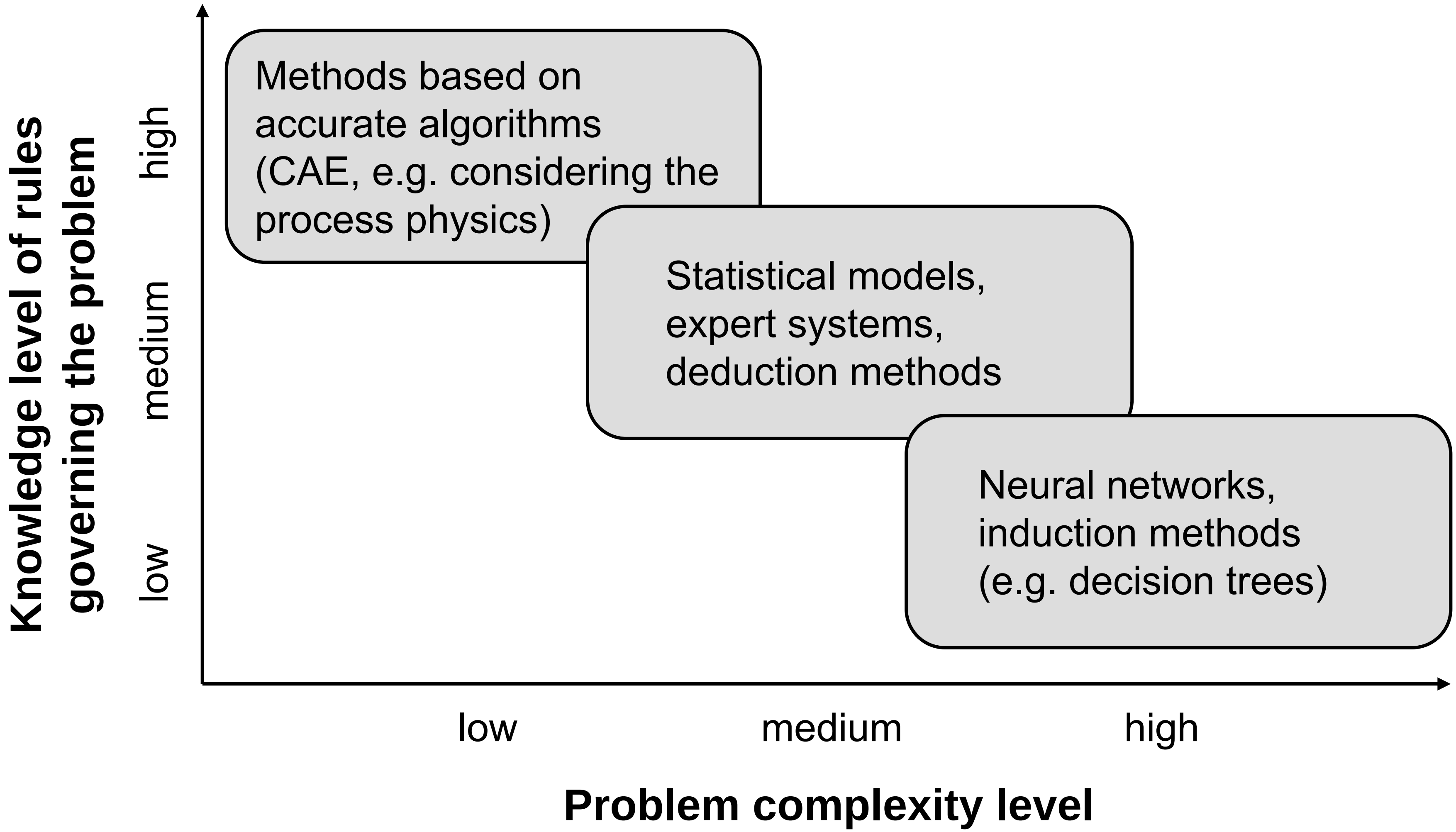
- Model constants (parameters) are determined from the experimental data, usually collected during normal production, only sometimes obtained from especially planned experiments.
- Various types of optimisation methods and algorithms are used, dependent on the kind of the model.
- Fully defined model, i.e. with determined values of its parameters, can be further used for prediction of the output values (dependent variables) for new values of the input values (or their combinations) i.e. not appeared in the data sets used for determination of the model parameters.

Main types of models ignoring the physical nature of the processes

- Statistical type
- Utilising artificial intelligence methods, in particular learning systems (machine learning methods)

<u>Definition</u>
A system can be treated as ‘learning’ if the changes leading to improvement of its functioning occur due to presenting to it experimental results (in a form of training examples).

Application areas of different types of models (after R. Tadeusiewicz)



Types of data used in process modelling

Data appearing in a model (sometimes called attributes) can be of the following types:

- Nominal, i.e. being an element of a finite set of disordered discrete values, called categories.

  Often those values are expressed verbally, e.g. melting furnace can be: ‘electrical induction’, ‘electrical resistance’ or ‘gas-fired’; a product (casting) can be: ‘good’ or ‘defective’.

- Ordinal,i.e. being an element of a denumerable set of ordered discrete values.

  Example: temperature can be: ‘low’, ‘medium’, ‘high’.

- Numerical continuos, having real values.

Types of data used in process modelling

Some models require discrete types of data (nominal or ordinal). However, continuos types can be also handled by such models in the following way:

- Continuos variables can be changed to categorical ones by assignment of their current values to an appropriate interval, denoted by its ordinal number.
- If the output categories were defined in that way, a conversion of the obtained categories to real numbers can be done by replacing them by the average value of the interval.

Types of data used in process modelling

Replacement of continuos variables by discrete ones (i.e. nominal or ordinal, should take into account that:

- small number of categories (intervals) decreases precision of the model, both in the training and interrogating procedures
- large number of categories makes it likely that some of them will be not represented in the training set or the representations will be very small.

Main types of tasks performed by models of production processes

- Regression (approximation of a function)

  Fitting an analytical relationship (formula) between continuos input variables and a continuos output variable to a set of experimental points. *Example:* strength of an alloy in a function of its composition.

- Classification

  Assignment of results to one of predefined classes (categories, i.e. nominal or ordinal values), represented in the output variable. *Example:* classification of a product as ‘good’ or ‘faulty’.

- Detection of regularity

  Detection of important characteristics in the input data, without any knowledge about existing patterns.
  Example of application: grouping of shapes of parts in mechanical design.

Statistical models
Elementary information

- Statistical models perform regression type tasks (function approximation)
- General form of the function:

$$y = f(x_1, x_2, \ldots\ a_1, a_2, \ldots, a_n)$$

x_i – independent variables,

y – dependent variable,

a_j – constants (function parameters), which must be determined from the experimental data

- Particular form of the function *must be assumed*

Statistical models
Types of used functions

- Single or multivariable functions (one or multidimensional regression)
- Linear and nonlinear functions (linear and nonlinear regression)
  - Nonlinear functions:
    - polynomials (of arbitrary order)
    - other functions (e.g. power, exponential etc.)

Choice of the type of function is usually carried out after plotting the experimental data. Significant difficulties occur for multivariable functions.

Statistical models
Determination of model parameters

- Criterion of the model's *minimum error ε,* defined as a sum of squares of deviations form the observed (real) values:

$$\varepsilon = \sum_{k=1}^{n} \left(y_k - d_k \right)^2$$

  where n denotes number of experimental examples, y_k – values calculated (predicted) from the function (dependent on its parameters a_i), and d_k – experimental output values

- Analytical (unique) methods of determination of the parameters are available only for linear or polynomial types of functions.
- For other types one can employ:
  – linearisation of the function (most common), or
  – other optimisation methods of the function parameters (seldom).

Linear and polynomial statistical models
Determination of parameters

A set of linear equations can be obtained in the following procedure:

- differentiate of the error formula:

$$\left(f\left(x_1, x_2, \ldots \quad a_1, a_2, \ldots, a_n\right)_k - d_k\right)^2$$

  successively, in respect of all parameters a_i
- equate the calculated partial derivatives to zero
- sum up the obtained equations for all k experimental examples.

The obtained set contains n equations. Minimal number of examples permitting determination of the parameters equals n.

Nonlinear statistical models
Determination of parameters

- Examples of the linearisation of a relationship by introducing a new variable (*X* or *Y*):

$$y = \frac{x}{a_1 \cdot x + a_2} \quad \Rightarrow \quad Y = a_1 \cdot x + a_2 \qquad Y = \frac{x}{y}$$

$$y = a_1 \cdot x^{a2} \quad \Rightarrow \quad Y = log(a_1) + a_2 \cdot X \qquad X = log(x), \quad Y = log(y)$$

- Other optimisation methods (minimisation of model's error) will be treated in the lecture on artificial neural networks and also in the genetic algorithms section, further in this lecture.

Main production processes models utilising artificial intelligence methods

- Artificial neural networks

  Most often applied and of highest capabilities. A subject of another lecture

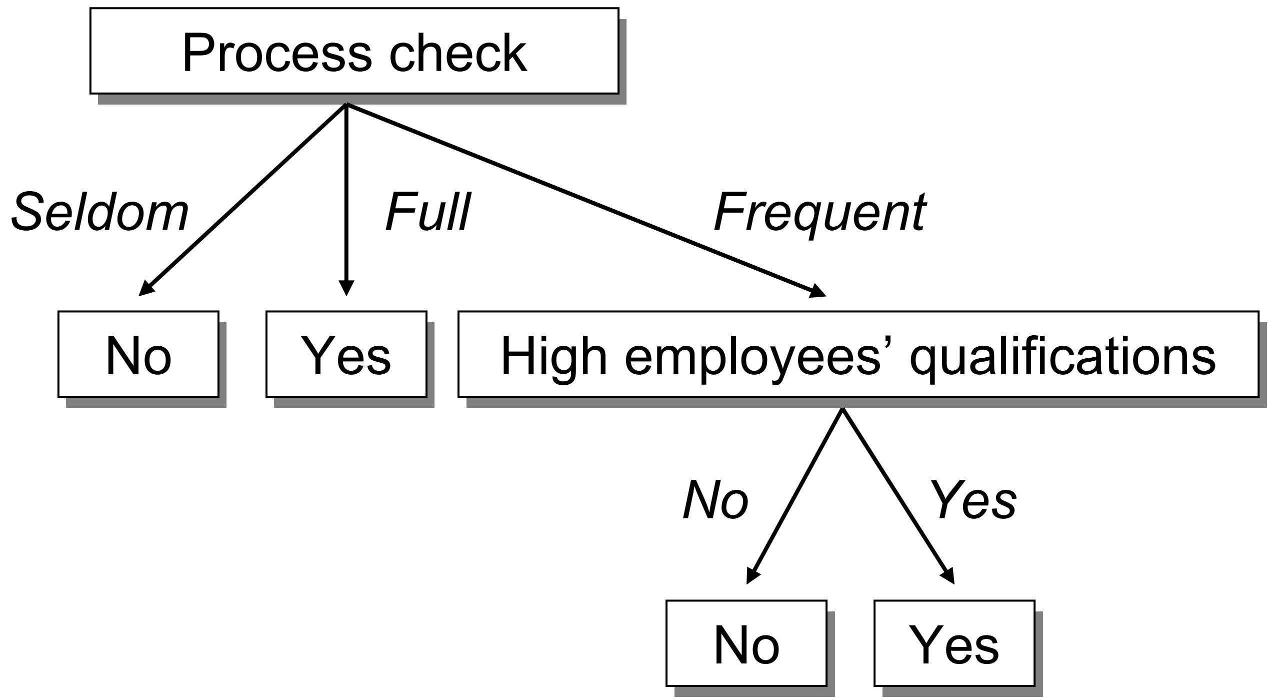
- Decision trees
- Models utilising fuzzy logic or fuzzy calculus
- Models utilising genetic optimisation
- Bayessian classification methods
- MARSplines (Multivariate Adaptive Regression Splines)

Decision trees
Introductory information

Decision tree is a model using basically nominal or ordinal type of data. It is a logical structure represented by a graph, which includes elements described below.

- *Root* is an origin of a tree, from which at least two *branches* go out (at least two), to the *knots* situated at a lower level.
- Every knot is associated with a *test* which checks values of attributes describing the input data (used for training or for interrogating the system).
- For each possible results of the test a branch leads to a knot situated at a lower level.
- Knots from which no branches originate, called *leaves*, have *classes* assigned to them.

Decision trees
Simple exemplary structure



This decision tree is for a quantity (concept) called 'low defectiveness level', which depends on several production parameters, all described in detail in a table on the next slide.

Decision trees
Training data used for generation of the exemplary tree

Example number	Process check	High employees' qualifications	Size of company	Low defectiveness level
1	frequent	yes	large	yes
2	frequent	no	small	no
3	full	yes	small	yes
4	seldom	yes	large	yes
5	rzadka	yes	small	yes
6	frequent	yes	small	yes
7	seldom	no	small	yes
8	frequent	no	large	yes

Decision trees
Generation (induction)

- A number of algorithms can be used is used for generation of decision trees from the collected (experimental) data. Specialised computer software is available.
- The presented exemplary tree has been generated by the classic ID3 algorithm.

  As a result it appeared that, for example, that the optimal attribute for the tree root is the type of process check and that the size of company has no influence on the defectiveness level.
- Other, newer algorithms enable handling missing and ambiguous data as well as simplifying (pruning) too complex trees.

Decision trees

Basic types of tasks and applications

- Main and classical type of task performed by decision trees is classification.
- Regression type tasks (approximation of continuous variables function) are also possible.

Use of continuous attributes requires assumption about limits of their values intervals for tests performed in the knots. Results are usually in a form of inequalities, which is different from statistical models.

Decision trees

Boosted type trees

- For difficult tasks, predictions generated by *sequences* of simple trees are closer to real values then predictions made by single complex tree.
- *Boosting* is a technique which applies a sequence of simple models generated in such way that each successive model assigns a greater weight to those observations (training examples) which have been wrongly classified by previous models.
- Boosting trees are better in modelling complex relationships but are more difficult in interpretation of the results and require more computational power.

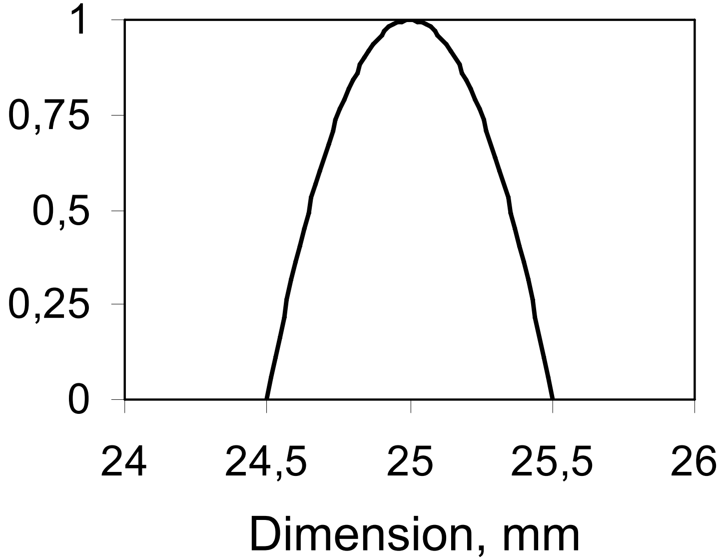
Models utilising fuzzy logic or fuzzy calculus

Basic concepts

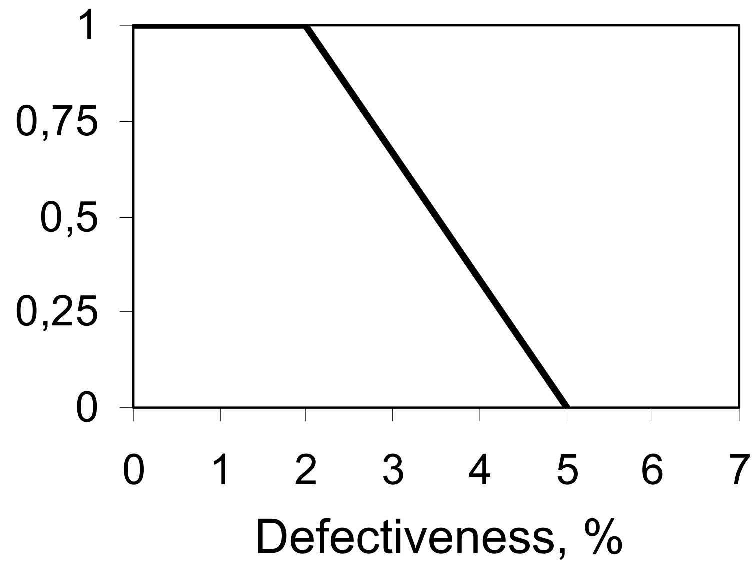
- *Fuzzy sets* and *fuzzy numbers* are used in order to express imprecise quantities.
- Imprecise value is defined not only by an *interval* of its possible real values (e.g. dimension ‘about 25 mm’ may be within range from 24,5 to 25,5 mm), but also by so called *membership function* or *preference function.* It describes degree of desire that the variable is equal to the given real value from the range.
- Membership (to the fuzzy set) function or, in other words, preference function may have values between 0 and 1 and is defined within possible range of the variable, so called universe of discourse.

Models utilising fuzzy logic or fuzzy calculus

Examples of membership function



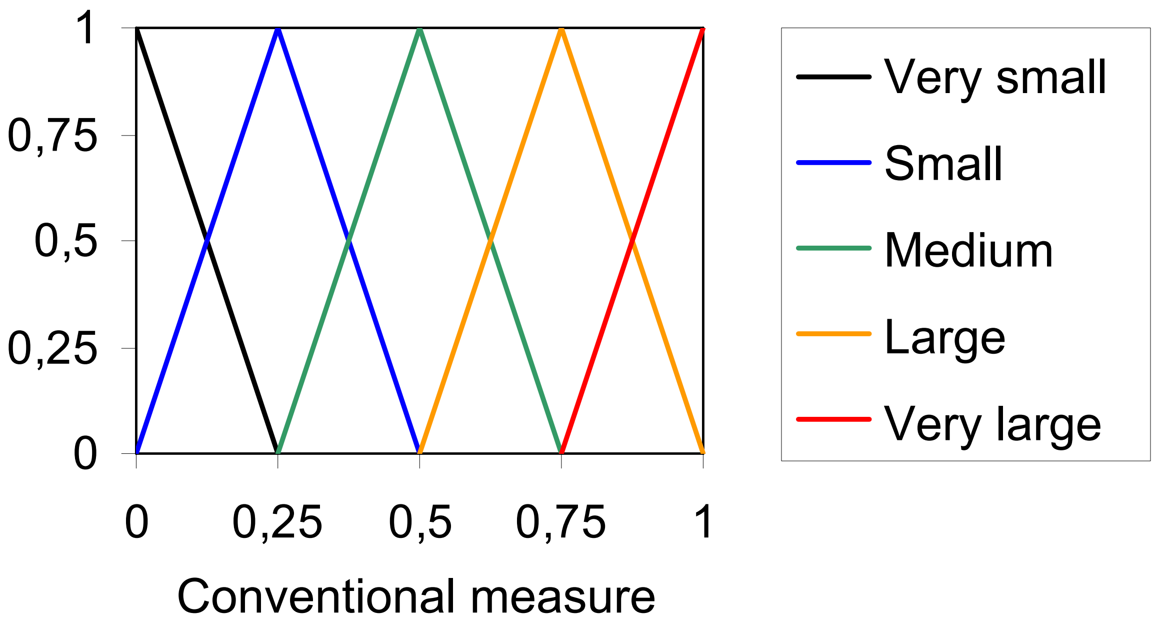
Exemplary membership (preference) function for dimension ‘about 25 mm’



Exemplary membership (preference) function for defectiveness level defined as ‘low’

Models utilising fuzzy logic or fuzzy calculus

Examples of membership function



Often used triangular membership (preference) functions expressing imprecise values of arbitrary variable (e.g. process parameter)

Models utilising fuzzy logic or fuzzy calculus

- Fuzzy logic is a modern and important field of mathematics.
- Most common applications of fuzzy logic are *fuzzy controllers*, widely used in automated systems.
- Fuzzy controller requires fuzzy logic *rule base*.

  Example: if temperature is ‘high’ and air humidity is ‘medium’ then the power setting (of air conditioner) is ‘high’.
- The rules can be designed or extracted (created) from numerical results of observations.
- The rules can be also created using *learning systems* such as artificial neural networks.This method resulted in construction of *neuro-fuzzy controllers*, widely used in industry.

Models utilising fuzzy logic or fuzzy calculus

- Fuzzy numbers can replace ‘ordinary’ crisp values in some analytical relationships, thus making more realistic models and offering new interpretation possibilities.
- Imprecise quantities appear in engineering often in various situations in design of products and processes.
- Application of fuzzy numbers to determination of the performance index, used for evaluation of design versions, will be shown.

Models utilising fuzzy numbers
Example of application

- Classic (crisp) performance index of design version j of process or product is defined as:

$$\gamma_j = \sum_{i=1}^{n} P_{ij} \cdot \alpha_i$$

where P_{ij} – value of parameter i for design version j, α_i – importance (weight) of parameter i for evaluation.

- Fuzzy numbers can replace the above crisp values appearing in the above formula, e.g. defined verbally as ‘very small’, ‘small’, etc. with corresponding preference functions as presented on the last chart.

Models utilising fuzzy numbers
Example of application

The triangle-type preference function is defined by a triplet of real numbers, determining its universe of discourse limits l and p (function values = 0) and maximum m (function value = 1). Basic arithmetic operations for that type of preference functions are simple:

- a sum of two fuzzy numbers x and y is also a triangle type fuzzy number defined by a triplet of real numbers calculated as follows:

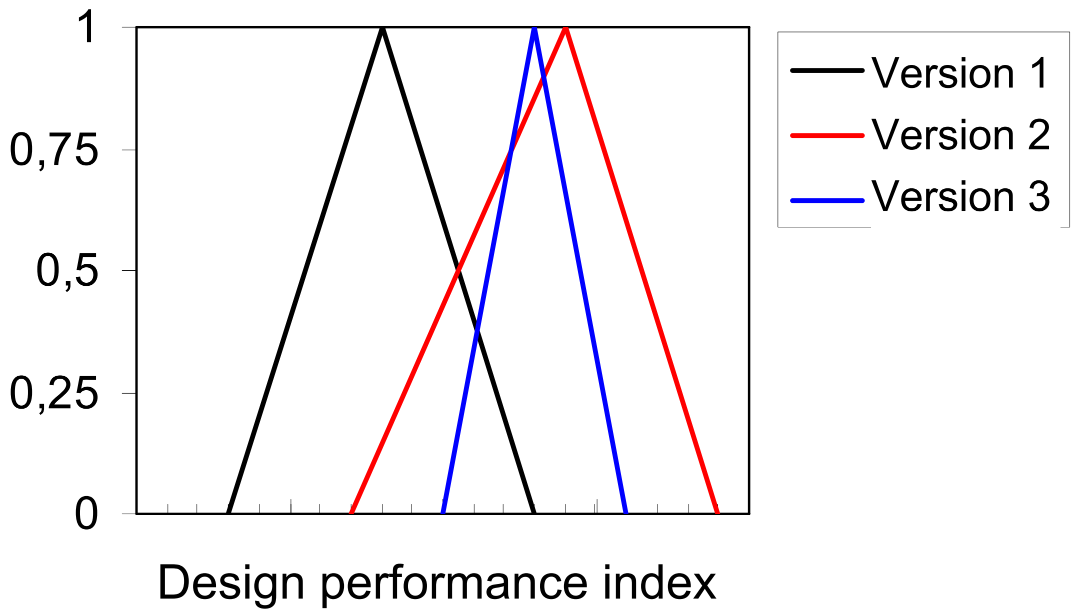
$$(l_x, m_x, p_x) \oplus (l_y, m_y, p_y) = (l_x + l_y, m_x + m_y, p_y + p_y)$$

- a product of such fuzzy numbers is (roughly):

$$(l_x, m_x, p_x) \otimes (l_y, m_y, p_y) \cong (l_x \cdot l_y, m_x \cdot m_y, p_y \cdot p_y)$$

Models utilising fuzzy numbers
Example of application

Exemplary results of calculations of fuzzy performance indices used for evaluation of deign versions



Version 1 jest apparently the worst one, but the decision weather Version 2 (more secure, but with a worse maximum) or Version 3 (with better maximum, but more uncertain) should be chosen, is not obvious.

Models utilising genetic optimisation

- Genetic algorithms are a modern and effective mathematical tool used for *optimisation arbitrary functions* (single or multivariable), imitating natural *evolution* processes.
- The optimised function, called *target* or *adaptation function*, is a model of a given problem (process).

  Example: minimising downtimes of a production line, being a function of parameters characterising schedule of performing manufacturing operations.
- The target function can be of an arbitrary form, e.g. an analytical formula or discrete. The only requirement is that the function value can be calculated for any values of independent variables.

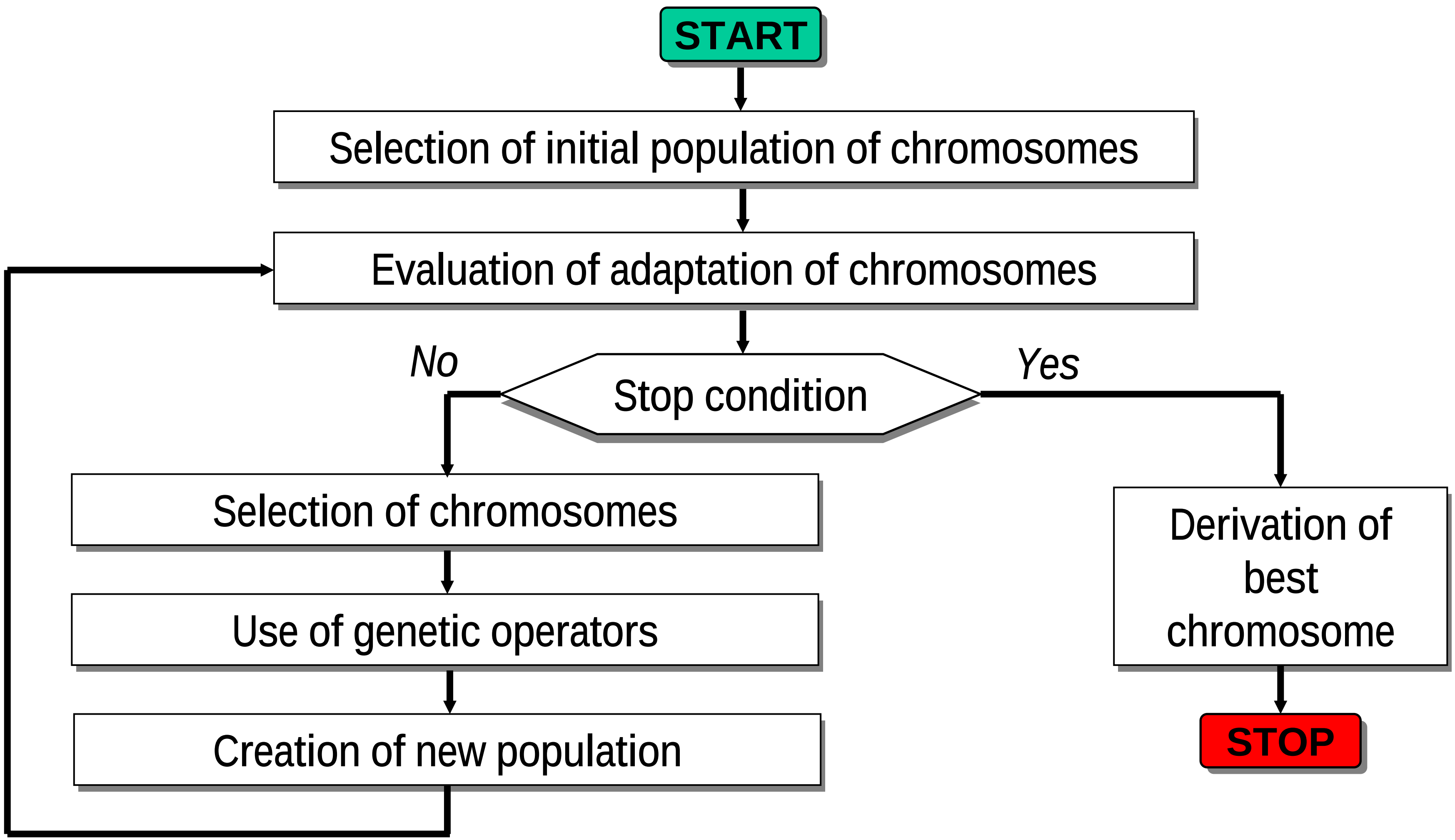
Models utilising genetic optimisation
Characteristics of genetic algorithms

- Genetic algorithms do not process the model parameters (which have to be found) directly but their coded values in a form of binary numbers sequences (*genes*), called *chromosomes*.
- The search originates not from a single point but from a *population* of points (set of chromosomes), e.g.:

  chromosome A **11011001**
  chromosome B **10010010**
  ....

- Genetic algorithms make use of the target function itself only, while many other popular optimisation methods (gradient type) use the derivatives.

Models utilising genetic optimisation
Functioning of genetic algorithms



Models utilising genetic optimisation
Functioning of genetic algorithms

- *Selection of initial population* of chromosomes (coded values of model parameters) is done by a random choice.
- *Evaluation* of the adaptation of chromosomes is based on the calculated target function value for each chromosome.
- *Decision about stopping of the computations* depends on fulfilment of the assumed condition, e.g. if the optimised function does change significantly its consecutive values.
- *Selection of chromosomes* for next generation is performed according to the natural selection rule, i.e. the largest chance for creation a new generation have those chromosomes which gained the best target function value.
- As a result of the selection a *new population* is created of the size equal to the previous population.

Models utilising genetic optimisation
Functioning of genetic algorithms

New population is created by application of so called genetic operators of two types:

- *Crossover* is done in the following steps: random mating of chromosomes from those selected for reproduction, random selection of a position in chromosome defining the crossing point, and finally creation of a new pair with mutually exchanged chromosome segments. For example for a pair of chromosomes of the i-th generation:

chromosome A_i **11011|001** → chromosome A_{i+1} **11011010**
chromosome B_i **10010|010** → chromosome B_{i+1} **10010001**
crossing point

- *Mutation* is a random change of a single gene (bit) to an opposite one, e.g. 0 to 1. Number of the changes made is small, usually several % of of all genes.

Models utilising genetic optimisation
Applications in manufacturing

- Production scheduling
- Optimisation of product design
- Optimisation of production processes and procedures
- Optimisation of operational parameters of devices

Genetic optimisation can be widely applied because of lack of limitations concerning the optimised function and a better chance of finding its global minimum. This is in opposite to gradient methods which often lead to local minima, which are obviously worse solutions .

Bayesian classification

This term covers learning systems based on probability calculus and utilising Bayes' theorem (formula). The systems include:

- Bayesian classifiers:
  - optimal, without practical significance
  - Naive Bayesian Classifier (NBC)
- Bayesian networks

Bayesian classification
Bayes' theorem

The Bayes' theorem is applicable to the events the occurrence of which was dependent on factors preceding those events. A probability that a given factor is a cause of the event (effect) can be calculated from the following formula:

$$P(h|D)=\frac{P(h)\cdot P(D|h)}{P(D)}$$

This formula allows to calculate the conditional probability of validity of a hypothesis h for data D, if the probability $P(D|h)$ is known (i.e. the probability that the data D are observed provided the hypothesis h is valid) as well as complete probability of the validity of the hypothesis $P(h)$.

Naive Bayesian Classifier (NBC)
Principles of application

A use of NBC requires calculations of probabilities on the basis of appropriate training set, which consists of examples described by discrete attributes (input variables) and the target (output variable).

Actions required for application of NBC:

- Defining categories for input and output variables. NBC requires usage of variables of the nominal or ordinal types. Continuos type variables have to be converted to categories as discussed previously.

Naive Bayesian Classifier (NBC)
Principles of application

Actions required for application of NBC (continued):

- *Creation of training set*, containing including attribute values of all inputs and the corresponding categories of the output.
- From the training set, the *probabilities* of individual output categories $P(h)$ *are calculated* as well as probabilities $P(D|h)$ of all individual categories of all inputs, for all output categories.

  This stage is defined as *training* of the classifier.

Naive Bayesian Classifier (NBC)
Principles of application

Actions required for application of NBC (continued):

- From the probabilities estimated during training stage one can calculate probabilities $P(h|D)$ of occurrences of all individual output categories for any given, new case (set of input values).
- The NBC *response* is that value of output category which achieved the largest probability value.

Naive Bayesian Classifier (NBC)
Principles of application

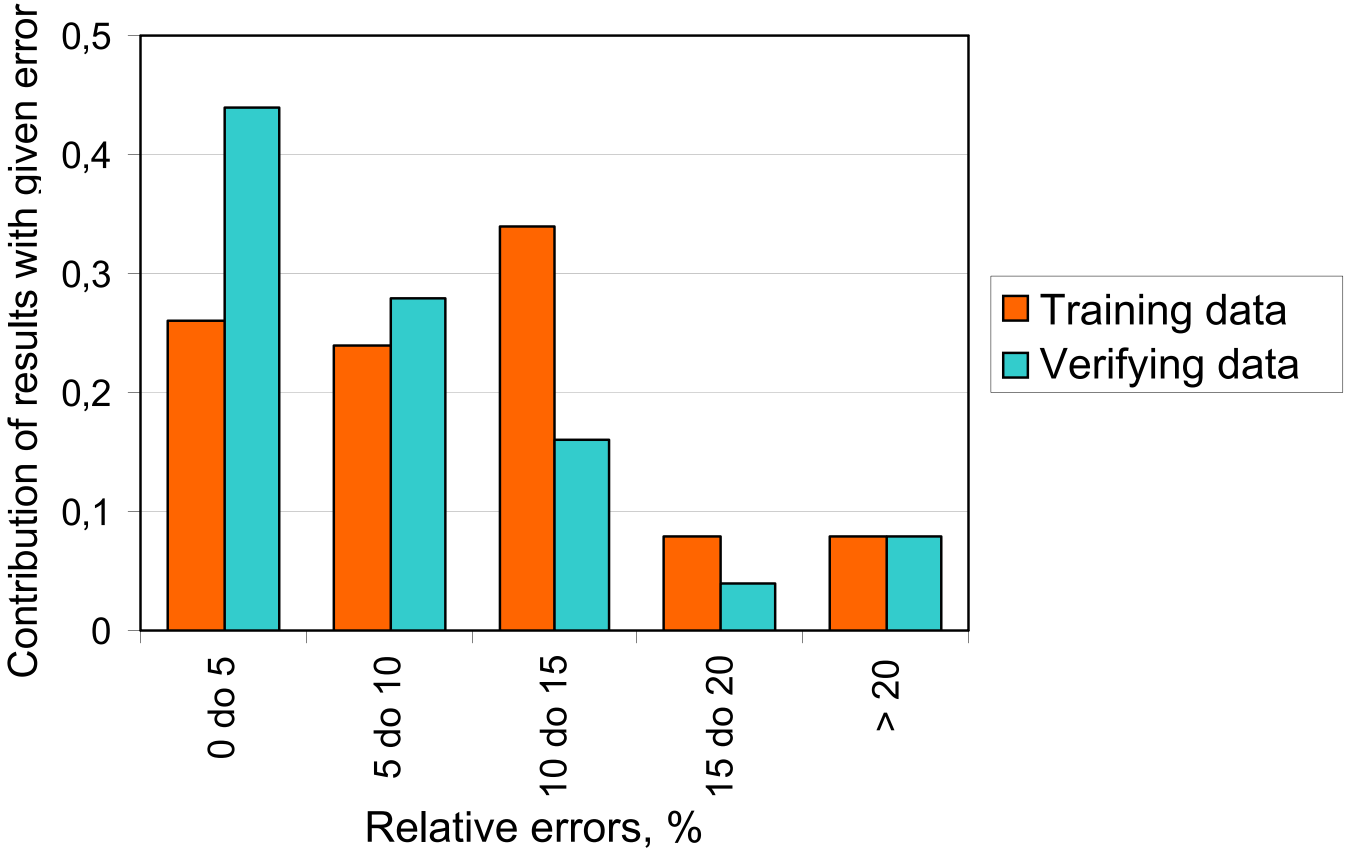
Calculation of probabilities in NBC:

- Use of NBC does not require determination of the probability $P(D)$ appearing in the denominator of the Bayes' equation, because its value is the same for all possible output categories.
- Conditional probabilities $P(h|D)$ and $P(h)$ are estimated from the frequencies of the appropriate occurrences in the training set.
- The formulas used for the calculations of those probabilities are so constructed that can handle the situations where some categories are not represented or are very few (automatic creation of 'virtual cases').

Naive Bayesian Classifier (NBC)
Example of application 1

Modelling of ductile cast iron strength as a function of its chemical composition

Training set 700 records, verifying set (independent) 90 records



Naive Bayesian Classifier (NBC)
Example of application 2

- Input variables were foundry process parameters related to sand mould (12 parameters) and the output was appearance of gas type defects in steel castings.
- In this example the original output was of discrete type (category ‘1’ – no defect, category ‘2’ – defective casting). Training set included 172 records (with no verifying set).
- Data was collected in one of Polish jobbing foundries using green sand moulding process.

Results of modelling:
NBC correctly predicted appearance or lack of defect in 87% of cases

MARSplines
Multivariate Adaptive Regression Splines

Models of that type have been developed by J. H. Friedman and applied quite recently (1991–2001).

They perform both regression and classification types of tasks.

They are an excellent tool used in data mining, in particular in solving difficult problems, i.e. where the dependencies between variables have a complex character.

MARSplines
General characteristics

MARSplines models utilise a nonparametric procedure that makes no assumption about the underlying functional relationship between the dependent and independent variables.

This relation is constructed form a set of coefficients and basis functions that are entirely ‘driven’ from the data.

General strategy of the model partitions the input space into regions, each with its own regression or classification equation.

MARSplines
Basic relationships of the model

The general MARSplines model equation is a relationship between the output (dependent) variable y and a set of the input (independent) variables X, in a form a linear combination of M basis functions:

$$y = f(X) = \beta_0 + \sum_{n=1}^{M} \beta_n \cdot h_n(X)$$

h_n – basis functions
β_n – weights of basis functions
β_0 – an intercept parameter

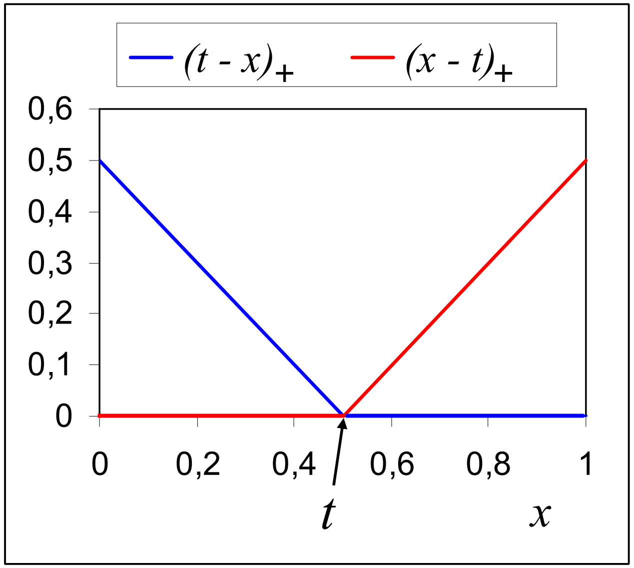
The model selects this weighted sum of basis functions from the set of a large number of basis functions that span all values of each output variable. Details of that selection are given later.

MARSplines
Basic relationships of the model

The basis functions are defined as two-sided truncated functions of the form:

$$(x-t)_+ = \begin{cases} x-t & x > t \\ 0 & x \le t \end{cases}$$

$$(t-x)_+ = \begin{cases} t-x & x < t \\ 0 & x \ge t \end{cases}$$



Parameter t is the knot of the basis function (defining the "pieces" of the piecewise linear regression). These are also determined from the data.

MARSplines
Procedure defining model

MARSplines algorithm searches over the space of all inputs and input values (knot locations t) as well as interactions between variables. During this search, an increasingly larger number of basis functions are added to the model to maximise an overall least squares goodness-of-fit criterion (minimise error).

Next the algorithm uses a pruning technique to limit the complexity of the model by reducing the number of its basis functions. Only those basis function are remaining in the model, which make a "sizeable" contribution to the prediction. Pruning also excludes insignificant input variables.

MARSplines
Most important characteristics

- High degree of flexibility combined with lack of overfitting to experimental data (as a result pruning technique).

  Note: too exact representation of the training data reduces generalisation capability of the model, i.e. their predictive capability for other (new) data.

- No particular knowledge about form of the relationship between input and output variables is required.
- MARSplines algorithm automatically determines the most important independent variables as well as the most significant interactions among them.
- It is possible to use single model for multiple dependent (output) variables. It utilises common set of basis functions in the independent variables, but estimates different coefficients for each dependent variable. This somewhat similar to artificial neural networks.

MARSplines
Application possibilities

The MARSplines models characteristics make their capabilities similar to artificial neural networks as well decision trees. They seem to be very competitive to both of them.

Applications of MARSplines to simulation of metallurgical processes are not known until now. However, some reports about applications in other fields of technology, including diagnosis of troubles in production processes.

METRO
MEtallurgical TRaining On-line

Computer methods for analysis and control of production processes

End of lecture

Education and Culture