Artificial intelligence in geotechnical engineering

Authors: Dipl.-Geoinf. Evelyn Bennewitz, Prof. Dr. habil. Heinz Konietzky,
(TU Bergakademie Freiberg, Geotechnical Institute)

Editor: Prof. Dr. habil. Heinz Konietzky Layout: Gunther Lüttschwager
TU Bergakademie Freiberg, Institut für Geotechnik, Gustav-Zeuner-Straße 1, 09599 Freiberg • sekr_fm@ifgt.tu-freiberg.de

1 Concept of Artificial Intelligence

Artificial **I**ntelligence (AI) is the ability of software applications and services to imitate cognitive thinking and intelligent behaviour based on **A**lgorithms for **D**ecision **M**aking (ADM). Software applications with AI are also called ADM-Systems.

For developing algorithms, which allow an artificial system to make decisions based on cognitive thinking and intelligent behaviour, the comprehension of the nature of intelligence, thinking and learning is required. In 1949, Donald Hebb postulated the learning rule (HEBB 1950). Considering thinking as related to neural activities required for processing information, the learning rule describes the impact of these activities on the connection between neurons and the synaptic plasticity on neural networks. For processing information, a neuron uses all input signals, which are arriving from different dendrites to form an output signal, which is send to connected neurons via axons. The intense use of an axon strengthens the connection between neurons, while not using may lead to a deletion of the connection and the axon. Strong connections facilitate the recovery of knowledge. In consequence, learning aims in building strong connections for relevant knowledge (see
Fig. 1).

Hebb's rule was a key finding in the development of the concept of artificial intelligence. When the term "Artificial Intelligence" was used first at Dartmouth Workshop in 1956, the research focused on finding formalism for representing the knowledge in implementable algorithms. Therefore, many scientific fields participate on the development of a concept of artificial intelligence:

- The analysis of thinking behaviour and autonomy from philosophers like Aristoteles, Hobbes and Pascal
- The methodology of formal logic from mathematicians like Bayes, Boole and Turing
- The game theory from economists like Smith and Neumann
- Models of brain and neural networks from neuroscientists like Broca and Berger

The control theory and cybernetics from scientist of robotics and machine control like Wiener and McCulloch

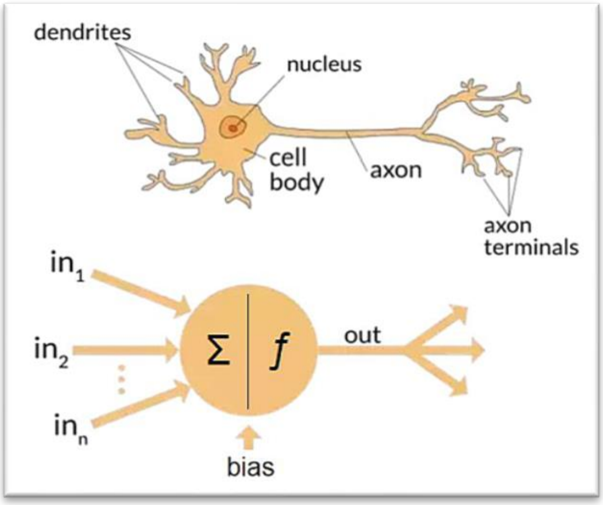


Fig. 1: Formalism of neuronal processing (company material of Dynardo GmbH: MOST et al. 2019)

In 1980, *Prolog* was the first formalism language, which allowed a programming of logical terms and knowledge. The name is consequently derived from "**Pro**gramming in **log**ic". With *Prolog*, it was possible to implement ADM-Systems.

ADM-Systems and thus artificial intelligence may be used in:

- Smart Things, for example for speech or pattern recognition
- Intelligent systems and robotics, for example for autonomous vehicles
- Fighting machines or environmental observations with drones
- Simulated worlds, for example for virtual realities and games
- Concept mining, data mining and text mining, for machine translation, document search and analyses of big data
- Analysis tools used for model calibration and optimisation
- Intelligent agents used in observation systems of complex technical networks and production plants

Considering rock mechanics, an ADM-System can be used for instance for selecting a suitable tunnel supporting system (see Fig. 2). The task or problem can be defined as a search resulting in the optimal tunnel supporting system as solution at the end of the decision making process. The decisions may be based on problem specific knowledge or criteria like geotechnical and geological properties of the area or underground water conditions (HAGHSHENAS et al. 2019).

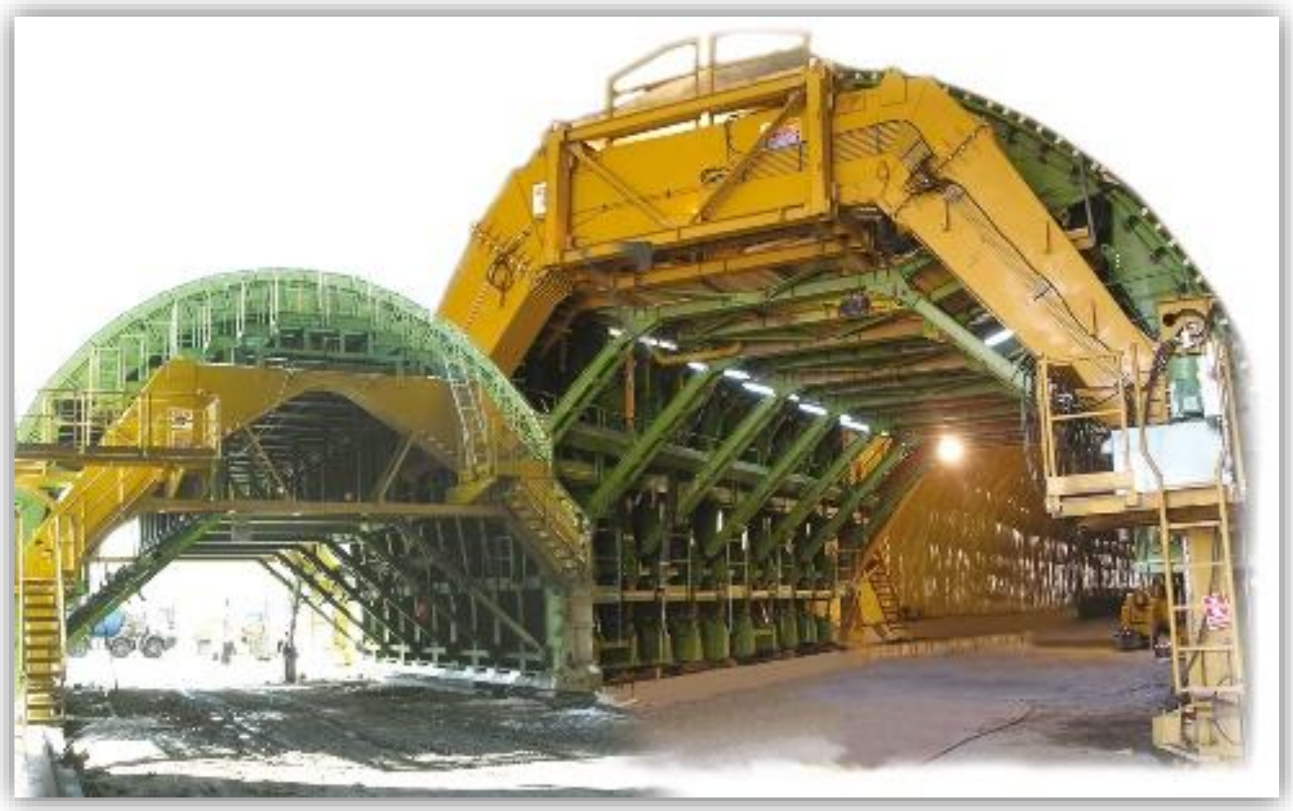


Fig. 2: Tunnel support systems (company material: CIFA 2020)

2 Types of ADM-Systems

ADM-Systems can be divided into knowledge-based and behaviour-based systems (JASPER 2020, see Fig. 3). The first type is represented in expert systems, while the second is related to agent systems.

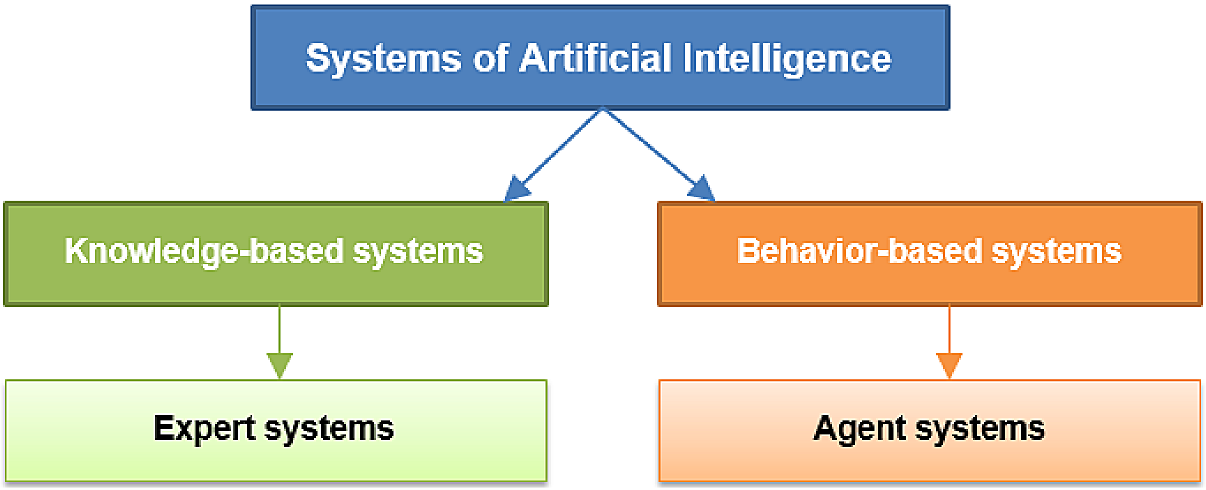


Fig. 3: Types of systems of AI

2.1 Expert Systems

Expert **S**ystems (XPS) are applications using ADM for a multiple criteria inventory classification by the usage of specific knowledge of experts and their ability to draw conclusions in form of problem solving strategies.

For selecting a suitable tunnel support system, HAGHSHENAS et al. (2019) developed a XPS based on mathematics and psychology (see Fig. 4).

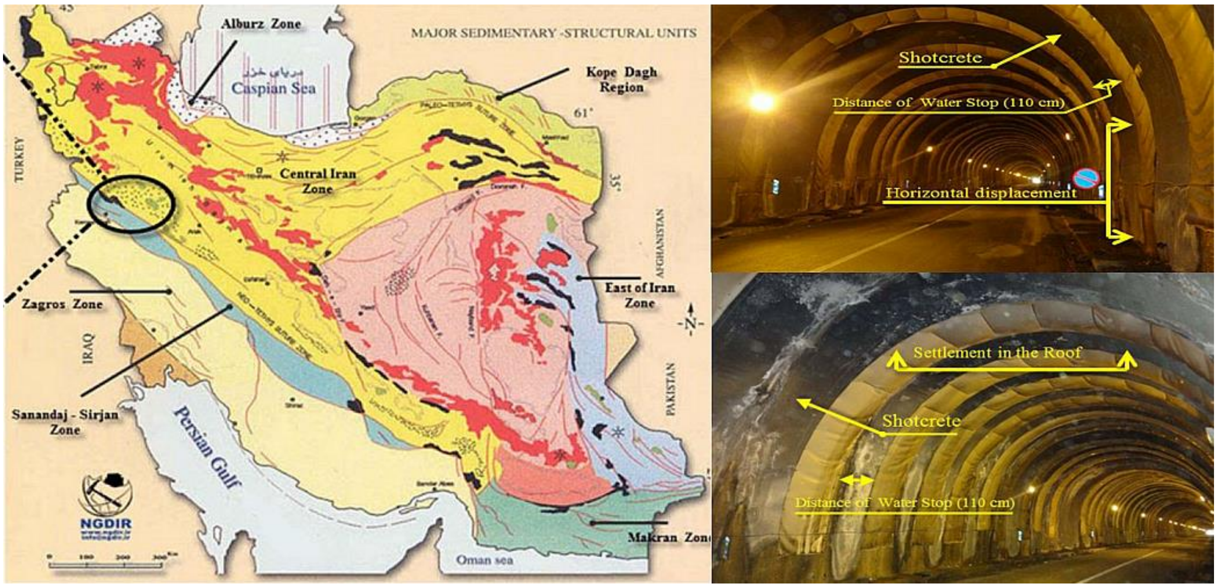


Fig. 4: Geological Map of the study region (left) and view into the Dolaei Tunnel with marked settlements and displacements (right) (HAGHSHENAS et al. 2019)

The development of this XPS as **I**ntegrated **D**ecision **S**upport **S**ystem (IDSS) started with a questionnaire to gain expert knowledge. For the questionnaire the **F**uzzy **D**elphi **A**nalytic **H**ierarchy **P**rocess (FDAHP) was applied. FDAHP is an extension of the **A**nalytical **H**ierarchy **P**rocess (AHP) for organising and analysing complex decisions, which uses a fuzzy instead of an exact value to express the decision maker's opinion in a comparison of alternatives. The Delphi technique was used in order to structure an effective group communication process. Different criteria were weighted in the decision matrix, which led to the identification of criteria most interesting for the selection.

Six significant criteria were determined for the IDSS:

1. Underground water condition
2. Geotechnical and geological properties of the area
3. Economical capacity
4. Access to implementation technology
5. Hardship of doing the job
6. Service life of the tunnel

After the process of data gathering, a multi-criteria decision analysis with ELECTRE was applied. ELECTRE is an acronym for **EL**imination **E**t **C**hoix **T**raduisant la **RE**alité, which can be translated as elimination and choice expression reality. The method of Bernard Roy is used for modelling the preference information between each pair of alternatives by outranking comparisons.

There are five alternatives for the tunnel support system:

a) Reinforced shotcrete
b) Metal frames
c) Concrete prefabricated segments
d) In situ reinforced concrete implementation
e) Rock bolt and reinforced shotcrete implementation.

The IDSS was evaluated in a case study for Dolaei tunnel of Touyserkan in Iran. The IDSS selected the rock bolt with reinforced shotcrete supporting system as the most suitable for the Dolaei tunnel. Experts agree with the decision to be the most appropriate system for stabilising the tunnel.

Summarising the principles of functioning of a XPS, the process can be described by six components (see Fig. 5).

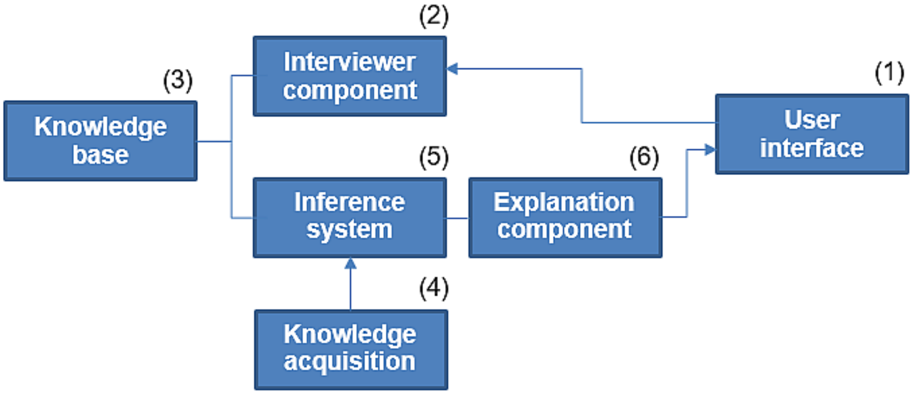


Fig. 5: Model of a XPS

The user interface (1) is used from the interviewer component (2) to gather information based on structuring ADMs like FDAHP. The information is used with a knowledge base (3) and maybe an additional knowledge acquisition system (4) by an inference system (5). The inference system derives or draws conclusions by an ADM like ELECTRE. The decision for a certain alternative in a special case defined by user input is represented with an explanation by the explanation component (6) via user interface to the user. The user can restart the XPS with other input information, for example for another case.

2.2 Agent Systems

Agent **S**ystems (AS) are applications using the input information from the environment and the user as well as own experiences for making decisions, completing orders, pursuing goals and running other applications independently.

For modelling a 3D geospatial environment, FRIDHI & FRIHIDA (2019) developed an AS in form of an **A**ugmented **R**eality (AR). Replacing mouse and touch screens by videocasque and gloves, the user is integrated into his environment and can interact with virtual objects, which are projected in front of his view. The overlay of computer graphics model on the daily environment was realised by a combination of AR, Google Sketchup Software (SketchUp) and ArcGIS.

For this ADM-System based on Sketchup, a special device called GeoScope was developed. With GeoScope, laser data can be received in real time, which is used for modelling the virtual reality from rough cloud processing in a defined mesh. However, the tools of Sketchup could not be used directly, because there are no direct commands of modelling. The manual adaption and combination with independently created tools based on Ruby scripting took months until an optimal result was obtained. The optimal result was received for data manipulation in building constructions based on virtual objects at the field side (see Fig. 6). The perception was enriched by the highlight of links for objects with additionally assigned information. FRIDHI & FRIHIDA (2019) assumed a great potential in using the concept in pedagogical systems regarding the acceleration of developing and evaluating hypotheses.

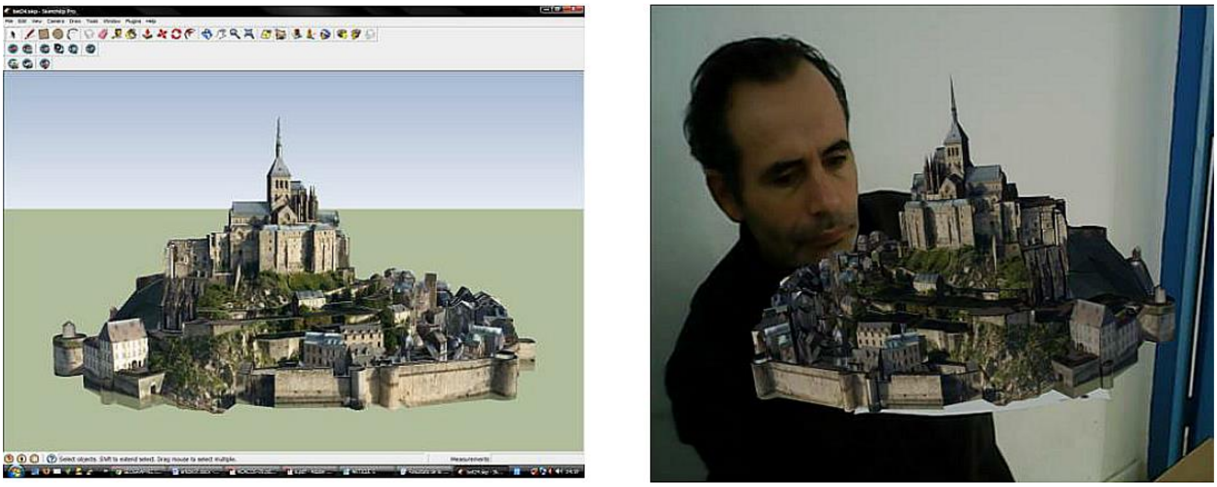


Fig. 6: Modelling a 3D Geospatial Environment within an Augmented Reality (FRIDHI & FRIHIDA 2019)

Considering rock mechanics, such an AR could be used in mine construction as well as in optimising mining and support machines. Another user scenario for an AS could be a mining warning system based on the observation of vibrations.

Summarising the principles of functioning of an AS, the process can be described as interaction between the environment and the AS. The AS consists of two components, the architecture of a special device with sensors, the knowledge base and effectors, and an ADM-System (see Fig. 7). The ADM-System (1) uses artificial intelligence (AI) to evaluate the input information, which can be received by sensors (2) like vibration values from the environment (3). The ADM uses required information of the knowledge base (4) to make a decision, in which way effectors (5) like running an alarm system (6), are used. Intelligent agents may also improve their ADM by own experience, which are gained either by user feedback (7) or evaluation functions based on the effectiveness of decisions and the way, how effectors affect environmental information.

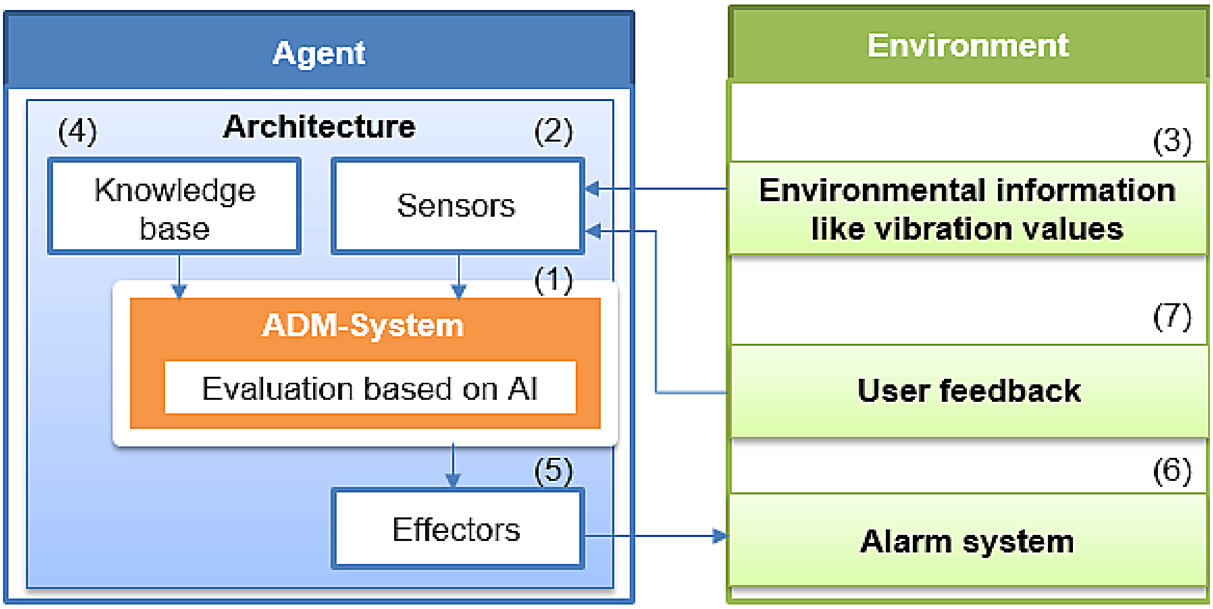


Fig. 7: Model of an AS

3 Principles of ADM within applications in geotechnical engineering

ADM-Systems are implemented by **A**lgorithms of **D**ecision **M**aking (ADM). The ADM can be used for three purposes:

1. Searching
2. Planning
3. Optimisation

The search algorithms are used to search for patterns and objects in a given search space, which is represented by the variability of each significant property or variable in a multidimensional space.

The search process can be visualised in a decision tree, where the required information given in a dataset is divided into subsets. The decision is based on given criteria, which results in a higher degree of disassembly up to a terminal node called leaf. The leaf contains a final solution following a path of decisions or conclusions (see Fig. 8). The solutions may be evaluated by problem-specific knowledge, and may be ranked depending on the costs regarding the number of decisions and the search time for finding the solution.

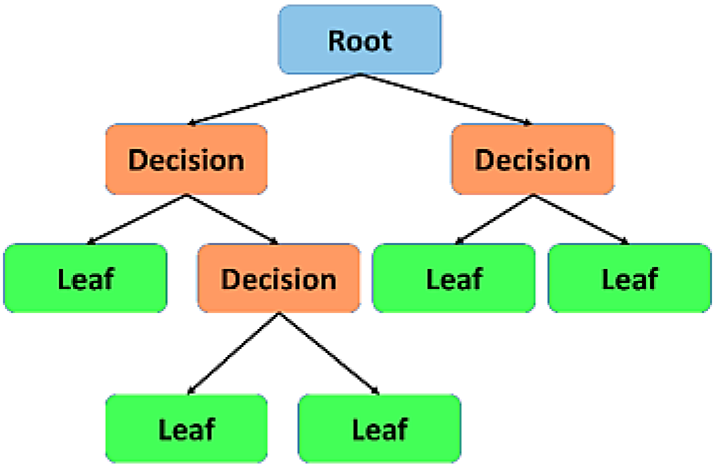


Fig. 8: Decision tree schematic showing root node, decision nodes and leaf nodes (Khan et al. 2019)

There are many different types of search algorithms (see Fig. 9), which can be classified regarding their approach of either structured searching with and without use of problem specific knowledge or searching with optimisation algorithms based on evolutionary or neurological processes. A deeper and more detailed differentiation can be realised with criteria like accuracy versus computational complexity and searching time.

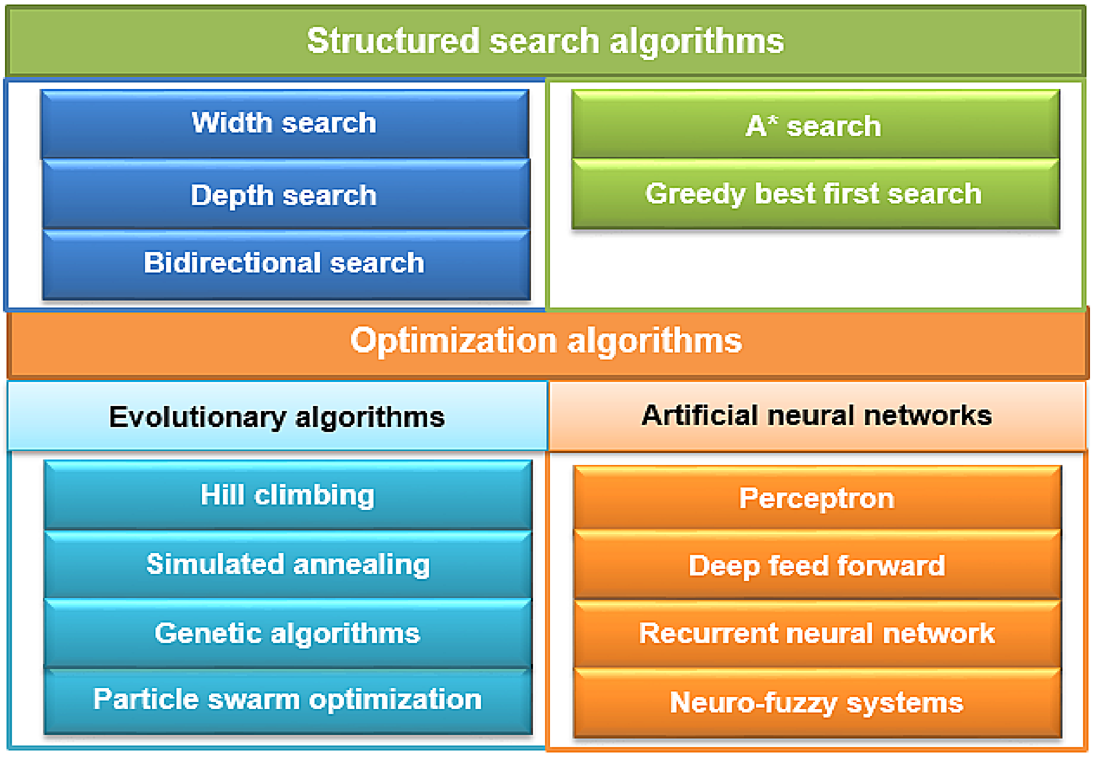


Fig. 9: Simple classification of ADM with selected, representative examples of algorithms

3.1 Structured search algorithms

Structured search algorithms are global searches exploring the whole knowledge base. When a user starts the process by a query, each information term of the knowledge base is checked regarding conditional rules for conclusions. Found information is used for substituting searched patterns in form of variables in the query and may add new search variables, which have to be substituted as well. If the solution in answering the query cannot be found, the algorithm returns to the penultimate decision for substituting information and search for alternatives. This process of backtracking requires the knowledge about the order of decisions, which have been done so far. The structuring of the search in the knowledge base enables a systematic search through all the data. The accuracy comes at the cost of computational complexity and searching time.

The computational complexity and searching time can be reduced either by using problem-specific information or information about the costs of making each decision. Using problem-specific information, the search algorithm is called informed search algorithm. Using the costs of the decision is done by limiting the steps, which have to be done to reach a solution. Decisions for alternatives, which exceed the limit, are ignored in the search process. In consequence, only a part of the knowledge base is explored to find the solution. If a solution is not found, the limit may be increased and the search is restarted.

Considering rock mechanics, a simple query could be the question: On which depth level a special mining machine like a “Development Jumbo Drill” can be used for mining? For answering the question of this example, a knowledge base can be used, in which mining machines are defined corresponding to their depth level, in which they can be used for mining operations. The search process of the ADM can be visualised with a node-based tree representation of knowledge (see Fig. 10).

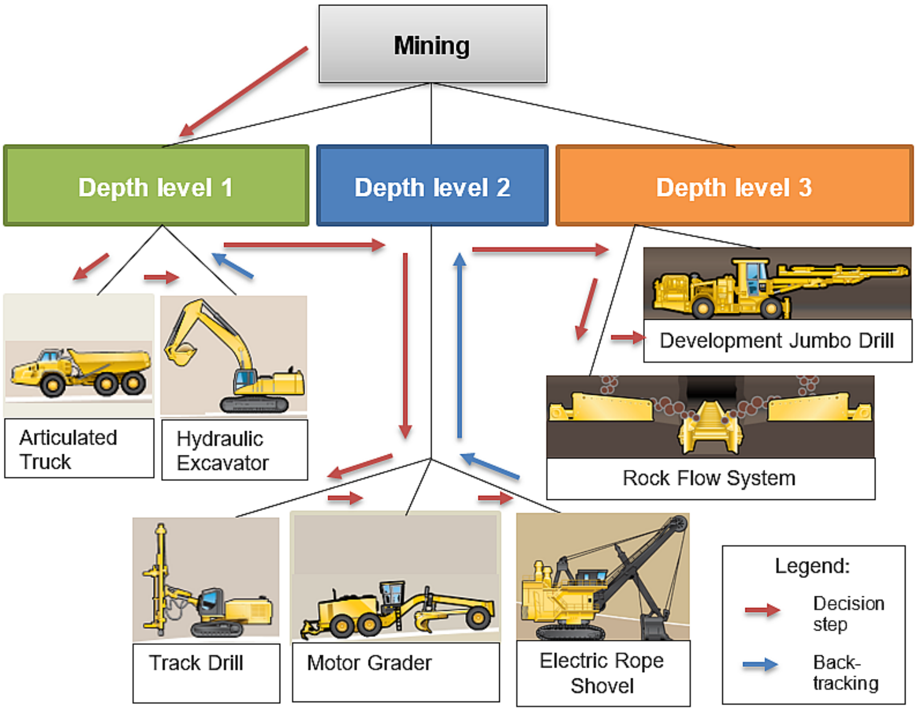


Fig. 10: Decision tree for a simple query to select the appropriate depth level for a Developed Jumbo Drill (CATERPILLAR 2020)

The search process may start with a pattern recognition technique for mining machines related to a lower depth (depth level 1), and end for one related to a higher depth (depth level 3). The depth level is the first decision point of the ADM. If all machines of one depth level failed in checking the searched pattern „Development Jumbo Drill“, the decision of the ADM was wrong, and backtracking is used to come back to the decision point, where the next alternative is selected. So, level for level, all corresponding machines were checked for identity with the searched pattern „Development Jumbo Drill“. The search algorithm stops, when it succeeded in finding the searched pattern, or when it failed.

In case of success, the knowledge base was only explored until the first match. There could be more than one match. More complex search algorithms continue searching until all matches have been found by a complete search through all the knowledge base. In case of failure, the search pattern could not be found, either by misspelling or incompleteness of the knowledge base. However, the search was realised completely and systematically through all the knowldege base.

Computational complexity and searching time increase exponentially with the size of the knowledge base. Regarding the increasing computational complexity and searching time, all structured algorithms become inefficient to solve greater problem tasks.

Optimisation algorithms use optimisation techniques for realising local searches without exploring the whole knowledge base. Instead of using all information, only relevant information should be used in the search algorithm. The number of input parameters might be reduced by known correlation between them. Only parameters with a significant effect on the output should be used. The selection of significant input parameters can be performed by means of a sensitivity analysis (KONIETZKY & SCHLEGEL 2013, see Fig. 11).

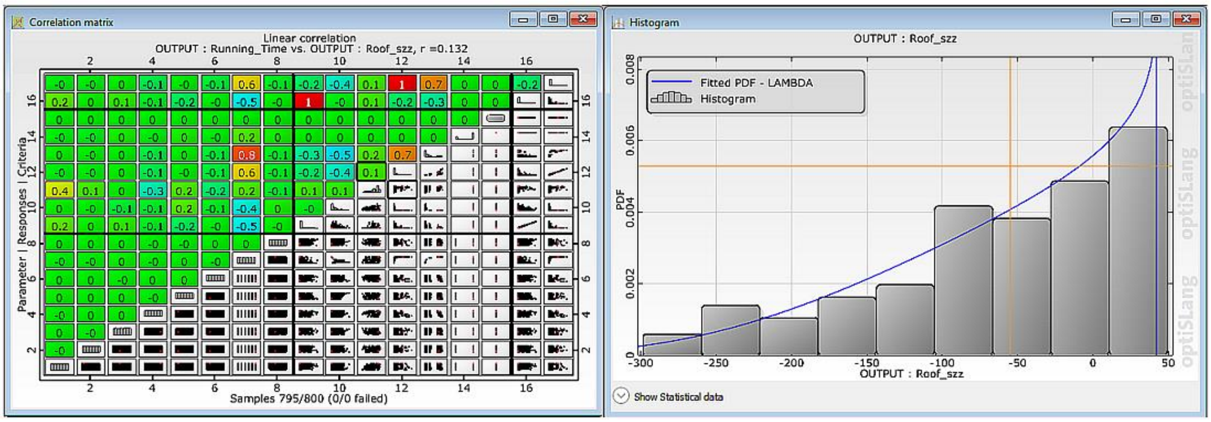


Fig. 11: Sensitivity analysis for mining parameters like normal stress on the roof

With the stochastic distribution of the variability of an input variable, sample points can be derived when required. The process gaining representative samples is called sampling.

The simplest way of sampling is the random sampling (LANCE & HATTORI 2016), where sample points were taken randomly without systematic division of the search space or considering other sample points. The sample points are just a set of random numbers, which may not guarantee to be representative for the whole search space or variability of a variable.

A sampling method, which creates subsets of the search space by division into equal intervals, is Latin Hypercube Sampling (LHS). The sample points are placed in rows and columns without threatening each other. LHS ensures that the sample points are representative for the search space or variability of the variable (see Fig. 12).

For instance, random sampling and LHS can be applied to identify the significance of normal stiffness of joints on normal stress σ for a specific rock mechanical model. The rock mechanical model is defined by different rock mechanical properties within a certain tolerance. Considering the optimisation task of maximising the stability or minimising stress in special locations, the impact of the variability of variables like the normal stiffness of joints could be investigated. The variability is given by a real number, for which the possible number in a tolerance interval is infinite and cannot be calculated for all cases. Instead, only representative numbers in the tolerance interval are taken as sample, for which the impact on the output like normal stiffness of joints on normal stress is calculated.

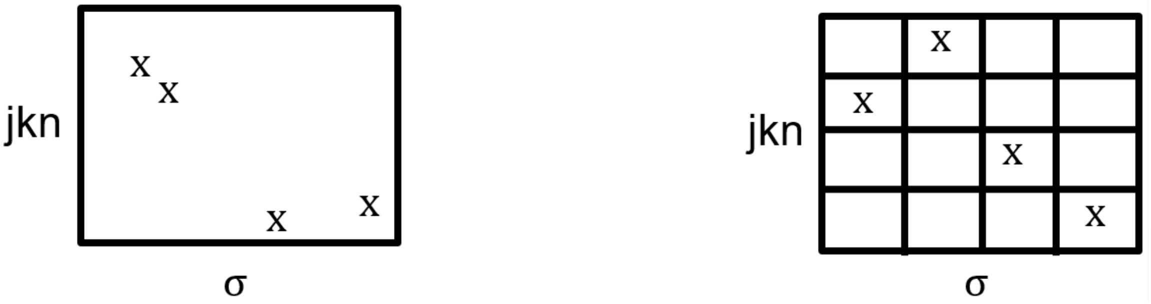


Fig. 12: Random sampling (left) and Latin Hypercube Sampling (right) in 2D

Sampling only results in points in the multidimensional search space defines only the impact for special single values. For mapping the behaviour of the full variability of the variable, the values between the sampling points are required too, which can be found by regression methods. The regression can be based on a clustering algorithm like **k**-**N**earest **N**eighbours (k-NN) classification to determine a numerical output (see Fig. 13). For the determination, the values of a given number of nearest neighbours, for example the three nearest neighbours, are considered.

Another simple regression and classification technique is the **S**upport **V**ector **M**achine (SVM, see Fig. 14), which enables solving a linearly constrained quadratic programming function, which results in a unique, optimal and global solution. The selection of linear classifiers is based on one or more optimal hyperplanes, whose margin between the two closest data points are the smallest. The determination of the upper bound of the margin of hyperplanes enables the minimisation of generalisation errors like risk minimisation. For finding the largest deviation from the actual target vector for training data, the insensitive loss function can be integrated. For the integration of nonlinear functions like the insensitive loss function, kernel functions for the nonlinear support vector regression like a **R**adial **B**asis **F**unctions (RBF) are required. The Kernel function mainly controls the complexity of the prediction.

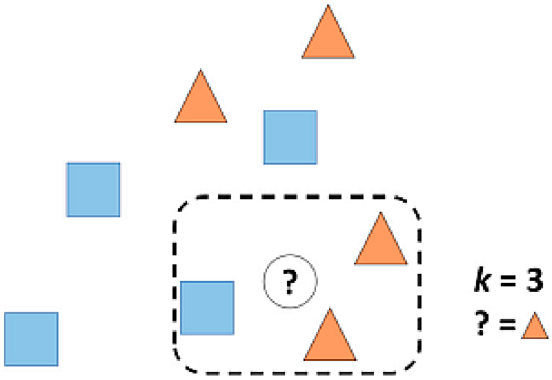


Fig. 13: k-NN classification scheme for a classification based on the three nearest neighbours (MORGENROTH et al. 2019)

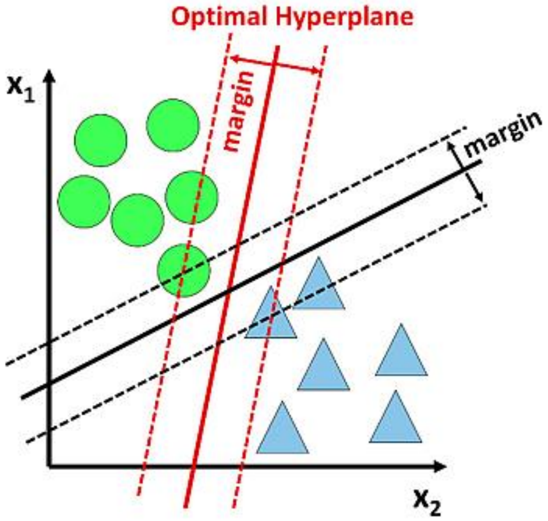


Fig. 14: Two separation straights for creating subsets obtained via SVM

Results of simple regression and classification techniques may highlight regions of interest, where further investigations may be recommended. Further investigations can be realised with advanced and extended sampling methods like Advanced LHS and Optimised LHS, which use the existing sample points for placing new ones.

Advanced sampling methods are for instance importance sampling, directional sampling, adaptive response surface method and first order reliability method (see Fig. 15). The impact of several input parameters like ***x1*** and ***x2*** on an output parameter are visualised in a multidimensional space. For example, these input parameters may correspond to the normal or shear stiffness, while the output may be the normal stress. For the approximation of the impact, the **p**robability **d**ensity **f**unction (pdf) can be used. Optima of the pdf may signalise a higher impact on the output. Therefore the investigation of these regions is more important than of other regions. For deeper investigations, a resampling can be realised focusing on regions with optima like **$f_X(x)$** and **$h_y(x)$**. The different regions of higher interest can be classified or clustered using advanced classification techniques.

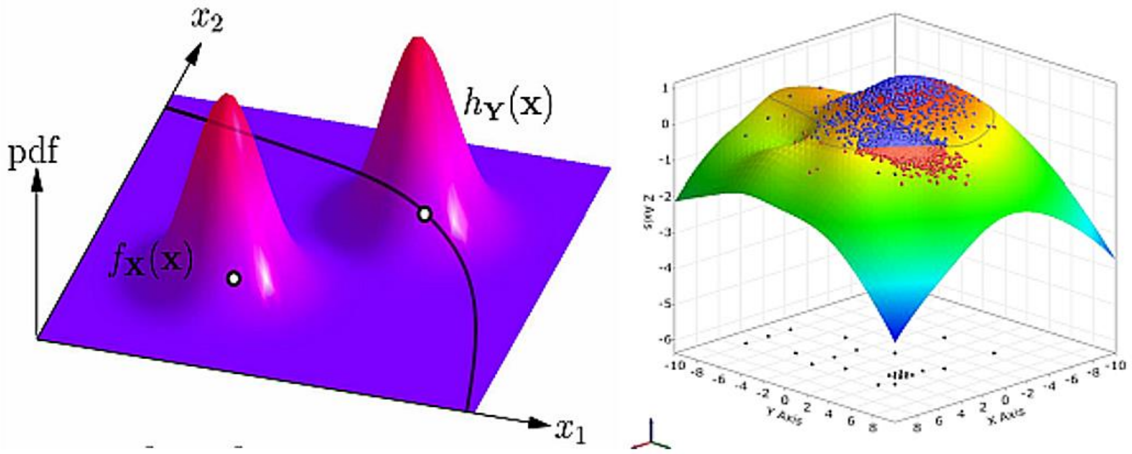


Fig. 15: Importance sampling (left) and Adaptive Response Surface Method (right) (BAYER 2019)

Different regression methods were evaluated by SHISHEGARAN et al. (2019) on predicting the earthquake magnitude along the Zagros fault (see Fig. 16 and Fig. 17). The prediction of the magnitude of the earthquake from previous earthquakes with magnitudes of more than 2.5 between 2009 and 2018 is based on two time series by applying the autoregressive conditional heteroscedasticity (GARCH), the **A**uto**r**egressive **I**ntegrated **M**oving **A**verage (ARIMA) and a combination of both by **M**ultiple **L**inear **R**egression (MLR) technique. For modelling, the time series data was split into training and test data.

The GARCH method is a statistical method using the variance of the error term from the squared previous error terms and the current error term with a mean offset. The ARIMA method is based on calculating the correlation coefficient between the current and up to four previous earthquake data to calculate the error parameter. The combination of ARIMA and GARCH with MLR is a common method, whose outputs are independent variables with the purpose of keeping each model property to enlarge the dataset.

For the evaluation of the three different models, their accuracies were calculated by statistical parameters like correlation coefficient (*r*), root mean square error (*RSE*), normalizes square error (*NSE*), and fractional bias. The best fit was obtained by an model combining both methods, whereas results from ARIMA model are still better than those from GARCH model.

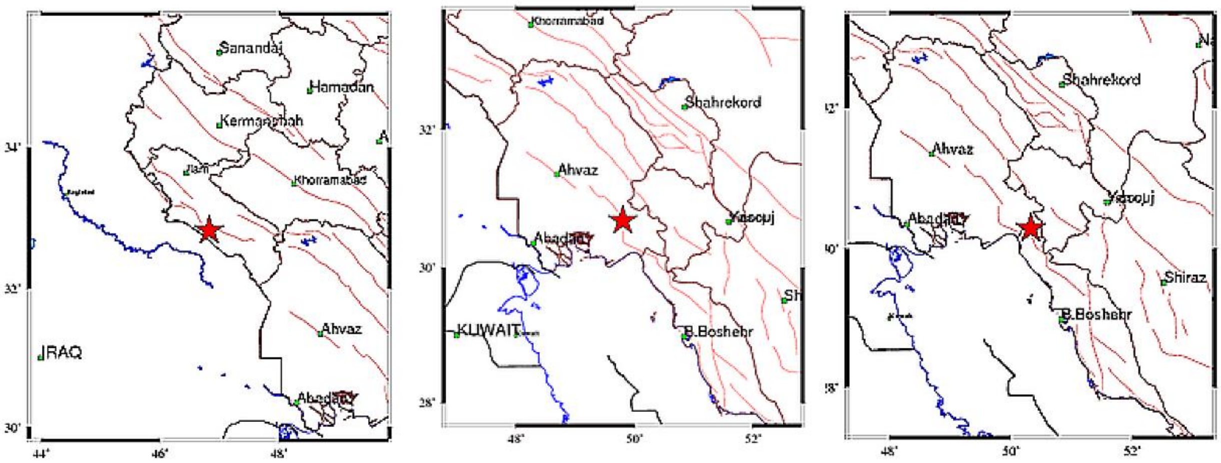


Fig. 16: Study regions of earthquake events (Shishegaran 2019)

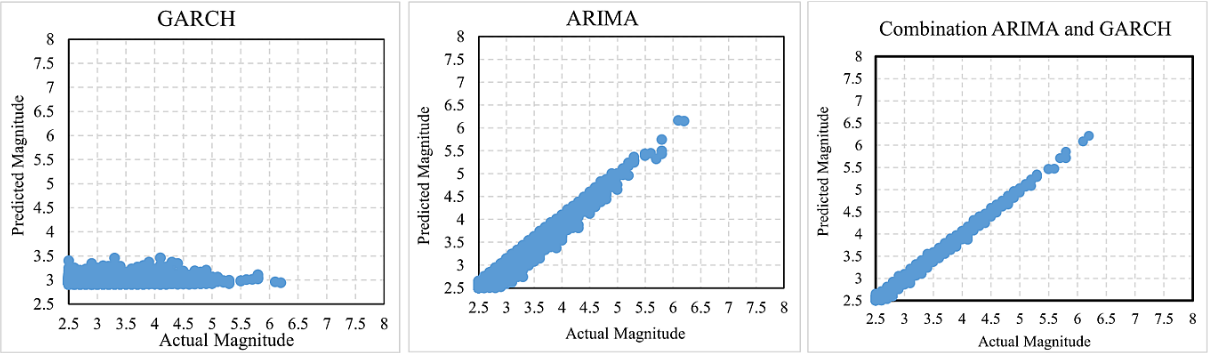


Fig. 17: Evaluation of different regression models (Shishegaran 2019)

Linear regression functions can also be used in conventional decision trees for predicting continuous classes or classes taking continuous numeric values. Therefore, Quinlan's algorithm for developing model trees M5, was combined with the possibility of linear regression functions at the nodes of the conventional decision tree. The improved algorithm is called **M5P**.

The development of the conventional decision tree (see Fig. 18) is realised in two steps:

1. Examining all possible splits with a splitting criterion based on the standard deviation of the class values
2. Application of a pruning method to replace sub trees with a linear regression function to avoid overfitting

The nodes of the tree are calculated by an expected reduction of the error computed from the standard deviation of the class values. Parent nodes have a higher standard deviation than children nodes, which are then considered to be more pure. A large tree is created, which has to be pruned back in order to avoid overfitting. M5P enables the prediction of continuous numerical attributes at the terminal nodes (leaf).

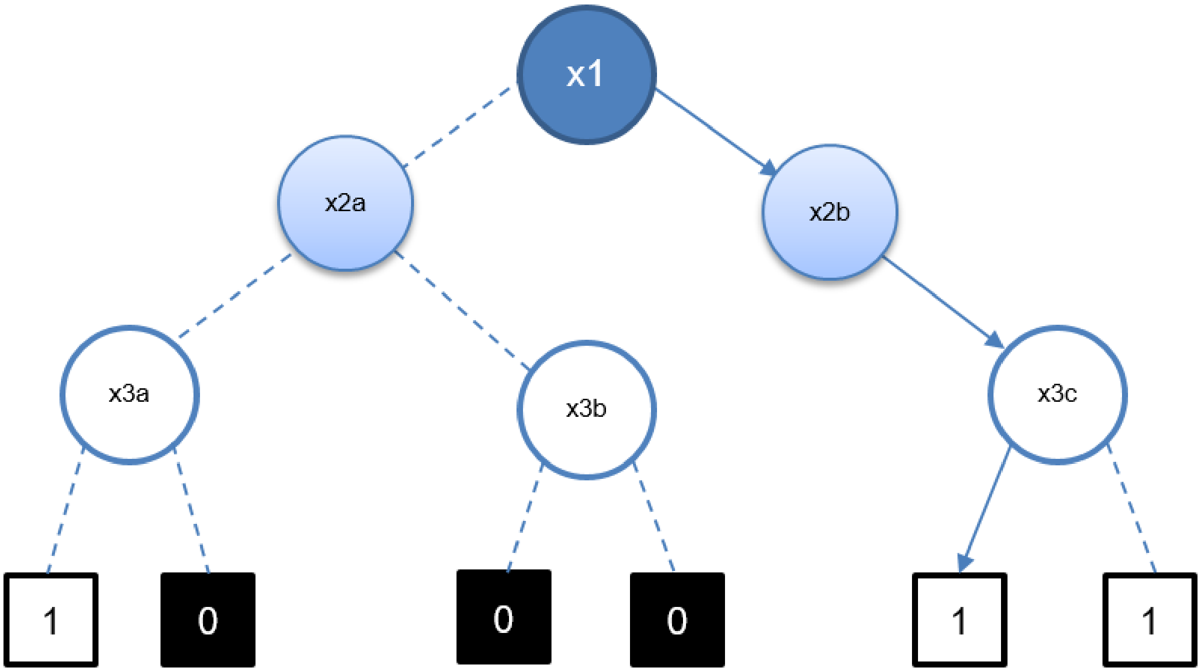


Fig. 18: Binary decision tree

For creating a forest of many individual trees, the **R**andom **F**orest **R**egression (RFR) can be used (see Fig. 19). This regression and classification technique is based on a combination of tree predictors. The tree predictors randomly draw and replace a combination of parameters or selected parameters using the bagging technique. This tree generation technique chooses random values for the vector of the RF classifier, which were independently picked up from the input vector. The RF classifier uses numerical values as opposed to classification labels. There are two additional parameters required for the RFR, the number of trees to be developed, and the number of variables required to create a tree at each node. A prune method may be applied to design a tree predictor. To measure the quality, the impurity of the variable with respect to the output (Gini index), can be assigned.

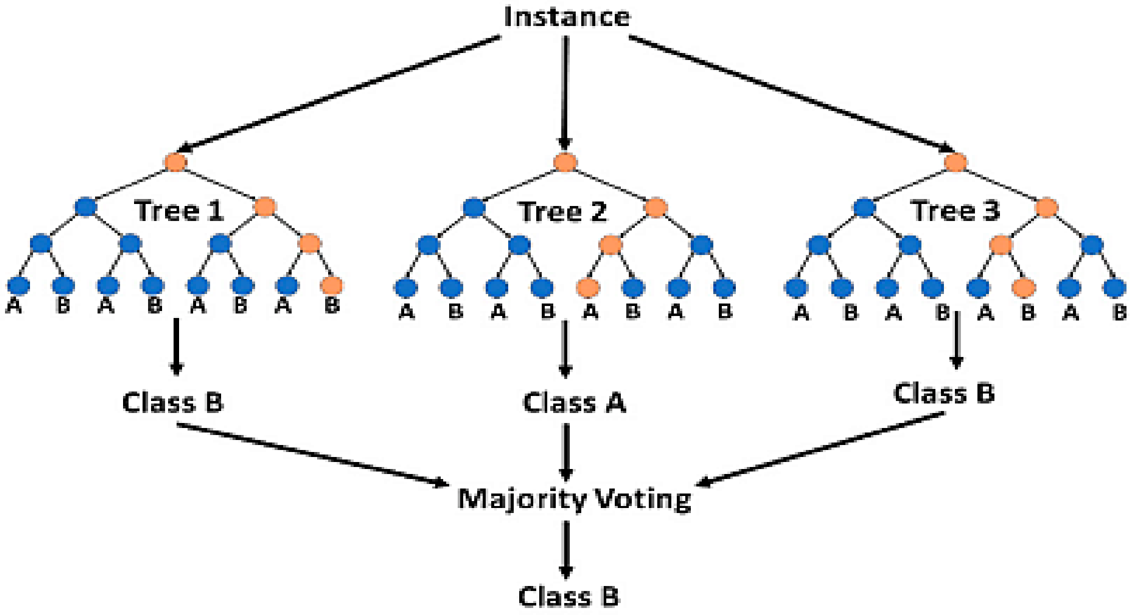


Fig. 19: Random Forest schematic for a tree decision (tree classifiction), in which the final class is determined by majority voting (MORGENROTH et al. 2019)

For predicting the **U**ltimate **B**earing **C**apacity (UBC) of strip footing subjected to eccentric inclined load and resting on sand, DUTTA et al. (2019) compared the performance of three regression and classification techniques:

1. Support vector machine with a radial basis function (SVM RBF kernel)
2. M5P model tree (M5P)
3. Random forest regression (RFR)

The UBC was computed with a reduction factor. A sensitivity analysis was performed to study the major input parameters, which affect the reduction factor. Four significant parameters were identified, whereas inclination ratio and eccentric ratio are the most important parameters:

1. Ultimate bearing capacity of the footing subjected to vertical load
2. Eccentric ratio
3. Inclination ratio
4. Embedment ratio

For five different combinations of all input parameters, the reduction factor was computed with SVM RBF kernel, M5P model tree and RFR. For bagging of RFR, 67 % of the original data was used for the training and 33 % was left out from every tree grown (see Fig. 20). Realising overfitting negligible, fully grown trees were not allowed to prune back with any prune method. In consequence, RFR could be better maximising the expected error reduction than M5P.

The evaluation was realised by the comparison of the performance measures:

- Correlation coefficient (*r*)
- Coefficient of determination (R^2)
- Mean square error (*MSE*)
- Root mean square error (*RMSE*)
- Mean absolute error (*MAE*)
- Mean absolute percentage error (*MAPE*)

Ranking the regression and classification techniques used for predicting UBC by an ascending error, results in SVM RBF kernel, M5P and RFR.

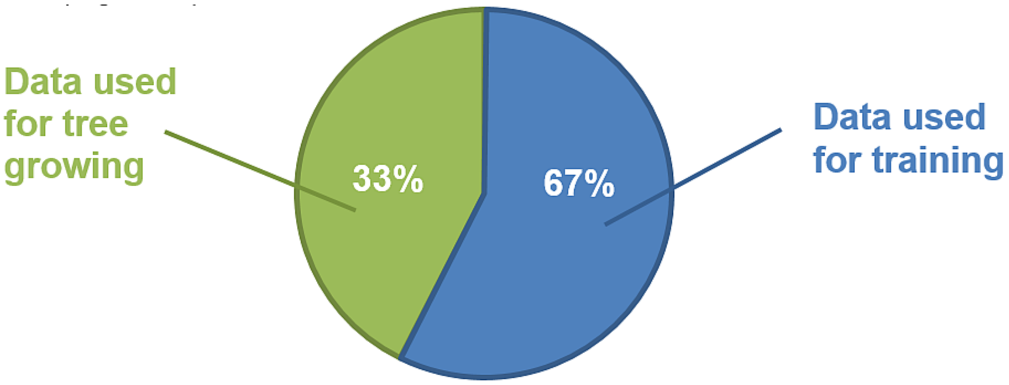


Fig. 20: Ratio of input data used for training and tree growing in the bagging technique (DUTTA, RAO & KHATRI 2019)

3.2 Optimisation algorithms

Optimisation algorithms can be based on imitating natural, biological processes like evolution, and can be learning based using neuronal activities. Evolutionary-based search algorithms are for example hill climbing, simulated annealing, genetic algorithms and particle swarm optimisation.

Hill climbing starts with a sub-optimal solution to a problem and repeatedly improves the solution, until the condition like maximising a target function, is fulfilled. The heuristic optimisation process evaluates the change in each step and only takes the changes, if there is an improvement. If changes can not improve the result anymore, the algorithm stops with the last accepted design as optimal solution. Therefore, Hill climbing may also stop in local optima instead of finding the global solution.

Simulated **A**nnealing (SA) is a metaheuristic approach to approximate the global optimum for solving unconstrained and bound-constrained optimisation problems. A Markov-Chain-Monte-Carlo technique, the Metropolis algorithm with the Boltzmann distribution, is used for transferring the start to the final state. The start state is given by the design of input parameters, while the final state is the one, where the optimisation problem is solved. The main advantage is to overcome local optima by accepting less unfavourable interim solutions.

Genetic **A**lgorithms (GA) are based on a population of individuals, which are defined in a vector of significant object variables (see Fig. 21).

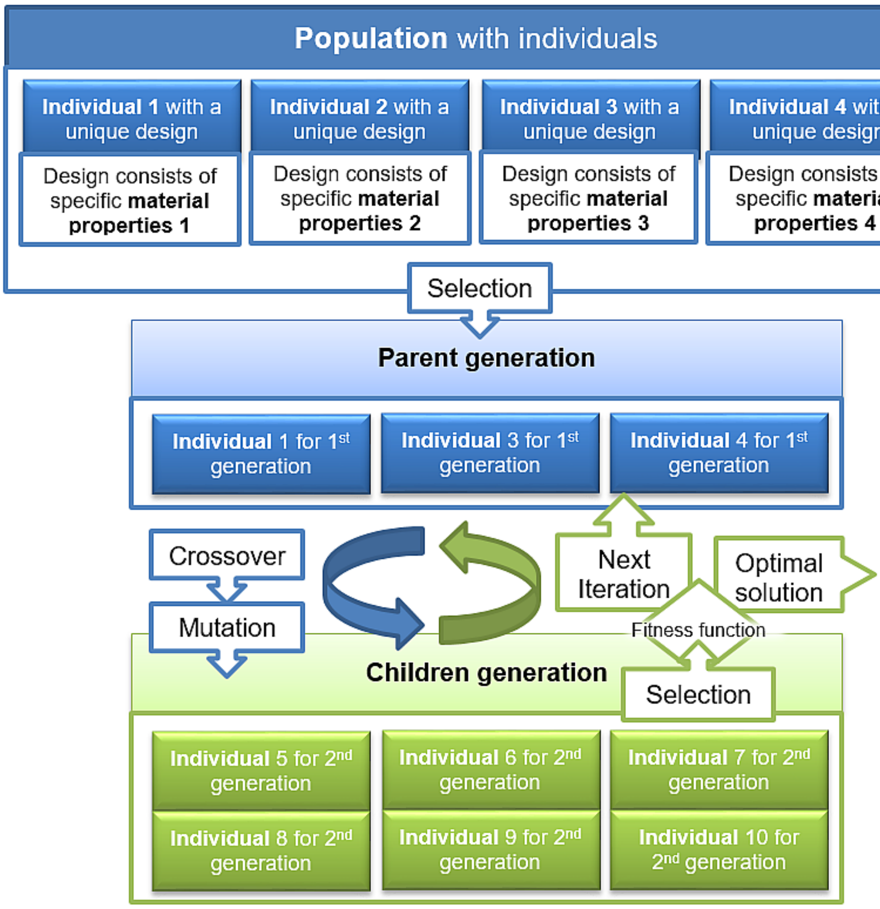

Population with individuals
Individual 1 with a unique design
Individual 2 with a unique design
Individual 3 with a unique design
Individual 4 with a unique design
Design consists of specific material properties 1
Design consists of specific material properties 2
Design consists of specific material properties 3
Design consists of specific material properties 4
Selection
Parent generation
Individual 1 for 1st generation
Individual 3 for 1st generation
Individual 4 for 1st generation
Crossover
Mutation
Next Iteration
Optimal solution
Fitness function
Children generation
Selection
Individual 5 for 2nd generation
Individual 6 for 2nd generation
Individual 7 for 2nd generation
Individual 8 for 2nd generation
Individual 9 for 2nd generation
Individual 10 for 2nd generation

Fig. 21: Flow chart of a GA

The design of the vector can be represented in a chromosome with different genes. The definition of the values of all genes in a bit string is called genotype, which is consequently the representative coding of individuals based on their design. From the original population a subset of a given population size is selected by a specific selection operator. For the selection operator, many methods such as Roulette wheel method, Tournoment method, and random method can be used. The subset of the population used for starting the evolution process, is called parents, whereas the size can be greater than two. The change of individual design is realised with operations like crossover and mutation. These operators are working without problem specific knowledge. The recombination of individual object variables with crossover affects only some genes in the chromosome, and the difference between parent's genotype and children's genotype is small. For more dispersion, mutation is used. Mutation can be realised in a randomised bit inversion of the genotype. This creates a new generation of the population with different design, which are called children. The proportion of the population of the children due to crossover is either called crossover ratio or percentage of crossover. The ratio of the mutated population to the population is either called mutation ratio or percentage of mutation. From the children population, a subset is selected to be the next parent generation. The selection is done by a fitness function, by which the design of the individuals is evaluated to find the ones with the best changes for coming closer to the optimal solution. This optimal solution is found, if the results of the fitness function cannot show any further improvements.

GA were used by FARAHANI (2020) for finding the optimum repair and maintenance scenario for a corroded **R**einforced **C**oncrete (RC) structure in the marine environment of Bandar-Abbas coasts located in south side of Iran. The objective function was based on minimising **L**ife **C**ycle **C**osts (LCC) and maximising service lifetime.

The research investigates the time-dependent capacity of a corroded circular RC column by using a nonlinear **F**inite **E**lement (FE) analysis for 40 years failure time in terms of corrosion. The investigation included:

- Five different concrete surface coatings used on the external surface of the concrete
- Four different increasing concrete cover thicknesses
- New longitudinal and horizontal reinforcements after initial cracking of concrete cover

The optimised scenario offered by GA in a FE model determines a mixture design for the concrete given by a water-to-binder ratio and replacement ingredients like silica fume for Portland cement, as well as design parameters for reinforcements like the diameter, for minimised life cycle cost with an initial concrete cover and increasing concrete cover thickness. FARAHANI (2020) evaluated the result as an acceptable solution for the case study.

Particle **S**warm **O**ptimisation (PSO) imitates the search of a bird swarm for finding the best localization of an optimal resting place (Esmin et al. 2015, see Fig. 22), whereas the properties of possible places are additionally considered besides individual properties like the rank of the bird in the swarm. Each bird is handled as particle with a randomised start position with a velocity vector.

The evaluation of places in the search space is based on

- Inertia of movement
- Individual best value of each particle
- Global best value
- Special weighting factor like cognitive or social weights (social rank)

PSO demonstrates its proper functioning in many areas, such as finding optimal solutions for functions, training neural networks or controlling fuzzy systems.

A comparison of different evolution-based search algorithms, in particular hill climbing, genetic algorithms (GA), particle swarm optimisation (PSO), differential evolution and artificial bee colony, has proven, that GA, PSO and the differential evolution method have the best performance, which is given by the accuracy presented in the solution quality, and the processing time for obtaining the solution (Chassiakos & Rempis 2019).

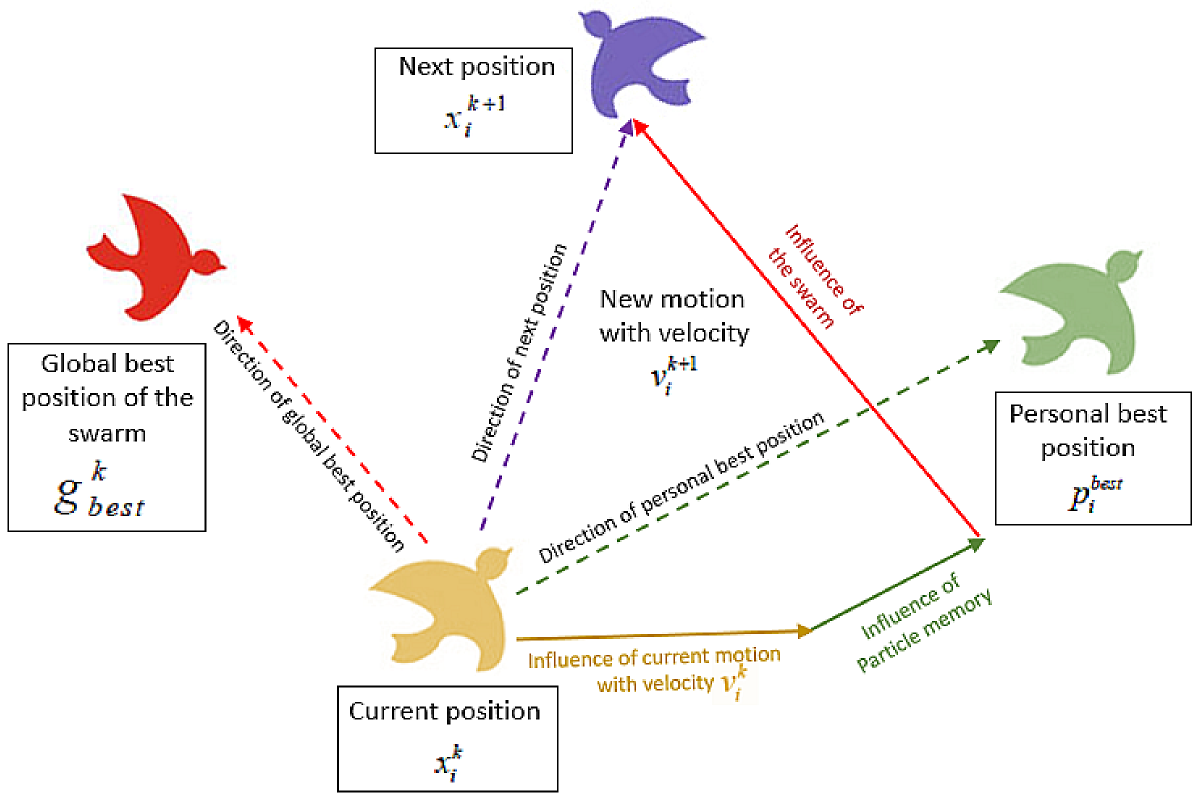
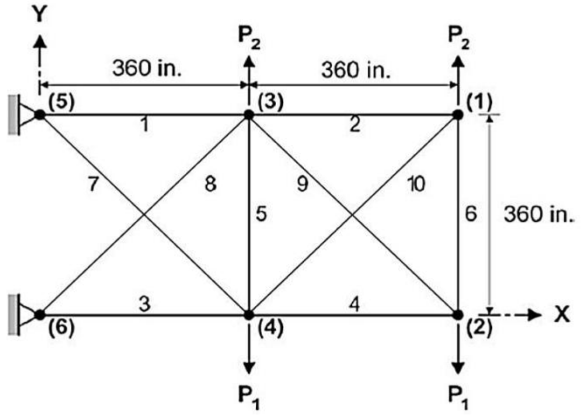
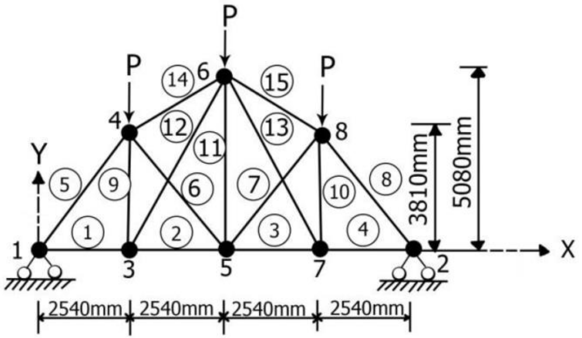
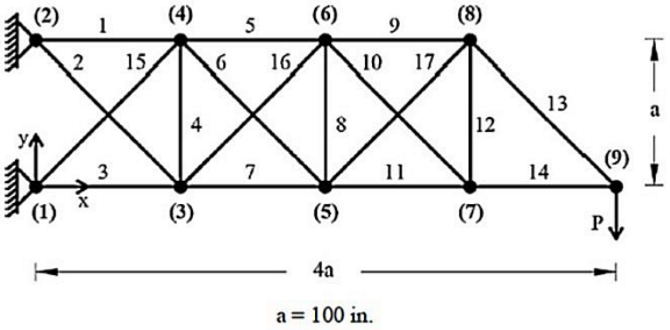
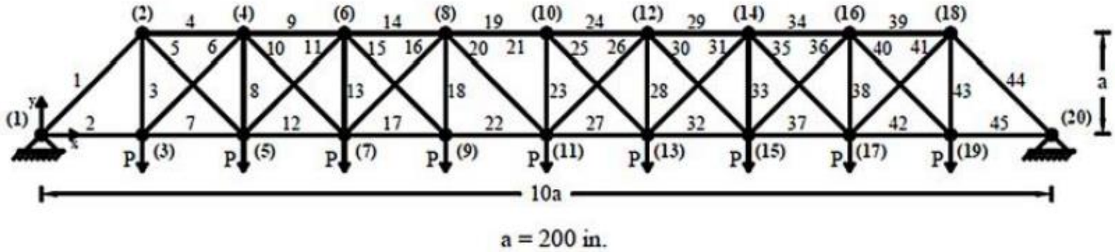

Next position x_i^{k+1}
Global best position of the swarm g_{best}^{k}
Direction of global best position
Direction of next position
New motion with velocity v_i^{k+1}
Influence of the swarm
Direction of personal best position
Personal best position p_i^{best}
Current position x_i^k
Influence of current motion with velocity v_i^k
Influence of Particle memory

Fig. 22: Schematic representation of a velocity component construction of a PSO algorithm (Esmin 2013, Al-Shamman 2018)

AKBARI & HENTEH (2019) compared GA and PSO for discrete and continuous size optimisation of 2D truss structures (see Tab. 1).

Tab. 1: Different sizes of truss structures (AKBARI & HENTEH 2019)

Six-node, 10-member truss

Eight-node, 15-member truss

Nine-node, 17-member truss

Twenty-node, 45-member truss


Generally, truss structures are optimised in three ways:

- Optimising the size or cross-section, in which case the cross-section of the members is selected as the design variable and the coordinates of the nodes and topology of the structure are fixed
- Optimisation of the shape, in which the coordinates of nodes are considered as design variables
- Optimisation of the topology, in which the connection of the members is examined.

In their study, AKBARI & HENTEH (2019) have based the decision variables on the fixed section area of the members from two-dimensional truss structures with fixed topology and shape. The investigation included different cases of four truss structures with different material properties for three different sizes in discrete or continuous size and load. In particular there are differences in elastic modulus E, density, maximum allowed stress, maximum allowed node displacement in horizontal and vertical direction and load. The assessment of the member is related to the sections in between a given number of nodes.

The OpenSees software was linked with codes of GA and PSO in MATLAB. The objective function aimed in weight minimisation. The constraints were limits of the member stresses and node displacements. A penalty function was combined with the objective function for considering violations of these constraints. For GA, the population size was set to 100, the percentage of crossover to 80 %, and the percentage of mutation to 30 %. As selection operator, the Roulette wheel method was chosen. The Fitness function was based on the exponential relation of the value of the penalised objective function to the maximum value of the objective function. The structure weight reduction process with the **N**umber of **F**unction **E**valuation (NFE) was observed during the execution of both algorithms, GA and PSO. The amount of weight obtained from PSO is far less than GA. AKBARI & HENTEH (2019) drew the conclusion, that GA is the most generally economical ADM for discrete problems, while PSO is the most economical ADM for continuous problems. The comparison of the weight loss diagrams in terms of number of simulations (NFE) shows that the convergence of GA to the optimum solution is faster than PSO.

Besides GA, Global Optimisation algorithms can be based on different Evolutionary Automatic Programming techniques like Genetic Programming, Grammatical Evolution and **G**ene **E**xpression **P**rogramming (GEP).

GEP aims in improving the adaptive fit of an expressed program for a problem specific cost function. The program is a candidate solution, which consists of symbols and functions, whereas each symbol maps to a function or terminal node of an expression tree in a breadth-first manner. On a node, the linear string of symbols is called Karva notation or a K-expression. It is like a gene expression, whereas the gene of fixed length is divided into two parts, the head and the tail. The head is related to the function or terminal symbols, and the tail is a kind of genetic buffer with only terminal symbols. The gene expressions forming simple linear chromosomes, define a genotype-phenotype system, which can be transferred to programming. The application of evolutionary mechanisms like genetic recombination, mutation, inversion and transposition is related to a learning process resulting in adapting the sizes, shapes, and composition of the tree structures controlled by minimising the cost function. With the schematic representation of GEP algorithm, the procedure can be visualised (see Fig. 23).

Before the evolution process of the GEP can start, the function set, terminal set and load dataset for the fitness evaluation have to be selected. Then, the chromosomes of the initial population of programs are created randomly. An iteration process starts for each program in the population based on gene expression of chromosome, execution of the program and evaluation of the fitness. The evaluation of the fitness leads to the verification of the stop condition, which terminates the GEP evolution process in case the stop condition becomes true and the optimal solution is found. Otherwise, for the next generation the programs are selected, whose chromosomes have been evaluated to have the highest fitness. These programs were reproduced to form the blank of the next generation. On the blank, genetic operators for modifying the chromosomes are applied in order to create the new generation. The evolution process of GEP continues until the stop condition becomes true and the optimal solution is found.

The advantage of GEP is the ability to express the relationships between independent variables and the response using a mathematical equation, whereas the mathematical equation can be highly nonlinear.

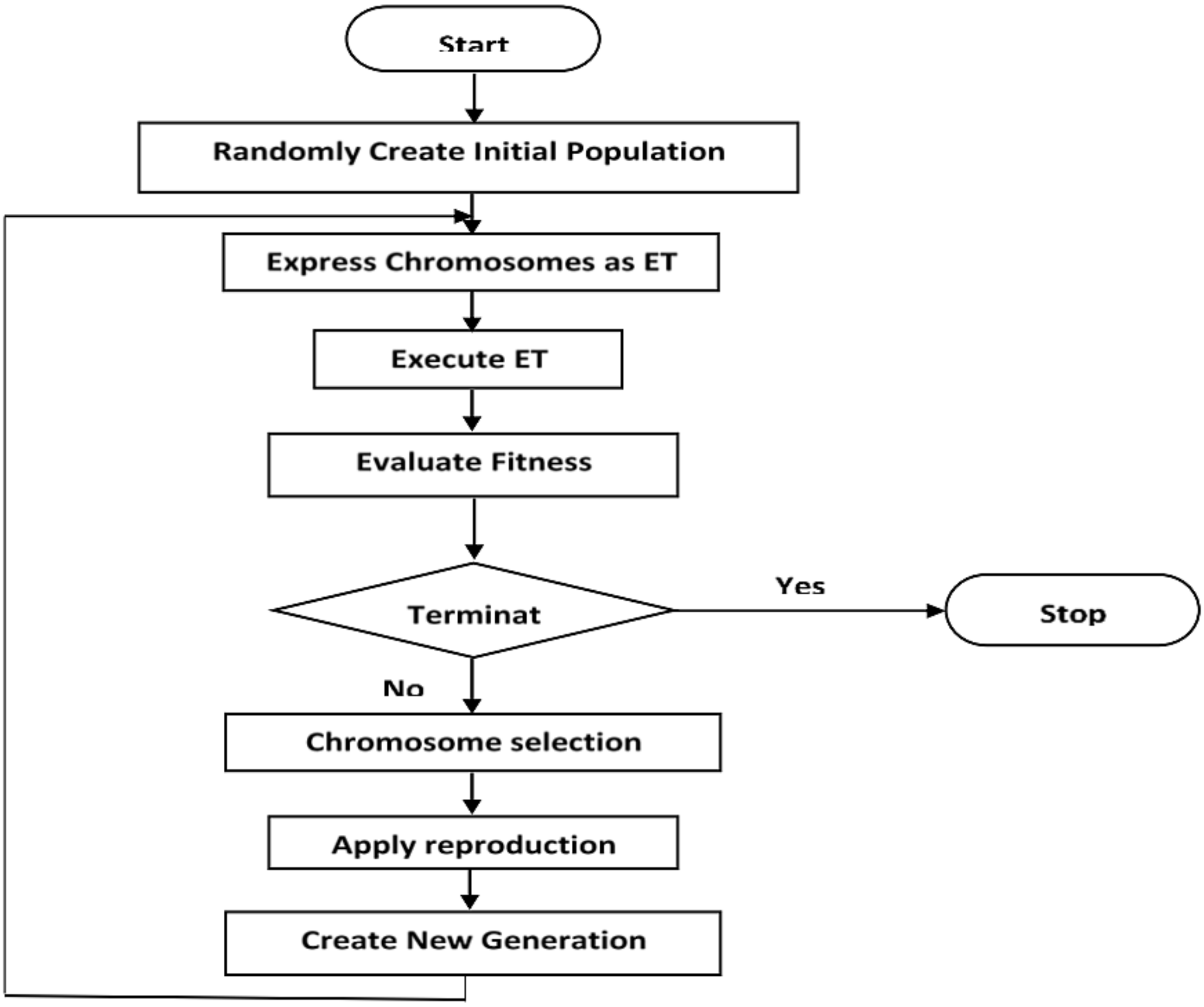


Fig. 23: Schematic representation of GEP algorithm (AKIN & ABEJIDE 2019)

GEP was used by AKIN & ABEJIDE (2019) for modelling the concrete compressive strength admixed with **G**round-**G**ranulated **B**last-**F**urnace **S**lag (GGBFS) as **S**upplementary **C**ementitious Materials (SCMs). GEP enabled the usage of a highly nonlinear function for modelling the compressive strength of concrete. An experimental dataset was used and analysed in order to identify the main input variables: Portland cement, GGBFS, fine aggregate, coarse aggregate and water.

The GEP was realized for 30 chromosomes with 3 genes, a mutation rate of 0.00138, and various recombination rates of 0.00277. Within a huge computational time, 100 000 to 500 000 generations for the concrete mix dataset were created with GEP. The evaluation was based on different performance measures:

- Coefficient of determination (R^2)
- Mean square error (*MSE*)
- Root Mean Squared Error (*RMSE*),
- Mean absolute error (*MAE*)
- Relative standard error (*RSE*)

Compared to stepwise regression analysis using SPSS software with just a linear function from literature, the accuracy based on R^2 of GEP was proved to be significantly better. The predictive ability of GEP comes closer to the real solution, and thus it is more accurate than the classical statistical regression analysis.

Besides evolutionary-based ADM, there are biologically inspired optimisation ADM, which form an own subtype of ADM. This subtype is inspired by biological neural networks and their learning ability to perform tasks. For imitating the biological concept, artificial neurons are used to model neurons in a biological brain. Networks of artificial neurons are called **A**rtificial **N**eural **N**etworks (ANN).

In analogy to the natural information processing, artificial neurons process input information in nonlinear functions to produce an output signal, which might be send to connected neurons in an artificial neural network. The input signal consists of property values in form of real numbers. Summarising all input signals with a bias and an activity function like the sigmoid function or hyperbolic tangent, results in an output signal, which is additionally weighted. These weights influence the impact and usage of output signals and the artificial neurons, which can be interpreted in strengthening or weakening the connection between artificial neurons of an artificial neural network, which is used to perform an optimisation task. The process of changing the weights for finding the optimal solution is called training or learning like explained by Hebb (see chapter 1).The learning process of an ANN is based on training and testing, where input data is separately used, for which the solution is already known. The training is an iterative process, where the weights are changed to calculate an output signal, which is evaluated by the known solution by minimising the mean square errors. The main challenge is the question of knowing, when the iteration can be stopped, because the training already succeeded. Regarding, the computational effort and time, which increases with the training time, as well as the risk of overfitting, shorter training time in form of less iterations seems to be reasonable, in so far as sufficient accuracy is given to the real solution. The testing evaluates a trained ANN. A successfully trained ANN can be applied for similar optimisation tasks in significant shorter times than running an ADM from scratch.

There are many different types of ANN (see Fig. 24), which may use different types of artificial neurons regarding categorical or numerical functions used for information processing, as well as several architectural structures of connections between the artificial neurons based on layers, rings or combinations.

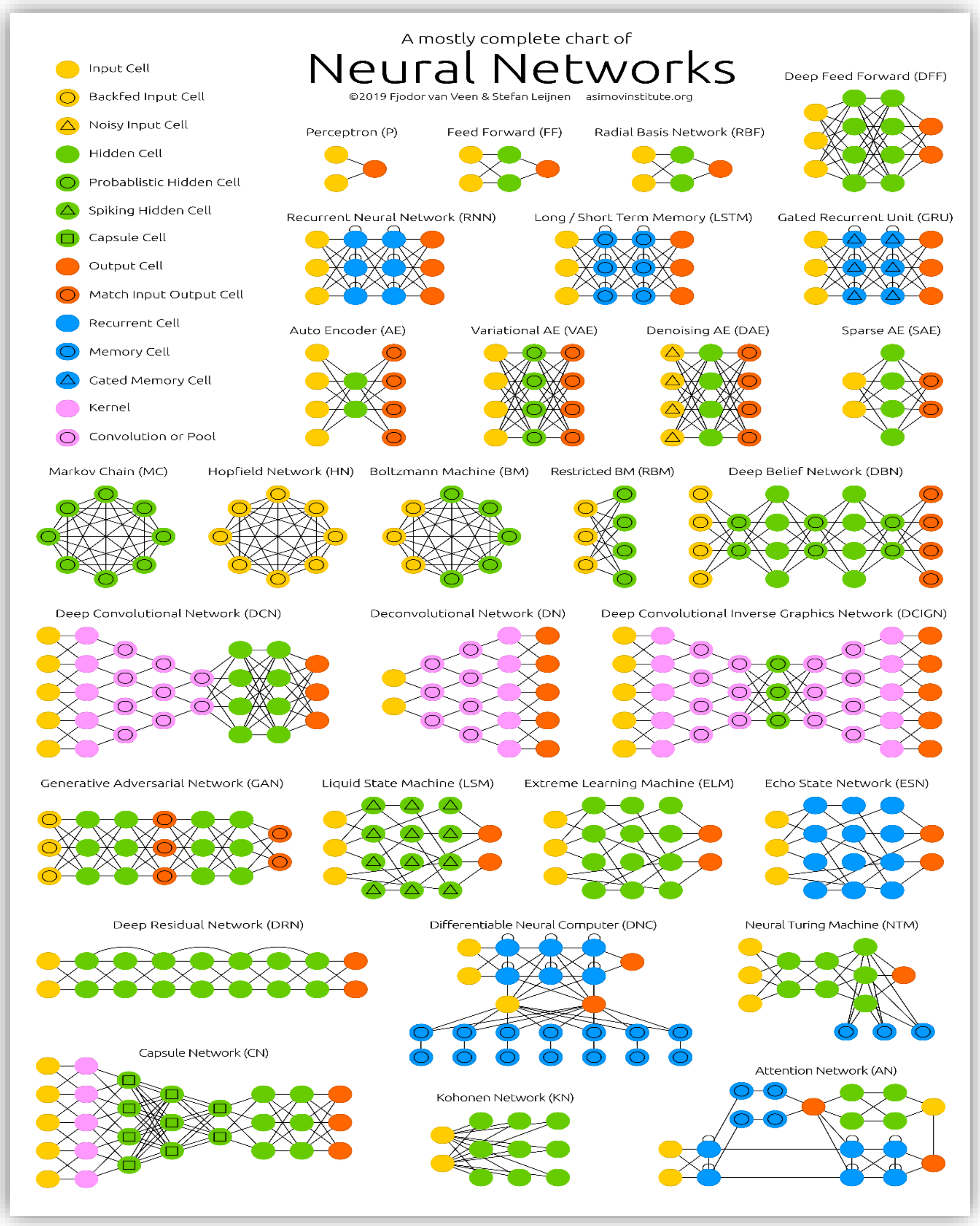

A mostly complete chart of
Neural Networks
©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org
Input Cell
Backfed Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Capsule Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Gated Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Differentiable Neural Computer (DNC)
Neural Turing Machine (NTM)
Capsule Network (CN)
Kohonen Network (KN)
Attention Network (AN)

Fig. 24: Different types of neural network architectures (VAN VEEN & LEIJNEN 2019)

An ANN (see Fig. 25) was used by MANE et al. (2019) in order to predict the flexural strength of concrete produced by using Pozzolanic materials and manufactured sand MA, which are partly replacing **N**atural **F**ine **A**ggregate (NFA).

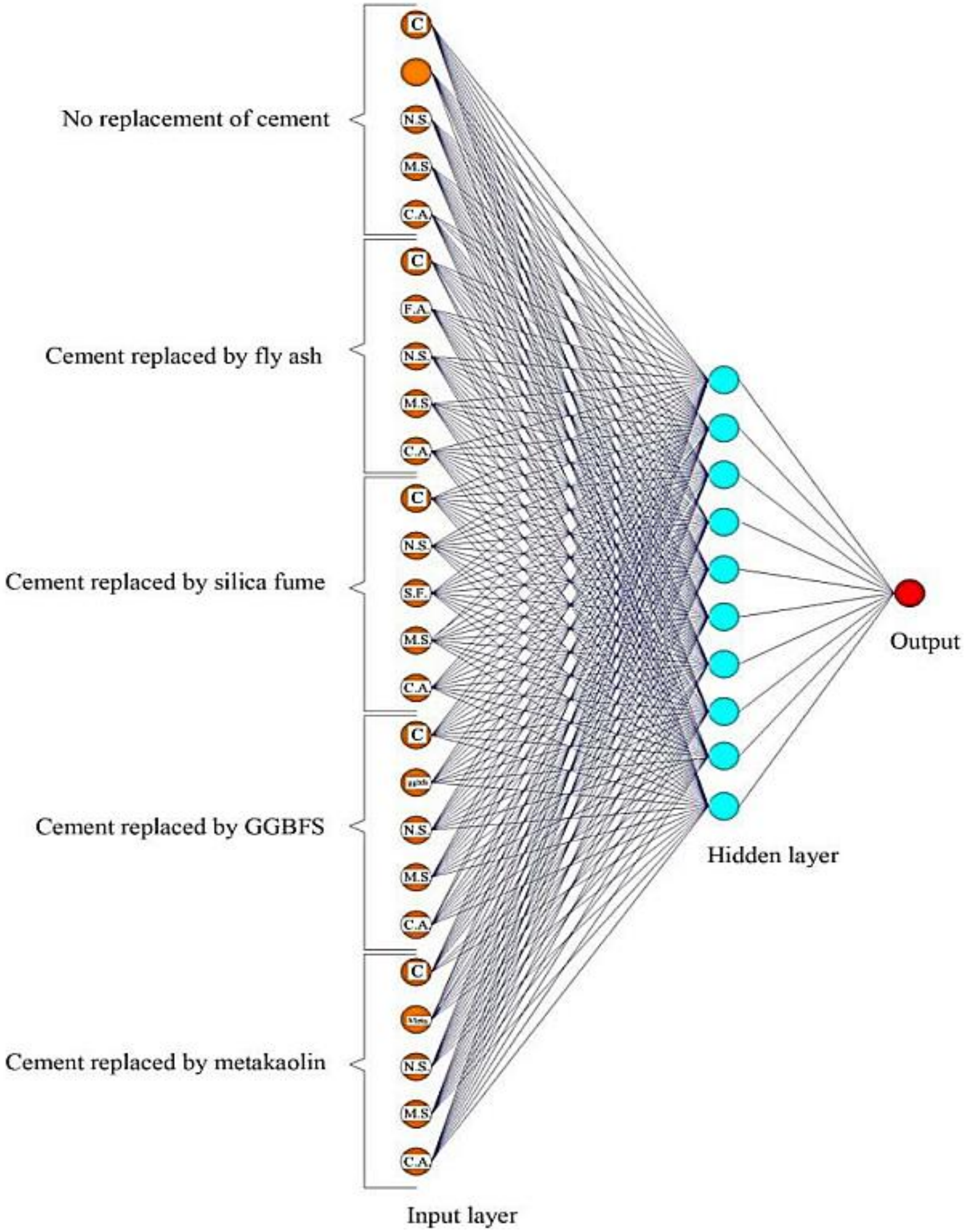


Fig. 25: Structure of the ANN used for predicting the flexural strength of concrete depending on different replacements (MANE et al. 2019)

Five replacement cases were investigated:

- No replacement
- Cement partly replaced by fly ash
- Cement partly replaced by silica fume
- Cement partly replaced by GGBFS
- Cement partly replaced by metakaolin

Flexural strength was experimentally determined on casting beam specimens to develop an ANN model based on MATLAB code. For the training, 70 % of the data was used; the rest was used for testing.

The evaluation was realised with three different performance measures:

- Coefficient of determination (R^2)
- Mean square error (*MSE*)
- Root Mean Squared Error (*RMSE*)

The *MSE* for the different replacements were compared and meant to be in the acceptable range. All obtained simulation results are agreeable and a strong correlation was observed between experimental and predicted flexural strength values.

For predicting the **U**ltimate **B**earing **C**apacity (UBC) of strip footing resting on dense sand overlaying loose sand deposits, DUTTA, RAO & SHARMA (2019) compared the performance of three techniques:

- Random Forest Regression (RFR)
- M5P model tree (M5P)
- Artificial Neural Network (ANN)

For computing the UBC, a sensitivity analysis was performed (see Fig. 26).

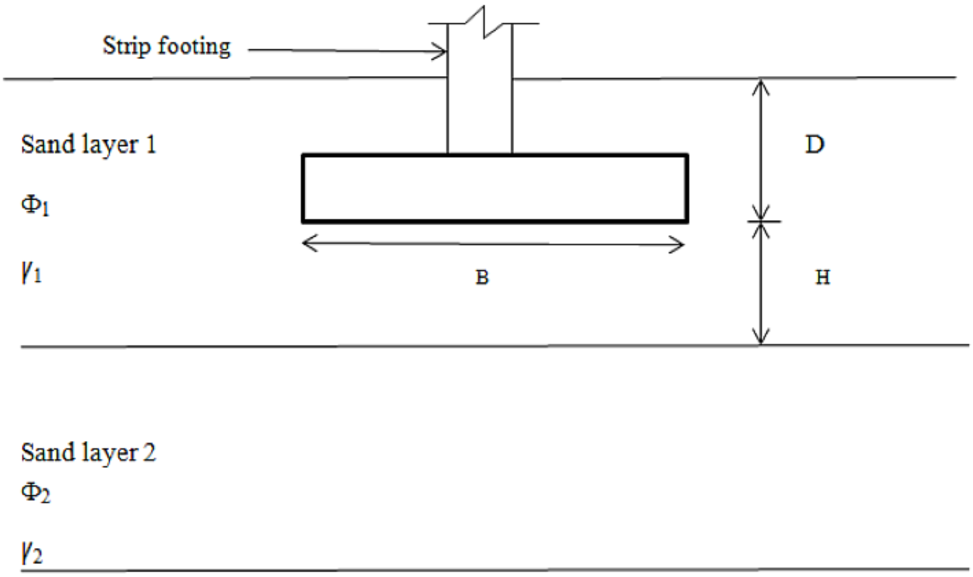


Fig. 26: Strip footing based in layered soil (DUTTA, RAO & SHARMA 2019)

In the sensitivity analysis the major input parameters were studied:

(1) Friction angle of the dense sand layer

(2) Friction angle of the loose sand layer

(3) Unit weight of the dense sand layer

(4) Unit weight of the loose sand layer

(5) Ratio of the thickness of the dense sand layer below base of the footing to the width of footing

(6) Ratio of the depth of the footing to the width of the footing

(7) Ratio of the ratio of this thickness (5.) to the ratio of this depth (6.)

Additionally, the sensitivity analysis revealed, that unit weight and friction angle of the loose sand layer are the most important parameters.

For prediction of UBC of strip footing on layered soil based on experimental and theoretical data reported in literature, RFR, M5P and ANN were used. For bagging of RFR, 70 % of the original data was used for the training and 30 % was left out from every tree grown. Realising overfitting is negligible, fully grown trees were not allowed to prune back with any prune method. In consequence, RFR could be better maximising the expected error reduction than M5P.

For ANN, a feed forward back propagation algorithm with a sigmoid function was used with *Weka 3.8* software. The sigmoid activation function has been proven to be the most accurate as it yields the minimum errors.

The evaluation was based on different performance measures:

- Correlation coefficient (*r*)
- Coefficient of determination (R^2)
- Mean square error (*MSE*)
- Root mean square error (*RMSE*)
- Mean absolute error (*MAE*)
- Mean absolute percentage error (*MAPE*)

The comparison of the performance measures results in a ranking with an ascending error for the prediction models: RFR, ANN and M5P.

A hybrid intelligent system is a **F**uzzy **N**eural **N**etwork (FNN) or a **N**euro-**F**uzzy **S**ystem (NFS) (LIN & LEE 1996; RAJASEKARAN & PAI 2017) which combines the concepts of ANN and fuzzy logic. The combination is realised by using fuzzy sets, and a linguistic model for conditional rules. In contrast to classical set theory, the membership of elements in a fuzzy set is not assessed in binary terms, but in degrees of membership. The degrees of membership were obtained by a membership function like a Gaussian membership function, where the gradual assessment is valued in the interval between 0 and 1.

Special forms of NFS are **A**daptive **N**euro-**F**uzzy **I**nference **S**ystems (ANFIS). ANFIS may use several inference mechanisms for fuzzy datasets like Takagi Sugeno controller or Tsukamoto controller. With the controller, new parameters were derived from the input parameters: The membership function coefficients and the coefficients of the linear output functions.

An ANFIS was developed by KALANTARI et al. (2019) for predicting the performance of shear connectors in composite structures (see Fig. 27 and Fig. 28). The neuro-fuzzy model is based on a fuzzy c-means algorithm to predict the shear strength of the shear connectors in composite frames. Since the shear strength involves the consideration of the behavioural complexity of two materials, concrete and steel, ANN with a fuzzy system was preferred for adjusting the system rules. Besides shear strength as output parameter, four input parameters related to the material properties, were used as members in the fuzzy sets:

- Compressive Strength of Concrete
- Total area of concrete dowels
- Area of transverse reinforcement bars in rib holes multiplied by yield stress reinforcement bars in rib holes
- Connector height

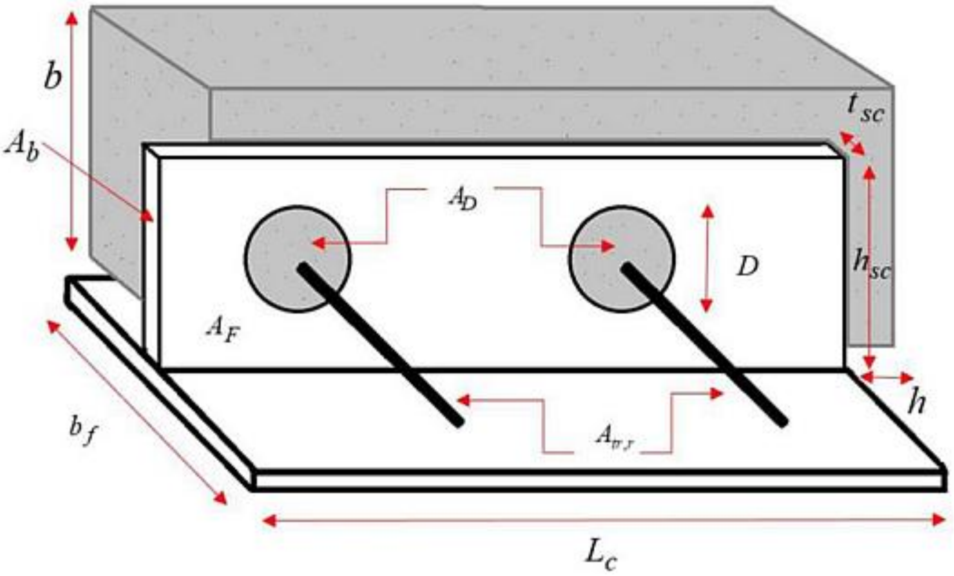


Fig. 27: Shear connector (KALANTARI et al. 2019)

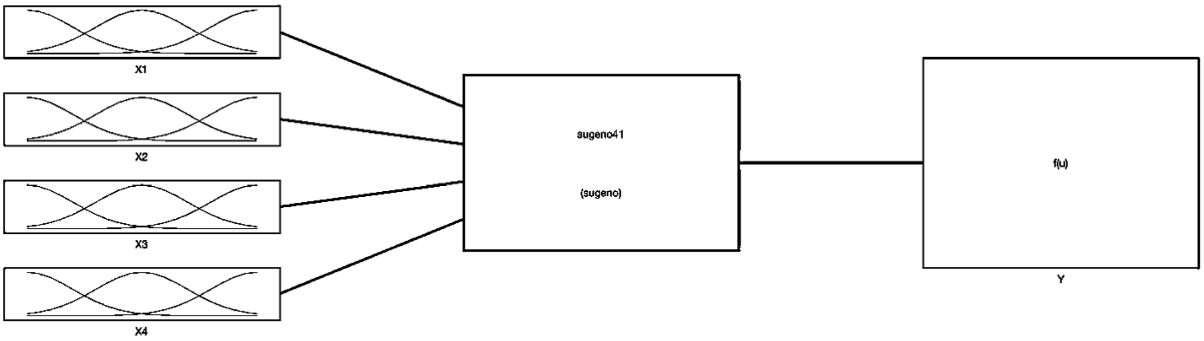


Fig. 28: General structure of a proposed ANFIS (KALANTARI et al. 2019)

With Gaussian membership functions, the coefficients of the input parameters were determined, while the coefficients for the output were obtained by four linear output functions. Each Gaussian membership function has two unknown parameters, the variance and the mean.

To train the ANFIS, a fuzzy clustering approach with c-means is applied. The advantage to the sub-clustering approach is that the fuzzy clustering approach needs fewer clusters to present the best answer. Also, fuzzy c-means is more accurate and faster than the grid partitioning algorithm.

For the evaluation, different performance measures were used:

- Coefficient of determination (R^2)
- Mean absolute error (*MAE*)
- Root mean squared error (*RMSE*)

After the evaluation with acceptable errors, KALANTARI et al. (2019) conclude that ANFIS is an appropriate framework to predict the shear strength of the shear connectors in composite frames.

NFS can also be used for data stream mining, when it is sequentially updated with new incoming samples on demand and on-the-fly. Additionally to the adaption of model parameters, a dynamic evolution and prune method for model components like neurons and rules are applied in order to handle changes of the target value over time (concept drift).

Summarising, and pointing out the potential of optimisation algorithms in geotechnical engineering (WAQAS 2018), ADM like RFR, M5P, ANN and NFS were successfully applied for:

- Predicting the earthquake magnitude from previous time series
- Predicting the **U**ltimate **B**earing **C**apacity (UBC) of different strip footings
- Finding the optimum repair and maintenance scenario for a corroded **R**einforced **C**oncrete (RC) structure in the marine environment
- Optimising two dimensional truss structures
- Modelling of concrete compressive strength admixed with **G**round-**G**ranulated **B**last-**F**urnace **S**lag (GGBFS) as Supplementary Cementitious Materials (SCMs)
- Predicting the flexural strength of concrete produced by using Pozzolanic materials and Manufactured sand MA, partially replacing **N**atural **F**ine **A**ggregate (NFA)
- Predicting the performance of shear connectors in composite structures
- Modelling a 3D geospatial environment based on **A**ugmented **R**eality (AR)

Further, many rock engineering problems may be solved with various ADM according to (MORGENROTH et al. 2019, see Tab. 2).

Tab. 2: Selection of rock engineering problems and possible ADM for solving them (MORGENROTH *et al.* 2019)

Rock Engineering Problem	***ADM***
Rock mass properties	Categorical ANNs
Laboratory testing and constitutive behaviour	Numerical ANNs SVM
Slope stability	Categorical ANNs SVM RF Clustering
Point cloud analysis	RF kNN
Tunnel performance	Categorical / Numerical ANNs SVM RF
Rock bursts	Categorical ANNs kNN RF SVM Decision Trees
Blasting	Categorical / Numerical ANNs SVM RF

4 Literature

AKBARI, M. & HENTEH, M. (2019): Comparison of Genetic Algorithm (GA) and Particle Swarm Optimization Algorithm (PSO) for Discrete and Continuous Size Optimization of 2D Truss Structures, *Journal of Soft Computing in Civil Engineering*, 3(2): 78–98

AKIN, O. O. & ABEJIDE, O. S. (2019): Modelling of Concrete Compressive Strength Admixed with GGBFS Using Gene Expression Programming, *Journal of Soft Computing in Civil Engineering*, 3(2): 43–53

BAYER, V. (2019): Unit 4: Robustness Evaluation and Robust Design, optiSLang E-learning, Dynardo GmbH

CATERPILLAR (2020): Wherever there's mining, we're there, Website, http://s7d2.scene7.com/is/content/Caterpillar/C840945,
last accessed: 25.05.2020

CHASSIAKOS, A. P. & REMPIS, G. (2019): Evolutionary Algorithm Performance Evaluation in Project Time-Cost Optimization, *Journal of Soft Computing in Civil Engineering*, 3(2): 16–29

CIFA (2020): Tunnel support systems, Website, http://www.cifa.com/formworks/-/16/1, last accessed: 25.05.2020

DUTTA, R. K., RAO, T. G. & KHATRI, V. N. (2019): Application of Soft Computing Techniques in Predicting the Ultimate Bearing Capacity of Strip Footing Subjected to Eccentric Inclined Load and Resting on Sand, *Journal of Soft Computing in Civil Engineering*, 3(2): 30–42.

DUTTA, R. K., RAO, T. G. & SHARMA, A. (2019): Application of Random Forest Regression in the Prediction of Ultimate Bearing Capacity of Strip Footing Resting on Dense Sand Overlying Loose Sand Deposit, *Journal of Soft Computing in Civil Engineering* , 3(4): 28–40.

ESMIN, A. A., COELHO, R. A. & MATWIN, S. (2015): A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data, *Artificial Intelligence Review*, 44(1): 23–45.

FARAHANI, A. (2020): Life Cycle Cost GA Optimization of Repaired Reinforced Concrete Structures Located in a Marine Environment, *Journal of Soft Computing in Civil Engineering,* 4(2019): 41–50.

FRIDHI, A. & FRIHIDA, A. (2019): GIS 3D and Science of Augmented Reality: Modeling a 3D Geospatial Environment, *Journal of Soft Computing in Civil Engineering*, 3(4): 78–87.

HAGHSHENAS, S. S., MIKAEIL, R., KAMRAN, M. A. & GHAREHGHESHLAGH, H. H. (2019): Selecting the Suitable Tunnel Supporting System Using an Integrated Decision Support System, (Case Study: Dolaei Tunnel of Touyserkan, Iran), *Journal of Soft Computing in Civil Engineering*, 3(4): 51–66.

HEBB, D. O. (1950): The Organization of Behavior: A neuropsychological theory, *Science Education*, 34(5): 336–337

JASPER, H. (2020): Modul Künstliche Intelligenz, *Lecture notes, Institut für Informatik (TUBAF)*, Freiberg

KALANTARI, S. M., MORTAZAVI, S. M. & TAFAZZOLI, M. S. (2019): Fuzzy-Based

Approach to Predict the Performance of Shear Connectors in Composite Structures, *Journal of Soft Computing in Civil Engineering*, 3(4): 1–11

KONIETZKY, H. & SCHLEGEL, R. (2013): Optimization, Sensitivity and Robustness Analysis -- Applications and Trends in Geomechanics, *EUROCK2013*, *Proceedings*, 69–77

LANCE, P. & HATTORI, A. (2016): Sampling and Evaluation – A Guide to Sampling for Program Impact Evaluation — MEASURE Evaluation, Website, https://www.measureevaluation.org/resources/publications/ms-16-112, last accessed: 25.05.2020

LIN, C.-T. & LEE, C. S. G. (1996): Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems, *Prentice-Hall*, 797

MANE, K. M., KULKARNI, D. K. & PRAKASH, K. B. (2019): Prediction of Flexural strength of Concrete Produced by Using Pozzolanic Materials and Partly Replacing NFA by MS, *Journal of Soft Computing in Civil Engineering*, 3(2): 69–77.

MORGENROTH, J., KHAN, U. T. & PERRAS, M. A. (2019): An overview of opportunities for machine learning methods in underground rock engineering design, *Geosciences (Switzerland)*, 9(12)

MOST, T., ROTERMUND, J., WILL, J. & GRÄNING, L. (2019): Artificial Intelligence and Machine Learning - Application in Computer Aided Engineering, 16. Weimarer Optimierungs- & Stochastikertage, *Dynardo GmbH*, Weimar.

RAJASEKARAN, S. & PAI, G. A. V. (2017): Neural networks, fuzzy systems, and evolutionary algorithms : synthesis and applications, 2nd Revised Edition.

SHISHEGARAN, A., TAGHAVIZADE, H., BIGDELI, A. & SHISHEGARAN, A. (2019): Predicting the Earthquake Magnitude along Zagros Fault Using Time Series and Ensemble Model, *Journal of Soft Computing in Civil Engineering*, 3(4): 67–77.

VAN VEEN, F. & LEIJNEN, S. (2019): The Neural Network Zoo - The Asimov Institute, Website, https://www.asimovinstitute.org/neural-network-zoo/?utm_content=buffer607bd&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer, last accessed: 25.05.2020

WAQAS, M. (2018): Discrete element and artificial intelligence modeling of rock properties and formation failure in advance of shovel excavation, *Ph .D. Thesis,* Department of Mining and Nuclear Engineering, Missouri University of Science and Technology, USA