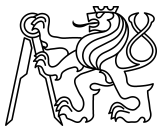
Czech Technical University in Prague



Faculty of Civil Engineering

Soft computing methods in concrete mix performance approximation and optimization

Matěj Lepš

Habilitation thesis

Prague, December 2009

Acknowledgments

The presented work was supported by the Ministry of Education, Youth and Sports under the project MSM 6840770003 and by the Grant Agency of the Czech Republic GAČR 103/07/P554. Fruitful discussions with professors and colleagues, in particular Z. Bittnar, J. Zeman, V. Šmilauer, A. Kučerová, Z. Vitingerová, M. Valtrová, J. Nosek, A. Pospíšilová and E. Myšáková are greatly appreciated.

Most importantly, I would like to thank all family members and friends for their never ending encouragement and support that help me to fully concentrate on this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Modeling of hydrating concrete represents a challenging task especially due to multiscale nature and missing mathematical formulation of several underlying phenomena. This opens a way for soft computing techniques, namely a cellular automata-based hydration model, non-traditional optimized Design of Experiments for sensitivity analysis, evolutionary algorithms applied to identification purposes, Kriging and Genetic Programming approximations and, last but not least, Evolutionary Multi-objective Optimization methods.

The term *soft computing* [109] is widely used for all approaches that do not solve the given problem directly, like e.g. an analytical search for a solution of differential equations. Oppositely, the aim is to guess the optimal solution as fast as possible yet precisely as possible. And any trick that helps to find an appropriate solution in shorter time is welcomed. Particularly, Cellular Automata (CA) [106] is a tool that was originally aimed at the simulation of an artificial life. Later on, CA have been used for the optimization purposes leading to the development of Genetic Algorithms (GAs) [32] and the whole Evolutionary Algorithms (EAs) area [55] that tries to mimic processes in nature to solve the given optimization tasks. Similar history is behind Kriging [62], which is the method originally used to discover new mining areas for gold, diamonds etc. Twenty years later, the nice mathematical properties of Kriging have led to its usage within the approximation of computer experiments [82]. Genetic Programming (GP) [48] can be seen as an enhancement of GAs, however, from the philosophical point of view, the goal is to find a solution in the form of tree structures. Therefore, GP application is very wide, from programming of autonomous systems, through placing of transistors on a computer chip up to a symbolic regression presented in this work.

1.1 Aim and scope

The aim of this work is to show possible applications of soft computing tools within one specific problem of a material design domain. Since the main emphasis of the presented work is placed on soft computing methods, the phenomenon of hydrating cement is only sketched in the next section. The objectives are concentrated at an approximation and optimization of hydration heat by above mentioned techniques, taking into account chemical and physical cement composition and boundary conditions. The works starts with the Design of Experiments (DoE) for sampling-based sensitivity analysis to investigate the influence of individual components and processes during concrete hydration, as is presented in Chapter 2. The identification of important models' parameters from obtained experimental data is briefly summarized in Chapter 3. Then, the combination of approximation techniques, so-called Kriging metamodel combined with the Genetic Programming (GP) algorithm presented in Chapters 4 and 5, aims at prediction of hydration heat solely from experimental data sets. The Kriging method gives good prediction when monotonicity evolution of hydration heat is assumed. Finally, Chapter 6 presents the multi-objective optimization of maximal Young's modulus and minimum of hydration heat.

1.2 Simulation of cement performance[1]

Concrete is one of the most durable, reliable and cheap construction materials. Cement represents an essential component and cement paste governs a majority of concrete properties such as elasticity, strength, creep, freeze-thaw resistance, permeability, or durability. All components for concrete production are readily available throughout the world and cement kilns with concrete mix facilities are located virtually everywhere.

While the production of cement in the kilns is controlled under nearly laboratory conditions, concrete casting and curing on site suffers from vary-

[1]Reproduced from: Z. Bittnar, M. Lepš, and V. Šmilauer. *Soft Computing in Civil and Structural Engineering*, chapter Soft computing in concrete mix optimization, pages 227–246. Stirling: Saxe-Coburg Publications, 2009.

ing boundary conditions including a human factor. As a consequence, higher uncertainty is needed to cover possible variations and this is reflected in design codes by a higher material safety factor, e.g. 1.5 while 1.05 for steel[2]. Moreover, elementary rules have to be fulfilled for durable concrete as has been summarized in many guidelines and books [74, 73]. Although such rules summarize hundred-year experience, they are neither exhaustive nor complete and leave an open way for further refined analysis.

The first penetration of computers into concrete engineering came in the field of structural analysis and reinforcement design. The solution of equations coming from physical description proved to be very efficient. The opposite is true on the material side, where the complex and multiscale concrete nature is captured only partially in the mathematical framework. It is worth mentioning that computational micromechanics paved the design of engineered cementitious composites.

Soft computing techniques bring another alternative to tackle a description of complex systems such as hardening concrete. Missing physical and mathematical description is replaced here by cellular automata providing a robust framework which was previously used in the simulation of panic escape during fire [30]. Here, the automata will describe ongoing chemical reaction during concrete hydration and the formation of cement microstructure.

1.2.1 Cement hydration model CEMHYD3D and its inputs

The simulation at the scale of cement paste will be carried out by hydration model called CEMHYD3D, developed at NIST [6], see Fig. 1.1. The idea is to split the representative microstructure into voxels usually with an edge of 1 μm. The 3D microstructure, typically in the size of 50 $\times$ 50 $\times$ 50 μm, consists of chemical phases that are implemented as an ID assignment to each voxel. Cellular automata rules define how voxels dissolve, move and what happens on their collision thus representing ongoing chemical reaction and placement of hydration products in the microstructure.

The model can capture directly the chemical composition of cement (C_3S,

[2]Eurocode 2: Design of concrete structures, ČSN EN 1992-1-1

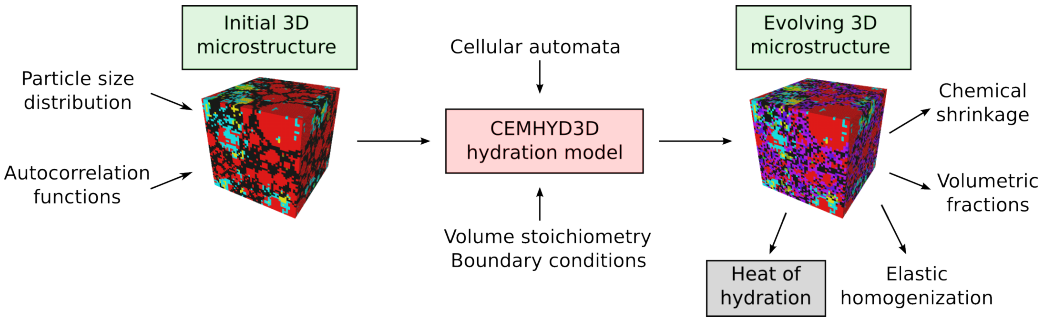


Figure 1.1: A flowchart of hydration model with consecutive outputs

C_2S, C_3A, C_4AF, gypsum), distribution of cement particle sizes, hydrating temperature and saturated/sealed conditions, see [100] for more details. The optimization in Chapter 6 will therefore focus on changing these inputs.

Within this work, standard CEMHYD's input values were used and the model capabilities were extended with the prediction of C-S-H_{LD} and C-S-H_{HD} to include additional mechanical stiffening at confined capillary space [101]. Cycle-time mapping parameter $\beta = 0.0003$ was used throughout for the mapping of CEMHYD3D cycles to a real hydration time. Microstructure size of $50 \times 50 \times 50$ μm was assumed to be representative up to approximately 28 hydration days. The drawback of using such a small size is a limited maximum grain diameter to 31 μm. The consequence is a slight overestimation of hydration degree at later stages especially in coarse cements. All simulations took place at isothermal 20 °C.

The selection of input parameters is chosen in such a way that CEMHYD3D is able to take them realistically into account in the simulations. Their ranges correspond approximately to contemporary limiting values of Portland cements used in civil engineering, see Tab. 1.1.

The particle size distribution (PSD) is expressed by the value of Blaine fineness [m^2/kg]. Rosin-Rammler cumulative function is fitted to PSD data [29] with x [μm] being the particle diameter

Parameter	*Minimum*	*Maximum*
Fineness [m^2/kg]	200	600
Gypsum content [vol. of cement]	0	0.1
Autocorrelation file designation	0	9
C_3S [cement mass fraction]	0.4	0.8
C_2S [cement mass fraction]	0	0.35
C_3A [cement mass fraction]	0	0.15
C_4AF [cement mass fraction]	0	0.15
w/c	0.2	0.6
Curing condition regime	0-saturated	1-sealed

Table 1.1: Input parameters with limits used in virtual tests

$$G(x) = 1 - exp(-bx^n) \tag{1.1}$$

$$n = -0.00083333 \cdot fineness + 1.1175 \tag{1.2}$$

$$b = 0.000754 \cdot fineness - 0.143 \tag{1.3}$$

Before the simulation of hydration, spherical cement grains have to be thrown into the computational volume. Since Portland cement is composed basically from four clinker minerals, each grain has to be consequently divided into silicates, aluminates and later to individual clinker minerals, relying on a triplet of autocorrelation functions [29]. Ten triplets obtained from CCRL cements were taken from the NIST cement database to explore their effect [29].

Saturated or sealed curing conditions result in a different morphology of cement paste. While the saturation ensures enough water to percolated capillary space, sealed conditions slow down the hydration process due to the lack of water, implemented as emptying of larger capillary pores with an influence on reactions.

1.2.2 Affinity hydration model

Since the CEMHYD3D model is computationally demanding, sometimes a utilization of a much simpler model is on demand. An affinity model is such

a tool that provides a simple framework describing all stages of cement hydration. The rate of hydration can be expressed by temperature-independent *normalized chemical affinity* $\tilde{A}(\alpha)$

$$\frac{\mathrm{d}\alpha}{\mathrm{d}t} = \tilde{A}(\alpha) \exp\left(-\frac{E_a}{RT}\right), \tag{1.4}$$

where T is an arbitrary constant temperature of hydration, R is the universal gas constant (8.314 $Jmol^{-1}K^{-1}$) and E_a is the apparent activation energy.

The affinity can be obtained experimentally easily from calorimetry. Isothermal calorimetry measures a heat flow $q(t)$ from a sample and quantifies, after an integration, the hydration heat $Q(t)$. Recognizing $Q(t)/Q_{pot}$ as a degree of hydration DoH leads to an approximation which has been slightly modified in [100]

$$\tilde{A}(\alpha) = B_1 \left(\frac{B_2}{\alpha_\infty} + \alpha\right)(\alpha_\infty - \alpha) \exp\left(-\bar{\eta}\frac{\alpha}{\alpha_\infty}\right) \tag{1.5}$$

where B_1, B_2 are coefficients to be calibrated, α_∞ is an ultimate hydration degree and $\bar{\eta}$ represents a microdiffusion of free water through formed hydrates. The parameters for a general type of cement were set to $B_1 = 1.0 \cdot 10^7$ 1/h, $B_2 = 2.0 \cdot 10^{-4}$, $\alpha_\infty = 0.9$ and $\bar{\eta} = 7.6$. Since these constants are used for all possible cement pastes, a general curve of DoH development has been obtained by numerical integration of Equation 1.5 with one hour step. A potential hydration heat Q_{pot} can be obtained from the portland cements' mineral

Designation	C3S	C2S	C3A	C4AF	Gyp.	w/c	Fin.
	Mass %	Mass %	Mass %	Mass %	Vol %	-	m^2/kg
Aalborg white	0.666	0.238	0.034	0.004	0.036	0.400	390
Principallo	0.554	0.184	0.082	0.091	0.051	0.375	530
BAM Fontana	0.492	0.243	0.090	0.076	0.0652	0.300	380
Hua	0.688	0.075	0.081	0.092	0.04	0.420	400
Robeyst	0.634	0.084	0.074	0.100	0.05	0.500	390
Smolik_Litos	0.612	0.126	0.070	0.100	0.05	0.500	306
Tamtsia early	0.465	0.246	0.104	0.083	0.05	0.500	340

Table 1.2: Cement properties of considered data sets

Designation	**Hydration heat (time in h, heat in J/g of cement)**			
Aalborg white	24	48	168	[h]
	170.3	234	327	[J/g]
Principallo	9.42	80.24	400.00	[h]
	63.388	323.247	377.466	[J/g]
BAM Fontana	10.01	144.03	310.69	[h]
	159.2624	295.6692	322.3247	[J/g]
Hua	24.00	168.00	600.00	[h]
	233.4	317.25	339.8	[J/g]
Robeyst	14.66	45.79	140.99	[h]
	94.07	238.623	348.757	[J/g]
Smolik_Litos	10.01	19.19	261.78	[h]
	59.9159	329.2083	466.1589	[J/g]
Tamtsia early	18.00	24.00	102.00	[h]
	279.4076	307.3484	447.0522	[J/g]

Table 1.3: Experimental results of hydration heat scaled to a reference temperature 20 °C

composition

$$Q_{pot}\ [\mathrm{J}] = 517\mathrm{m}_{\mathrm{C_3S}} + 262\mathrm{m}_{\mathrm{C_2S}} + 1144\mathrm{m}_{\mathrm{C_3A}} + 725\mathrm{m}_{\mathrm{C_4AF}} \tag{1.6}$$

with the masses in grams.

Although the value of potential heat is uniquely given by the amount of clinker minerals, see Equation 1.6, it cannot be directly used with the equation of the DoH development (1.5) because of unknown B_1, B_2 coefficients for particular cement paste[3]. In a detail, the general expression of hydration heat $Q(\cdot)$ reads

$$Q(\mathcal{C}, \mathcal{B}, t) = Q_{pot}(\mathcal{C}) \cdot DoH(\mathcal{B}, t), \tag{1.7}$$

where $\mathcal{C} = \{C3S,\ C2S,\ C3A,\ C4AF\}$ stands for the amount of clinker minerals, $\mathcal{B} = \{B_1,\ B_2,\ \alpha_\infty,\ \bar{\eta}\}$ are affinity model parameters and t is time. Note that Q_{pot} is independent on time. As a general solution, Chapter 5 shows

[3] B_1, B_2 coefficients are somehow dependent on all input parameters, however, these relationships are unknown and the search for them exceeds the scope of this work.

an attempt to find an analytical expression of hydration heat Q dependent directly on all input parameters that concurrently best fits the given data.

1.3 Experimental data

Particular applications will be shown on experimental data see Tab. 1.2 and Tab. 1.3, obtained from seven sources, consecutively from the top: data measured at CTU by TAMAir isothermal calorimeter, from [78], data from private communication and determined from evaporable water content and assumed potential hydration heat 480 J/g, from [40], [80], data measured at CTU by TAMAir isothermal calorimeter and finally, from [94] assuming potential heat 500 J/g. In Tab. 1.2 cement chemical composition, gypsum content, a water to cement ration w/c and fineness are presented. Measurements of hydration heat at three different times recalculated for the same reference temperature of 20 °C are shown in Tab. 1.3. For more points in data sets, see Fig. 3.2 in Chapter 3.

Chapter 2

DESIGN AND ANALYSIS OF MIXTURE EXPERIMENTS

2.1 Introduction

Space-Filling Design Strategies[1] known as *a Design of Experiments* (DoE) constitute an essential part of any experimentation. The DoEs' area is very wide covering a broad range of domains from real experiments through computer simulations to sampling for uncertainty analysis. An interested reader is referred to the fundamental book [65] for DoEs with real experiments, and works [82] and [27] for introduction on computer experiments.

This chapter is aimed at one particular domain of DoEs. We are interested in a composition of cement which is the most frequent example of constrained design spaces called *a mixture experiment*, where individual inputs (variables, factors, components) form a unity volume or unity weight [65, Chapter 11-5]. This only condition leads to the simplex space; further limits of individual inputs then form *a polytope*, still convex but generally irregular space. Therefore, all traditional DoEs [65] that are constructed for hypercube spaces cannot be applied here.

Although the problem of mixture experiments is known for decades, the progress of methods for DoEs does not follow current development within the area of computer experiments [27]. The main difference between classical and modern DoEs is the number of samples where, for the latter, the hundreds of samples is a usual scenario. Then, the classical approaches based on fixed small-sample templates [17, 18] cannot be used. Up-to-date, the authors have found only few references on DoEs in constrained design spaces. Ref-

[1] Since we are interested in traditional way of doing experiments, whether they are real or simulated ones, space-filling properties are investigated hereafter. An interested reader is therefore referred to nice works [31] and [44] for references on other criteria for optimized DoEs like discrepancy and correlation coefficients.

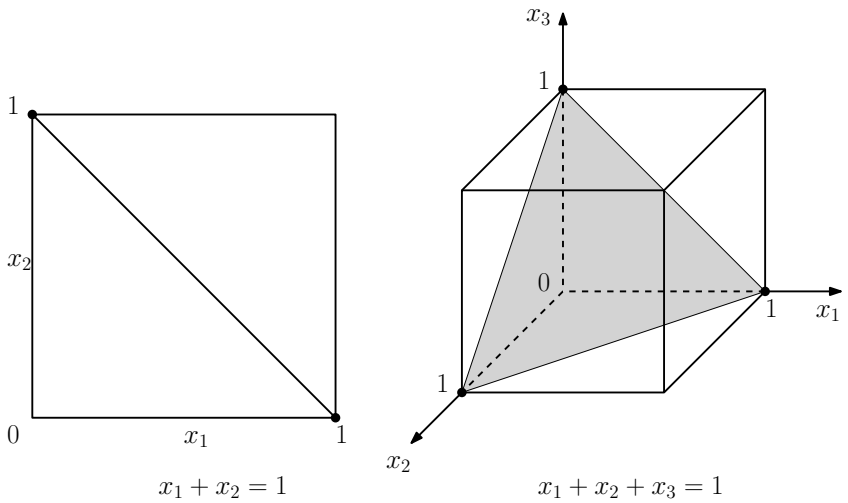


Figure 2.1: Mixture experiments spaces for (a) two and (b) three factors

erence [76] applies traditional Latin Hypercube (LH) designs to a bounding box followed by a Genetic Algorithm (GA) to fulfil original constraints. Here, the LH methodology is merely used for minimization of the searched space than for nice properties of LH designs. Another approach is presented in [38], where interesting points are found by a GA and then, the final solution is located by sequential linear programming. The solution is in this case general; however, the computational demands are enormous. Therefore, in this chapter a totally different approach based on *Delaunay triangulation* (DT) of an admissible domain is presented.

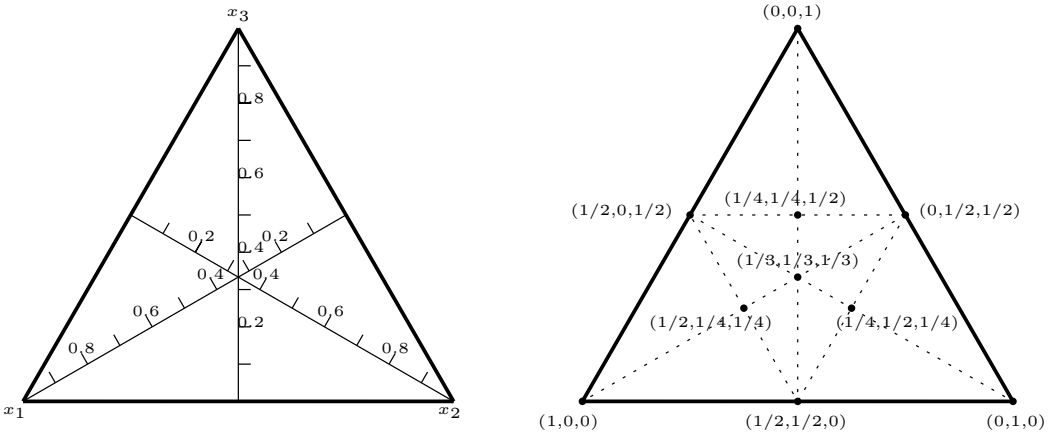
2.2 Designs of mixture experiments

In contrast to the classical Design of Experiments, the mixture experiment is characterized by dependent variables forming a unity mass or volume. More specifically, if there are p components of a mixture, then

$$0 \leq x_i \leq 1 \qquad i = 1, 2, \ldots, p \tag{2.1}$$

and

$$x_1 + x_2 + \cdots + x_p = 1 \qquad \text{(i.e. 100 percent)}. \tag{2.2}$$



(a) (b)

Figure 2.2: Trilinear coordinate system (left) and corresponding barycentric coordinates (right)

Such situation is depicted in Fig. 2.1 for two and three components, respectively. In the first case, all solutions form a line segment, in the second case, solutions come from a triangle. Generally, the admissible space for the mixture experiment is a $(p-1)$-dimensional simplex. Conveniently, individual components can be represented by barycentric coordinates [20], as a special case for $p = 3$ known as *triangular*, *trilinear* or *areal coordinates* see Fig. 2.2 and examples within the Finite Element Method (FEM) [28].

The classical uniformly spaced designs for a simplex domain, so-called *simplex lattice designs* [65, Chapter 11-5], can be constructed by proportionally changing barycentric coordinates. Assuming $m+1$ equally spaced values from 0 to 1 such that

$$x_i = 0, \frac{1}{m}, \frac{2}{m}, \ldots, 1 \qquad i = 1, 2, \ldots, p \tag{2.3}$$

and all possible combinations of these proportions that fulfil Eq. (2.2) form a $[p, m]$ lattice design with

$$M = \frac{(p+m-1)!}{m!(p-1)!} \tag{2.4}$$

number of points. For example, if $p = 3$ and $m = 2$, then $x_i = 0, \frac{1}{2}, 1 \quad i =$

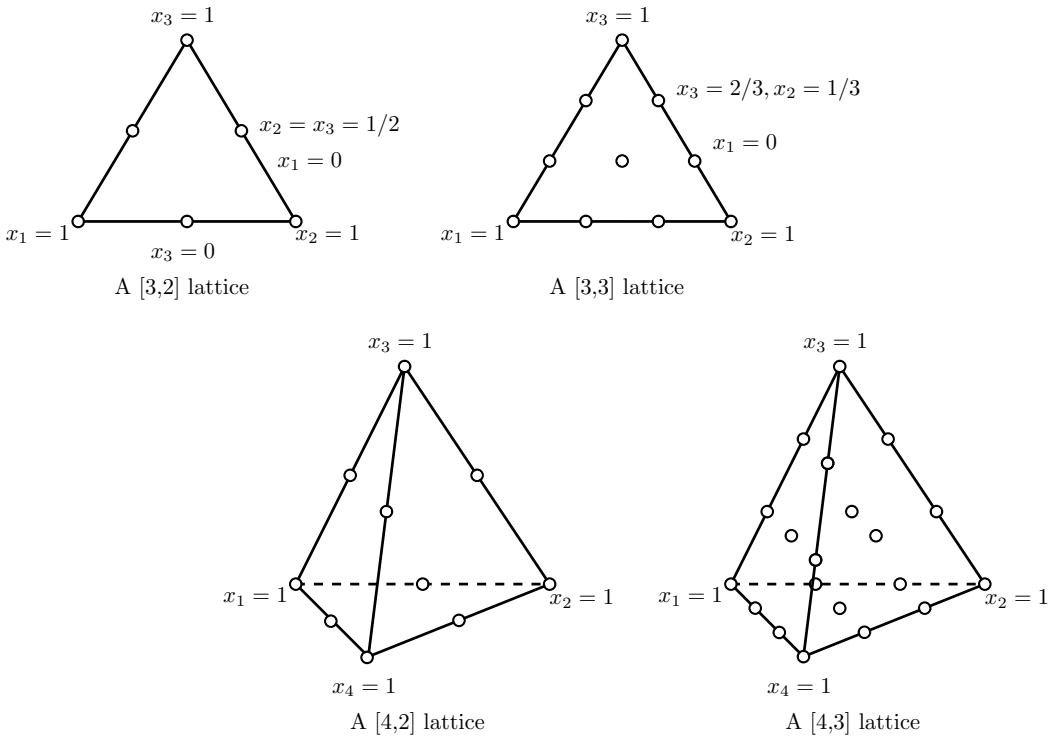


Figure 2.3: Examples of simplex lattice designs for three and four components

$1, 2, 3$ and the $[3, 2]$ simplex lattice design is created by the following six points, see Fig. 2.3:

$$(1,0,0); (0,1,0); (0,0,1); (\frac{1}{2}, \frac{1}{2}, 0); (\frac{1}{2}, 0, \frac{1}{2}); (0, \frac{1}{2}, \frac{1}{2}) \quad . \tag{2.5}$$

2.3 Designs of constrained mixture experiments

In case some limits are placed on individual components, i.e.

$$0 \leq a_i \leq x_i \leq b_i \leq 1, \qquad 1 \leq i \leq p \quad , \tag{2.6}$$

the admissible domain is no more simplex and forms a convex *polytope*. The problem of the mixture of cement clinker minerals from Section 1.2.1 will serve as an illustrative example. Since the problem composes of four components, the admissible space in case of no constraints is a tetrahedron, see Fig. 2.4c).

Here, the individual bounds for four clinker minerals (see also Tab. 1.1) are set as

$$\begin{aligned} 0.4 \leq \quad & C_3S & \leq 0.8 \\ 0 \leq \quad & C_2S & \leq 0.35 \\ 0 \leq \quad & C_3A & \leq 0.15 \\ 0 \leq \quad & C_4AF & \leq 0.15 \end{aligned}$$

and form a convex space bounded by 12 nodes and 8 faces. The sequence of the polytope visualization is shown in details in Fig. 2.4. The classical approaches for real experiments [18] will try to place experiments in vertices of polytope, however no simple methods are available for a bigger number of experiments in case of computer experiments.

2.4 DoE Methods based on classical approaches

Here, we present a methodology how to solve the constrained mixture DoE problem without solving the nonlinear optimization problem as is presented in [38]. The method is based on the creation of the bounding box of the admissible space and then, application of the traditional designs applicable for the particular bounding box. As traditional designs, one can assume all ranges of algorithms, from factorial designs [65], through Latin Hypercube Sampling (LHS) [15, 95] to quasi random numbers generators [58].

The simplest way is to define a bounding hypercube of the given polytope. Hence, any popular method for the DoEs for hypercube spaces [27] can be applied. The selected procedure then generates samples from the hypercube and admits only those points lying inside the original admissible space until the prescribed number of samples is attained. However, an ability to generate a required number of samples dramatically degenerates with the growing number of dimensions. Imagine an example of the N-dimensional simplex created with the vertex at the origin $(0, \ldots, 0)$ and vertices on individual axes $(1, 0, \ldots, 0)$, $(0, 1, \ldots, 0)$, ..., $(0, 0, \ldots, 1)$. The volume of this simplex is $V = \frac{1}{2^N}$ of the unit bounding hypercube. Then, e.g. for $N = 20$ only one design out of $1,048,576$ samples from the unit hypercube will lie in the orig-

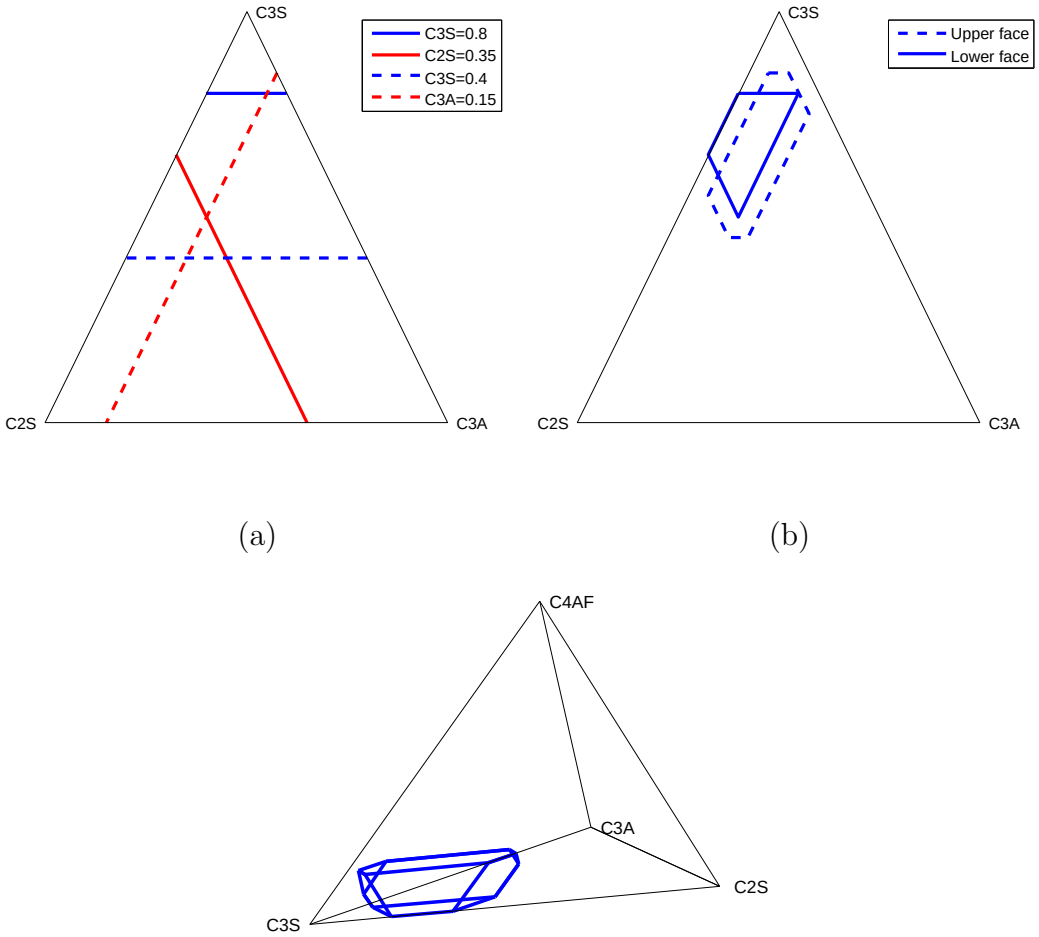
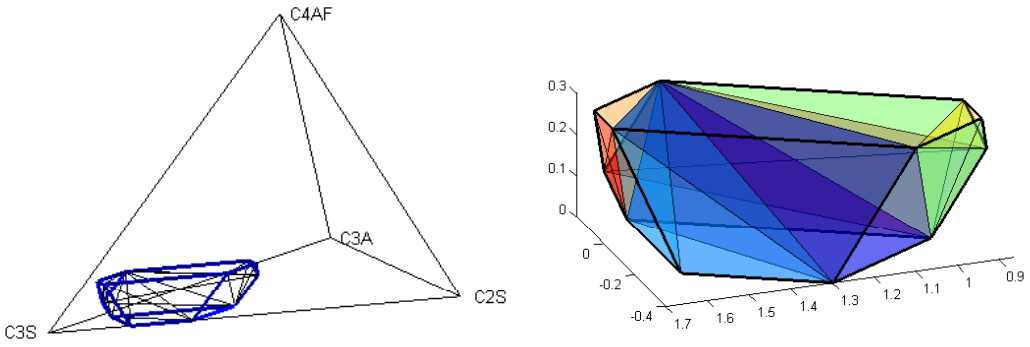


Figure 2.4: Constraints for cement mixture experiment: (a) contours at lower face ($C_4AF = 0$), (b) admissible spaces at lower and upper ($C_4AF = 0.15$) faces and (c) the whole 3D polytope

inal simplex. Therefore, such approach is suitable only for smaller number of dimensions.



(a) (b)

Figure 2.5: Triangulation of the admissible domain for the example of the mixture experiment by Delaunay triangulation: (a) a wire model and (b) a volume model

2.5 Delaunay triangulation based methods

A triangulation is a term suitable for 2D, generally it means the partition of the domain by simplexes. Delaunay triangulation (DT) is the most popular triangulation method [12]. It is based on a convex hull of given points V describing the admissible domain, where the convex hull is the smallest convex set containing all points in V. Then, DT triangulates the convex hull such that there is no point of V inside the circumsphere of any simplex in the triangulation.

Because it is relatively simple to create DT[2], see Fig. 2.5, and then compute a volume of simplexes, see Appendix A, we have a rough estimation, how is the admissible region formed. An example of utilizing such methodology has been firstly presented in [19] for regular design spaces. We extended this idea for constrained design spaces by incorporating three methodologies. The first one (RND) generates quasi random points in the triangulated admissible domain. The second method (RMV) starts with a doubled number of

[2]We assume that the admissible domain is defined by the set of vertices for which DT exists. If not, there is still possibility to add more temporary points inside the domain to create valid DT.

random points from the RND method. By removing unfavorable points the algorithm then reaches the optimum. The last methodology (DM) is based on the analogy with a dynamical system. The detailed description of individual methods follows.

2.5.1 A uniform random point generator for simplex meshes (RND)

First, a generator for uniform random numbers for a general simplex is needed. There are several solutions, however the one presented in [22] is very simple. It starts with the required number of samples from exponential distribution, i.e. sampling $\mathbf{X}$ from $[0,1]^p$ uniformly, and returning $-log(\mathbf{X})$. In such a way, the required number of samples is obtained. Then, uniform samples within the simplex are created only by normalization of obtained samples from the exponential distribution.

The algorithm for random uniform samples within the triangulation by simplexes is again simple - each simplex will contain a portion of the required number of samples based on a ratio of its volume to the total volume of the admissible space[3]. And again, since the computation of the simplexes' volumes is simple, see Appendix A, the procedure is for a smaller number of dimensions very fast.

2.5.2 Removal of superfluous points (RMV)

The above presented procedure is fast and reliable; however, the space-filling quality is usually very poor. Recently, we have tried to find a fast and reliable method for generating uniformly spaced samples within regular domains [69]. One of the most promising methods is based on the removal of superfluous points from overcrowded random designs generated with above-mentioned generator. Particularly, we construct a design with two times more points than needed, and then, we are repeatedly removing the point that creates the worst Euclidean Maximin distance until the original number

[3]Note that since the generation of points is dependent on the volumes of the individual simplexes creating the whole admissible region, the final distribution of points is not random, but quasi random.

of points is attained. Although this procedure is at least two times slower than the random number generator, the obtained results justifies expended effort, see Section 2.6.

2.5.3 A `Distmesh` *tool (DM)*

The `Distmesh` tool (DM) [75] is a heuristic smoothing algorithm for generating uniform meshes [12]. It is well-known that the most uniform meshes for the Finite Element Method (FEM) are characterized with uniformly spaced nodes (but not vice-versa!). Therefore, we have tried to utilize this nice property of the DM tool. The DM is based on a simple dynamical system of an expanding pin-jointed structure. Those trusses that are too short are causing repulsive forces that move the too close nodes apart. The main disadvantage apart from high computational demands is the need to return nodes that leave the prescribed admissible domain. The DM offers solutions for basic entities, however, a general simplex as well as a triangulation is missing. The description of our solution follows.

At every iteration of the DM tool, one has to solve three problems:

Which points are outside the domain is in our approach resolved by enumeration of the barycentric coordinates. If a point has at least one negative barycentric coordinate in accordance to a given simplex, then the point is outside this simplex. Therefore, at every simplex we search for a minimal value of barycentric coordinates. If the maximum of these minima over all simplexes is negative, then the point is outlier to the admissible domain.

In which direction to move outliers is the most crucial problem. The original task is, for every outlier, to move itself to the nearest point on the boundary of the admissible domain. The problem is that the boundary is in general N-dimensional convex hull created with a huge number of entities, see e.g. the list of entities for a general simplex up to five dimensions presented in Appendix B. Therefore, we have not used a direct analytical solution. Instead, Monte Carlo approach has been

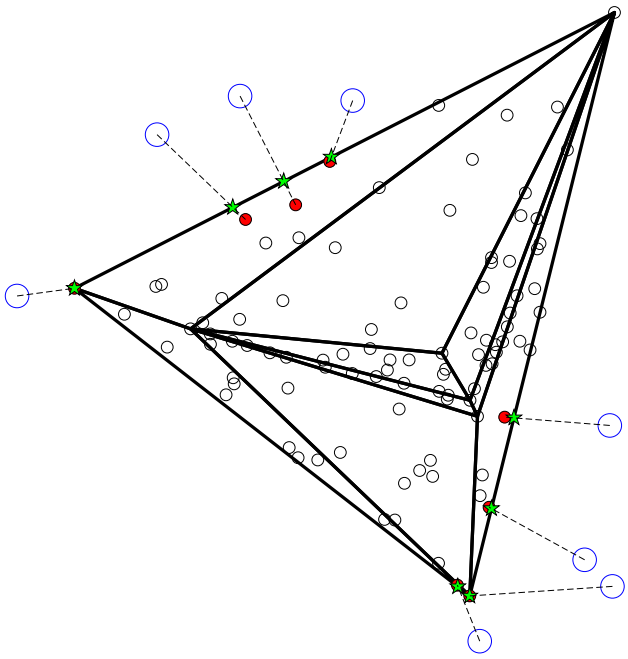


Figure 2.6: A random illustrative example of moving outliers (big blue circles) to the nearest random point on the boundary (vertices) and the search for intersection (green star) with the boundary in case the random point inside is not on the boundary (red filled circle)

used to describe the volume of the admissible space. One hundred of temporary random points are created within each simplex. All vertices describing the topology of the convex hull are added as well. Then the search for nearest boundary is replaced by the search for the nearest point from the temporary list, see illustrative example in Fig. 2.6.

How far to move outliers can be done analytically or numerically. We have selected the numerical solution. If the temporary point lies directly on the boundary, then is replaced by the outlier and this vertex is removed from the temporary list to prevent duplicities, see Fig. 2.6. In case the nearest point is inside the domain, the simple bisection method is used to find the intersection of the direction from the previous step with the domain boundary, see again Fig. 2.6, and the outlier is moved

to the intersection. All these three steps are done consecutively for all points lying outside the domain.

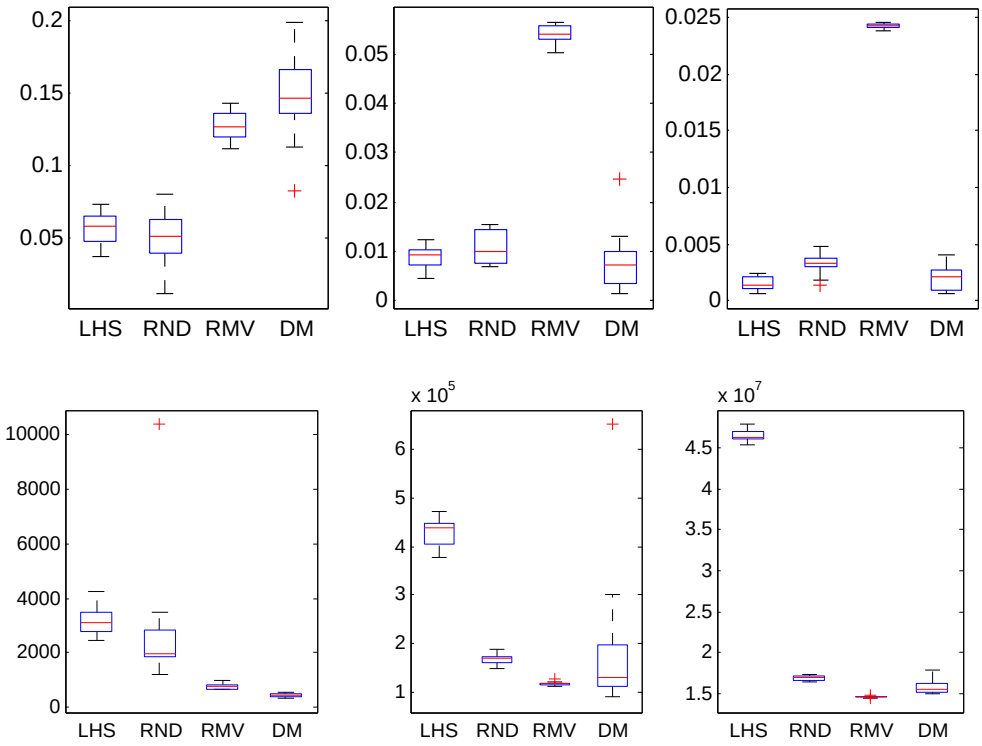


Figure 2.7: Comparison of four methods for generating constrained mixture designs: (Top row) Euclidean Maximin - higher is better and (Bottom row) Audze-Eglais potential energy - lower is better; (From left) results for 10, 100 and 1000 samples. Key: LHS = Latin Hypercube Sampling within bounding box, RND = random point generator in Delaunay triangulation, RMV = removal of superfluous points from overcrouded RND and DM = Distmesh tool

2.6 Comparison of methods

Since we are interested in space-filling properties, two most common objective functions are examined. The first is *Euclidean Maximin* metric (EMM) [97,

41] for its simplicity and easiness in visualization. The EMM is the minimal distance out of all distances between any two design points and is to be maximized. From the experiments point of view EMM expresses the worst case scenario of the closeness of two experiments. Even for computer experiments the assumption that an evaluation is costly is still valid. Therefore, the possible duplicity of two closed points remains a crucial task.

The second measure is *Audze-Eglais* objective function (AE) proposed by Audze and Eglais in [4]. It is based on an analogy with a potential energy of the set of points. The points are distributed uniformly when the potential energy E^{AE} proportional to the inverse of the squared distances among points is minimized, i.e.

$$E^{\mathrm{AE}} = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{1}{L_{ij}^2} \quad , \tag{2.7}$$

where N is the number of design points and L_{ij} is the Euclidean distance between points i and j. Since the objective is a sum of distances, it is not heavily disturbed by outliers from the potential energy point of view. Therefore, such measure represents an average property of the set of points.

Up-to-now three methods have been described in detail - a uniform random point generator for simplex meshes (RND), removal of superfluous points (RMV) and a modified Distmesh tool (DM). The last method that will be compared is the classical bounding approach presented in Section 2.4. It is a traditional Latin Hypercube Sampling (LHS) method, see e.g. [5], from the Matlab environment. Samples are generated in a bounding box of the admissible space presented in previous parts. The procedure consecutively increases the number of samples in LHS until sufficient points fall inside the admissible domain. Note that Matlab's implementation is not directly optimized for given metrics; with the default settings, MATLAB generates 5 set of samples and the best from the EMM point of view is introduced to a user. Our experience [69] shows that the performance of this algorithm can be improved by the heuristic procedure at least twice in terms of the EMM metrics.

The final results for the comparison of these four methods are presented in Fig. 2.7. The analysis has been done for 10, 100 and 1000 samples in the final design. For the statistical purposes, the results are presented as

a statistic out of ten independent runs. The results for EMM objective are presented in Fig. 2.7 in the top row. Note that the higher the boxplot is, the better the EMM property is. The LHS and RND methods perform equally. The DM methodology has outstanding performance in case of lower numbers of dimensions and samples. With both growing numbers the performance of the DM method deteriorates. The removal-based method is in terms of EMM metric clear winner.

The differences among methods based on the AE objective function are not so extreme. Results for the second objective are presented in Fig. 2.7 in the bottom line. Note that in case of AE the lower values are better. Only the LHS method is losing, however this can be improved by incorporating an optimization procedure as mentioned previously.

Since the codes have not been deeply optimized from implementation point of view, the analysis of computational demands cannot be rigorously done. However, we can state general requirements of the methods. The RND generator is the fastest one, with no optimization cycle. The RMV method is the second, since it needs two times more samples than RND. Moreover, there is a loop on removal of points. Although the LHS methods is for regular spaces one of the fastest, its performance is deteriorated here because of repetitive growing of number of samples until a sufficient number of samples fall inside the admissible domain. The DT based method is the most demanding one. There is several Delaunay triangulations inside the loop of the Distmesh tool that are needed to preserve the inner structure to be physically consistent. And still, as is visible from the EMM performance, the Distmesh has problems with the quality of the boundary surface mesh, see also the discussion e.g. in [12].

2.7 Conclusions

The Design of Experiments for constrained spaces and computer experiments is relatively new and unexplored area. The constraint in the form of sum of all inputs equal to one complicates the application of all contemporary DoE algorithms for regular design spaces. The presented chapter is a pioneer-

ing work that brings a totally new methods and unpublished results. Moreover, the modification of the Distmesh tool can enable improvement within the smoothing methods for FEM meshes for triangulated spaces. Finally, it is important to note that all presented methods are independent on the number of dimensions. Only the computational demands can limit the application of some of the methods in higher dimensions.

Chapter 3

PARAMETER ESTIMATION

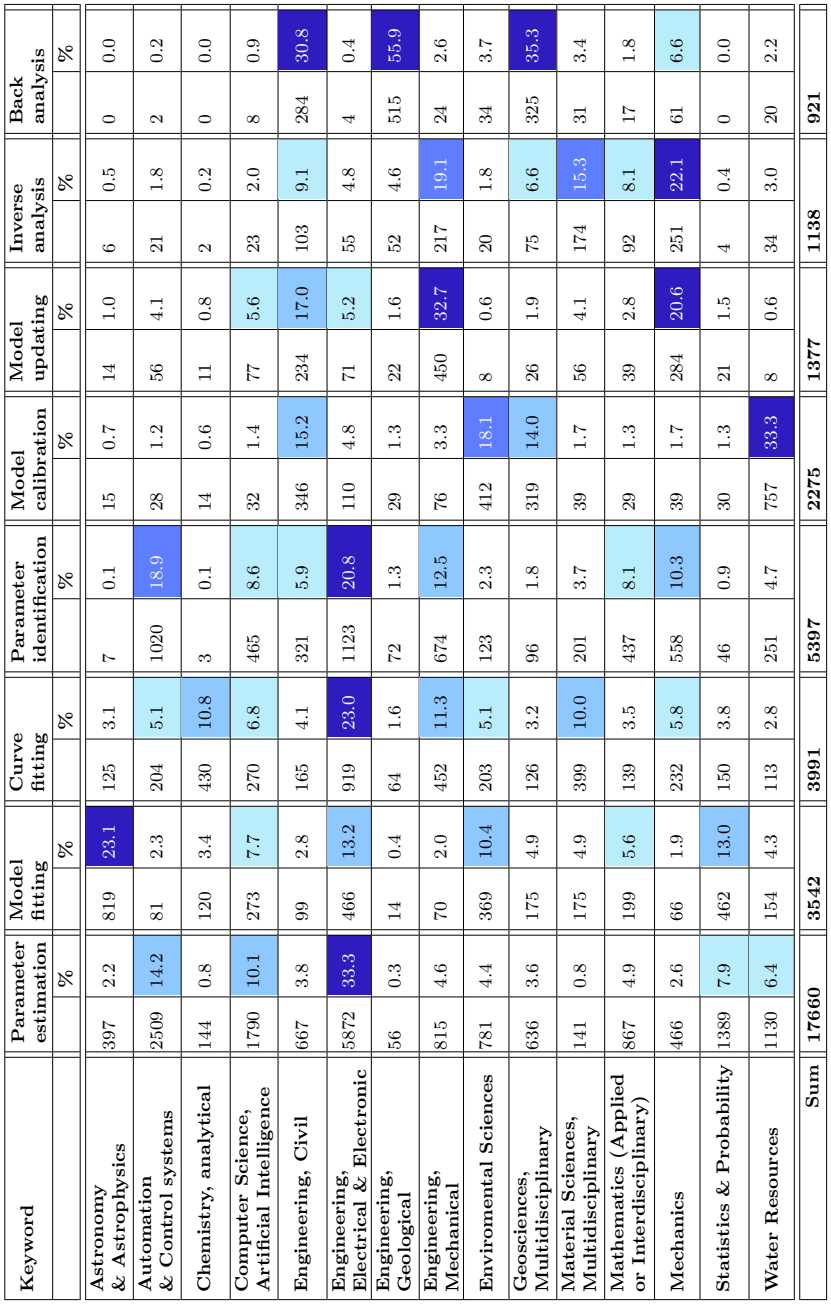
The problem of inverse analysis occurs in many engineering tasks and, as such, attains several different forms and can be solved by many very distinct methods. Note that the topic addressed in this work can be found under totally different terms, most frequent ones are *Parameter estimation*, *Model fitting*, *Curve fitting*, *Parameter identification*, *Model calibration*, *Model updating*, *Inverse analysis* and *Back analysis*, see Fig. 3.1 for the list of their appearance within individual scientific disciplines. In this chapter, an overview of two basic philosophies of the parameter estimation with an emphasis put on the area of soft-computing methods is presented. An interested reader is referred to [61] and PhD Theses [50] and [98] for more details and description of other approaches. An application on the fitting of one phenomenological model will be shown in the end.

3.1 Introduction[1]

A variety of engineering tasks nowadays leads to an inverse analysis problem. Generally, the aim of an inverse analysis is to rediscover unknown inputs from the known outputs. In common engineering applications, a goal is to determine the initial conditions and properties from physical experiments or, equivalently, to find a set of parameters for a numerical model describing the experiment. Therefore, existence of such numerical model is assumed in this work and the task is to find parameters of this model to match outputs from model with results from the experiment.

In overall, there are two main philosophies to solution of this problem.

[1] A part of this chapter is reproduced from: A. Kučerová, M. Lepš, and Z. Bittnar. Solutions to inverse analysis problems using soft-computing methods. In *Proceedings of ICCES'07 (International Conference on Computational & Experimental Engineering and Sciences)*, pages 1531–1537. Forsyth: Tech Science Press, 2007.



Keyword	Parameter estimation		Model fitting		Curve fitting		Parameter identification		Model calibration		Model updating		Inverse analysis		Back analysis	
		%		%		%		%		%		%		%		%
Astronomy & Astrophysics	397	2.2	819	23.1	125	3.1	7	0.1	15	0.7	14	1.0	6	0.5	0	0.0
Automation & Control systems	2509	14.2	81	2.3	204	5.1	1020	18.9	28	1.2	56	4.1	21	1.8	2	0.2
Chemistry, analytical	144	0.8	120	3.4	430	10.8	3	0.1	14	0.6	11	0.8	2	0.2	0	0.0
Computer Science, Artificial Intelligence	1790	10.1	273	7.7	270	6.8	465	8.6	32	1.4	77	5.6	23	2.0	8	0.9
Engineering, Civil	667	3.8	99	2.8	165	4.1	321	5.9	346	15.2	234	17.0	103	9.1	284	30.8
Engineering, Electrical & Electronic	5872	33.3	466	13.2	919	23.0	1123	20.8	110	4.8	71	5.2	55	4.8	4	0.4
Engineering, Geological	56	0.3	14	0.4	64	1.6	72	1.3	29	1.3	22	1.6	52	4.6	515	55.9
Engineering, Mechanical	815	4.6	70	2.0	452	11.3	674	12.5	76	3.3	450	32.7	217	19.1	24	2.6
Enviromental Sciences	781	4.4	369	10.4	203	5.1	123	2.3	412	18.1	8	0.6	20	1.8	34	3.7
Geosciences, Multidisciplinary	636	3.6	175	4.9	126	3.2	96	1.8	319	14.0	26	1.9	75	6.6	325	35.3
Material Sciences, Multidisciplinary	141	0.8	175	4.9	399	10.0	201	3.7	39	1.7	56	4.1	174	15.3	31	3.4
Mathematics (Applied or Interdisciplinary)	867	4.9	199	5.6	139	3.5	437	8.1	29	1.3	39	2.8	92	8.1	17	1.8
Mechanics	466	2.6	66	1.9	232	5.8	558	10.3	39	1.7	284	20.6	251	22.1	61	6.6
Statistics & Probability	1389	7.9	462	13.0	150	3.8	46	0.9	30	1.3	21	1.5	4	0.4	0	0.0
Water Resources	1130	6.4	154	4.3	113	2.8	251	4.7	757	33.3	8	0.6	34	3.0	20	2.2
Sum	**17660**		**3542**		**3991**		**5397**		**2275**		**1377**		**1138**		**921**	

0 - 4.9% | 5 - 9.9% | 10 - 14.9% | 15 - 20% | 20% and more

Figure 3.1: Frequency of the equivalent terms for parameters estimation among scientific branches according to WoS (last update 2009)

A forward (classical) mode / direction is based on the definition of an error function of the difference between outputs of the model and experimental measurements. A solution comes with the minimum of this function. The second philosophy, *an inverse mode*, assumes existence of an inverse relationship between outputs and inputs. If such relationship is created, then, the retrieval of desired inputs is a matter of seconds yet repeatedly. Both philosophies are introduced in the text and thoroughly discussed.

3.2 Forward mode of inverse analysis

As mentioned previously, the problem of an inverse analysis can be formulated based on the existence of an experiment E, which, physically or virtually, connects the known inputs (parameters) X^E to the desired outputs (measurements) Y^E. Formally, this can be written as

$$Y^E = E(X^E). \tag{3.1}$$

Then, the problem of an inverse analysis is defined as a search for unknown inputs X^E from the known outputs Y^E, i.e. *inversely* to the experiment E. In common engineering applications, the experiment E is usually simulated by some virtual model M. Often, the model is a program based on numerical methods such as the finite element method. Here, this work assumes that the model M is sufficiently precise to replace the experiment E, and thus, we can put $E \equiv M$. This automatically results to $Y^E = M(X^E)$. This step is important from the economy point of view, where the cost of the evaluation of the model M is assumed to be by an order of magnitude smaller than the cost of the physical experiment E.

Based on the above-mentioned statements, *the forward* (classical) mode / direction of an inverse analysis is defined as a minimization of an error function of a difference between the outputs of the model and the output of the experiment, i.e.

$$min\ f(X) = ||Y^E - M(X)||. \tag{3.2}$$

A solution X^* comes with the minimum of this function, where $f\ (X^*) = 0$ as well as $X^* \equiv X^E$.

The problem (3.2) has been classically solved by gradient-based optimization methods. Nowadays, the model M is usually hidden in a program which is limited by license conditions, compact code etc. and therefore, the knowledge of derivatives is missing even if the function is differentiable. Hence, the soft-computing methods can be successfully applied here. Methods like *the simulated annealing method* [42] with one solution in time or *evolutionary algorithms* [63] with a "population" of solutions are usually used.

The main advantage of this approach is that the forward mode is general in all possible aspects and is able to find an appropriate solution if such exists. This statement is confirmed with special cases like

a) A problem of a same value of outputs (Y) for different inputs (X), i.e. existence of several global optima. This case leads to a multi-modal optimization [59] but is solvable by an appropriate modification of an optimization algorithm.

b) There are different outputs (Y) for one input (X). This is the case of stochastic and probability calculations as well as experiments burdened with a noise or an experimental error. This obstacle can be tackled e.g. by introduction of stochastic parameters for outputs.

c) There is more than one experiment for one material. This task can be handled as a multi-objective optimization problem, see e.g. [16],[64] for the references on multi-objective optimization methods and Thesis [98] for multi-objective identification solutions.

3.3 Reducing disadvantages of forward mode

The biggest disadvantage of the forward mode is the need for a huge number of error function evaluations. This problem can be managed by two approaches: the first one is based on parallel decomposition and parallel implementation, the second one employs a computationally inexpensive approximation or interpolation method.

The parallel decomposition is based on an idea of the so-called *implicit parallelism*, i.e. the independence of any two solutions X. This can be utilized by *master-slave model* [10], where the main (master, root) processor/computer controls the optimization process while the slave processors compute the expensive evaluations of the model M. Thanks to independency of solutions, nearly linear speed-up can be reached until a high number of processors [56].

The second methodology applies similar idea used previously. It assumes another model $\underline{M}$ different from the model M but with a lower computational cost than M. Then, a goal is to use the cheap model $\underline{M}$ instead of M as often as possible. The equality is not fulfilled here, i.e. $\underline{M} \neq M$ and therefore, the optimum $\underline{X^*}$ obtained by optimizing the model $\underline{M}$ has to be always checked by equality $Y^E = M(\underline{X^*})$. A question arises, how to construct the model $\underline{M}$. Two, most often used, approaches will be inspected. First methodology, let's call it an *interpolation approach*, is created to interpolate the model M without any knowledge of an inner structure of the model M. As a solution, traditional mathematical interpolation methods like *Kriging* or *response surface methods* (RSM) [53] or soft-computing methods like *radial basis function networks* (RBFN) [46],[70] are used. The need for interpolation is crucial here, because new model $\underline{M}$, except for already computed values $\underline{Y} \equiv Y$, does not correspond to the model M and therefore the model $\underline{M}$ is assumed to be unreliable. To obtain a sufficiently precise interpolation of the model M, an iterative process is usually used: starting with $\underline{M_k}$ from known values (pairs X_k,Y_k), the minimum $f\ (\underline{X^*}) = ||Y^E - \underline{M_k}(\underline{X^*})||$ is found using multimodal optimization (this step should by computationally inexpensive due to usage of the cheap model $\underline{M}$), correct values of the model M are computed by $Y_{k+1} = M(\underline{X^*})$ and these new values are added to the set of already computed pairs. This procedure is repeated until the minimum of the function (3.2) is reached. A subscript k is used to describe a number of iterations. Finally, the problem is to minimize a number of evaluations of the expensive model M. Individual iterative procedures proposed by different authors, see e.g. [70],[77] or [51], differ in the details of how this problem is solved. Also note that the $\underline{M}$ model can be applied within the optimization process in two ways: (i) as an approximation of the response of the model M and then, the optimization

is running on the model $\underline{M}$, or (ii) as an pre-evaluation of the model M, as is presented within the area of *Meta-model assisted evolutionary algorithms*, see e.g. [25]. Our experiences with the last approach are not positive, the optimal results were strongly problem dependent.

The second approach, here called *approximation*, assumes that, for a physical model M, there is an inaccurate physical model $\underline{M}$, which is computationally less expensive than the model M. This situation occurs in cases where, for one physical phenomenon, there are two or more describing theories, e.g. wave vs. particle theories. More often, there are cases, where different topologies, geometries, a different number of finite elements, a simple or a difficult model, a 2D or a spatial model etc. for a studied problem can be used. The biggest problem here is that for such models, the inequality $\underline{M} \neq M$ will be almost valid. Moreover, it is not clear how to include this discrepancy into the simple model $\underline{M}$ (contrary to the interpolation approach). There is few papers published to address this topic and therefore, we refer an interested reader to available sources [33],[103] or [79].

The second disadvantage of the forward mode, following the definition, is a fact that the computationally expensive search should be repeated for any change in data, e.g. even for small change in an experimental setup. This feature handicaps the forward mode from an automatic and frequent usage. The opposite is true for the second mode of an inverse analysis.

3.4 Inverse mode of inverse analysis

The second philosophy, *an inverse mode*, assumes existence of an inverse relationship between outputs and inputs, i.e. there is an "inverse" model M^{INV} associated to the model M, which fulfils following equation:

$$X = M^{INV}(Y) \tag{3.3}$$

for all possible Y. Generally, this inverse model does not need exist. Nevertheless, we assume that the inverse model can be found sufficiently precise on some closed subset of the definition domain. Next, we will limit our attention to an approximation of an inverse relationship, not its exact description.

A quality of this approximation is easy to measure since a pair X, Y obtained using Equation (3.3) should also fulfill the original Equation (3.1). Final usage of this methodology is trivial because a desired value X^E can be obtained by simple insertion Y^E into Equation (3.3).

The main advantage is clear. If an inverse relationship is created, then the retrieval of desired inputs is a matter of seconds yet repeatedly. This can be utilized for frequent identification of one model. On the contrary, the main disadvantage is an exhausting search for the inverse relationship. Further obstacles are the existence problems for the whole search domain and inability to solve the a) case mentioned above in Section 3.2. Case b) can be handled by introducing stochastic parameters, case c) by sequential, hierarchical or iterative processes [104, 52]. As a solution, different approximation tools are applied. Nowadays, artificial neural networks (ANN) [35],[9] are commonly used due to their ability to approximate complex non-linear functions and their straightforward implementation and utilization.

Recently, authors of this contribution have published an example of the inverse mode applied to the above-mentioned microplane model parameter identification [52]. Within concrete modeling community, two research groups have implemented inverse mode using stochastic parameters of the searched parameters, see [26] and [91] for more details.

3.5 Identification of parameters for affinity model

For the purposes of Chapter 5, an example of the forward mode of the parameters estimation is presented here. The goal is to find B_1, B_2 and η coefficients for an affinity model mentioned in Section 1.2.2. The optimization is controlled by an evolutionary algorithm GRADE [50], see the reference for more details. Since the affinity model is relatively cheap from the computational point of view, the maximum number of evaluations has been set to 100.000 and the overall time spent was less than one hour. The objective function to be maximized is $\mathbf{R^2}$, a *coefficient of determination*[2]. Here, $\mathbf{R^2}$ provides a measure of how well future outcomes are likely to be predicted by

[2] `http://en.wikipedia.org/wiki/Coefficient_of_determination`

the model. A data set has values y_i each of which has an associated modeled value f_i. Here, the values y_i are called the observed values and the modeled values f_i are sometimes called the predicted values. The "variability" of the data set is measured through different sums of squares:

$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$**,** the total sum of squares (proportional to the sample variance);

$SS_{\text{reg}} = \sum_i (f_i - \bar{f})^2$**,** the regression sum of squares, also called the explained sum of squares.

$SS_{\text{err}} = \sum_i (y_i - f_i)^2$, the sum of squared errors, also called the residual sum of squares.

In the above, $\bar{y}$ and $\bar{f}$ are the means of the observed data and modeled (predicted) values, respectively. That is: $\bar{y} = \frac{1}{n}\sum_i^n y_i$ and $\bar{f} = \frac{1}{n}\sum_i^n f_i$, where n is the number of observations. Then, the most general definition of the coefficient of determination is

$$\mathbf{R^2} \equiv 1 - \frac{SS_{\text{err}}}{SS_{\text{tot}}}. \tag{3.4}$$

$\mathbf{R^2}$ is a statistic that will give some information about the goodness of fit of a model. In regression, the $\mathbf{R^2}$ coefficient of determination is a statistical measure of how well the regression line approximates the real data points. An $\mathbf{R^2}$ of 1.0 indicates that the regression line perfectly fits the data. Note that $\mathbf{R^2}$ can be even negative. This is the case where $SS_{\text{err}} > SS_{\text{tot}}$.

	B_1	B_2	η	$\mathbf{R^2}$	**Fig.**
Original setting	1.0e+7	2.0e-4	-7.6	0.9932	3.2
Fitted coefficients	8.543e+6	1.198e-3	-6.839	0.9972	

Table 3.1: Fitted coefficients of an affinity model for all data sets

The result show a slight improvement in both, the $\mathbf{R^2}$ measure as well as produced graphs, see Tab. 3.1 and Fig. 3.2. The reason is that there is no

enough degrees of freedom to bend one affinity model to all data samples. Note that optimizing all curves individually, i.e. searching for a unique set of parameters for every data set, leads to perfect fits (not shown here).

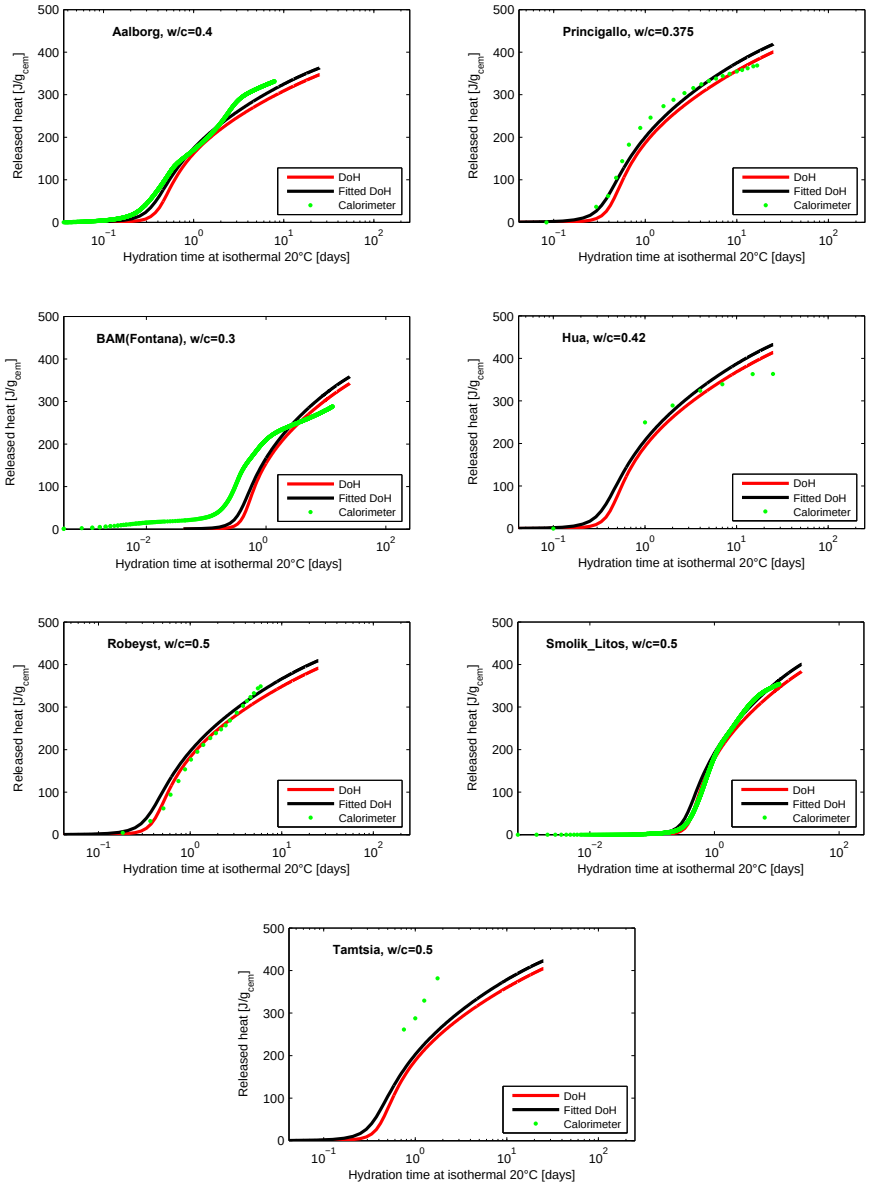


Figure 3.2: Fitted affinity model for all data sets. Key: DoH - original setting, Fitted DoH - simulations for new parameters

3.6 Conclusions

The two most often used approaches how to solve the problem of parameters estimation were presented here. Both methodologies are described and thoroughly discussed. As an illustrative example, the forward mode is used to find parameters for the affinity model of hydration that will be used later in this work. If the studied model is simple and fast, then the traditional forward approach is the most logical solution. More importantly, any of the new modern zero-order optimization methods like Evolutionary Algorithms [24, 23] can be used.

Chapter 4

KRIGING APPROXIMATION

This chapter presents experiences and difficulties encountered during interpolation of experimental results by Kriging/DACE metamodel [45]. Since the response of the cement paste studied in this work in terms of mixture parameters is non-linear, the Kriging approximation in the space of hydration heat of available real measurements seems a natural choice. Particularly, several combinations of regression and correlation parts have been tested and optimized in order to ensure monotonicity of the response. Even though a certain progress is reported, the selection of the proper model still remains a challenging task.

4.1 Introduction[1]

Cement paste is a fundamental scale from which concrete inherits majority of its properties. Experimental results show considerable scatter in the elastic response of cement paste samples; however, virtual testing in a computer allows testing the influence of input parameters on resulting macroscopic response, see the introductory sections or a reference [99]. Recently, a combination of a CEMHYD3D model with homogenization processes was tested as a basis for an optimization [102], where the Young modulus and heat of hydration appear as objective functions. Question arises, whether results from the optimization of the virtual model can be trusted. Our proposed solution is based on a so-called robust optimization [34] where some selected distance to existing experimental results is employed as the robustness measure, see Chapter 6. Hence, our goal is to create the closest approximation to available experimental data and to provide estimation of the quality of that

[1]Partially reproduced from: M. Valtrová and M. Lepš. Kriging approximation in cement paste experimental performance. In *Engineering Mechanics 2009*, pages 1–7. Institute of Theoretical and Applied Mechanics AV ČR, 2009.

approximation.

In this chapter, we demonstrate that actually the popular Kriging, also called DACE (acronym for Design and Analysis of Computer Experiments [82]), approximation is far away from the surface that is expected to describe physical process underneath. Therefore, nonlinear optimization of the maximum monotonicity is presented. Overall, dozens of combinations of regression and correlation parts have been tested. Unfortunately, the importance of a proper regression part is more crucial than presented in the optimization literature [45]. We have found a regression part that almost ideally describes the physical problem; however, the strict monotonicity has not been preserved.

Name	$R(\theta; d_j)$,	$d_j = w_j - w_i$
EXP	$exp(-\theta_j \lvert d_j \rvert)$	
GAUSS	$exp(-\theta_j d_j^2)$	
LIN	$max\{0, 1 - \theta_j \lvert d_j \rvert\}$	
SPHERICAL	$1 - 1.5\xi_j + 0.5\xi_j^3$,	$\xi_j = min\{1, \theta_j \lvert d_j \rvert\}$
SPLINE	$1 - 15\xi_j + 30\xi_j^3$,	$for \quad 0 \leq \xi_j \leq 0.2$
	$1.25(1 - \xi_j)^3$,	$for \quad 0.2 \leq \xi_j \leq 1$
	0,	$for \quad \xi_j > 1, \quad \xi_j = \theta_j \lvert d_j \rvert$

Table 4.1: Correlation functions

4.2 Kriging

Kriging is an approximation method frequently used in geostatistics, global optimization and statistics [96]. Kriging was originally developed by the South African mining engineer D.G. Krige in the early fifties. In the 1960s the French mathematician G. Matheron gave theoretical foundations to this method [62] and named the method after Krige. Generally, the Kriging predictor is composed of a regression and interpolation part that constitutes the nonlinear surface among available data [57]:

$$\hat{y} = f(x)^T \beta^* + r(x)^T \gamma^* \quad , \tag{4.1}$$

where $f(x)$ is an *a-priori* selected set of basis functions creating the response

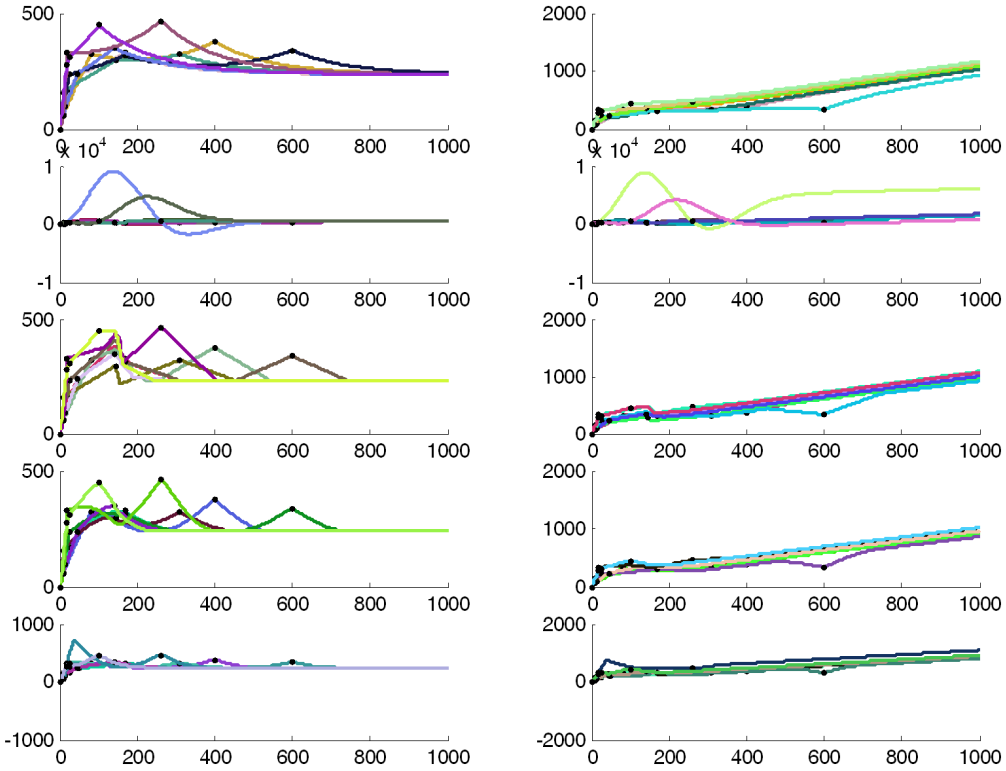


Figure 4.1: Cuts of approximations for **not optimized weights** θ: Constant regression term (left column), linear regression term (right column) and (from top) five correlation functions

surface and $r(x)$ is the correlation term between an unsampled point x and known points s_i, $i = 1, ..., m$: $r(x) = [R(\theta; s_1; x), ..., R(\theta; s_m; x)]^T$, where R is *a-priori* selected correlation function with unknown coefficients θ, see later. The regression part is solved by a generalized least squares solution

$$\beta^* = (\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{R}^{-1}\mathbf{Y} \quad , \tag{4.2}$$

where $\mathbf{F}$ is a matrix containing $f(x)$ evaluated at known sites s_i, $\mathbf{R}$ stems for correlation among s_i using again the correlation function R and $\mathbf{Y}$are known values of y_i at s_i. The Kriging part then interpolates the residual leading to the system of linear equations

$$\mathbf{R}\gamma^* = \mathbf{Y} - \mathbf{F}\beta^* \quad . \tag{4.3}$$

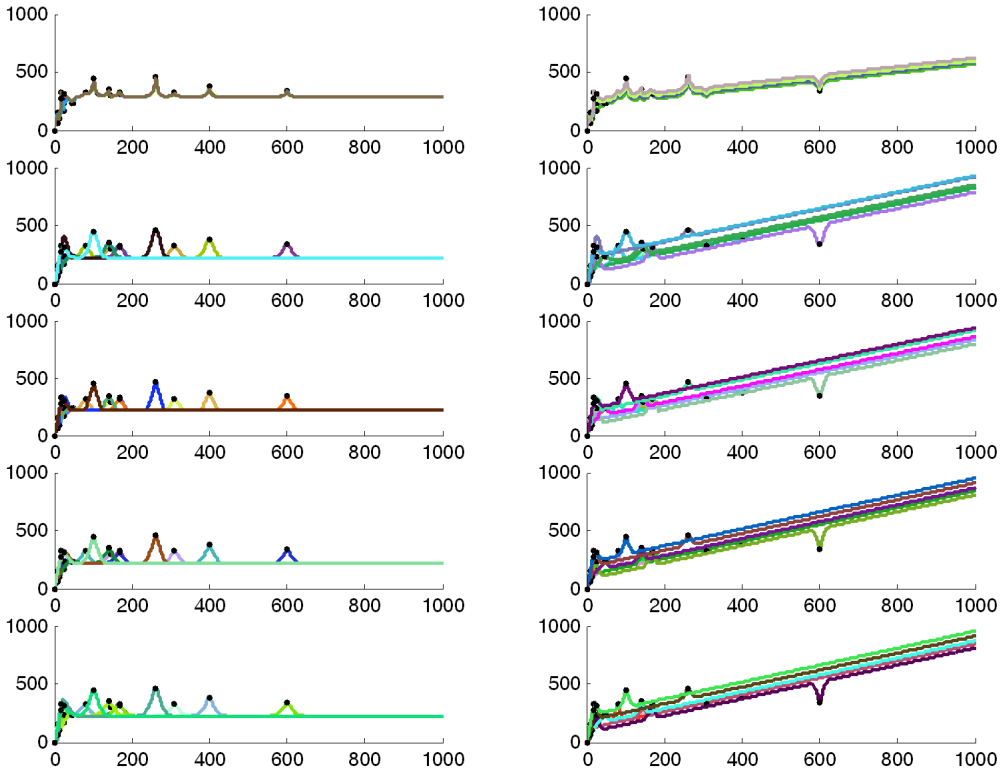


Figure 4.2: Cuts of approximations for **optimized weights θ for minimal MSE**: Constant regression term (left column), linear regression term (right column) and (from top) five correlation functions

The use of such a metamodel for optimization purposes is less demanding on the regression part since an interpolation is dominant and hence, the constant regression part usually suffices. Then, the correlation function is traditionally selected to obtain a positive-definite system of equations, mainly restricted to the form

$$R(\theta, w, x) = \prod_{j=1}^{n} R_j(\theta, w_j - w_i) \quad . \tag{4.4}$$

In our case, a free Matlab toolbox DACE [57] is utilized providing seven correlation functions, where five of them are presented in this work, see Tab. 4.1.

Note that at this point we still do not know the tuning/shape param-

eters θ. Their functionality is twofold: they express the anisotropy among dimensions and also determine the shape of the metamodel in the vicinity of given samples. Traditionally, these parameters are found *a-posteriori* by minimizing an expected mean squared error (MSE), which leads to the constrained nonlinear optimization problem. See e.g. [45] for discussion on how to efficiently solve this problem without re-calculation of β^* and γ^* for these new θ.

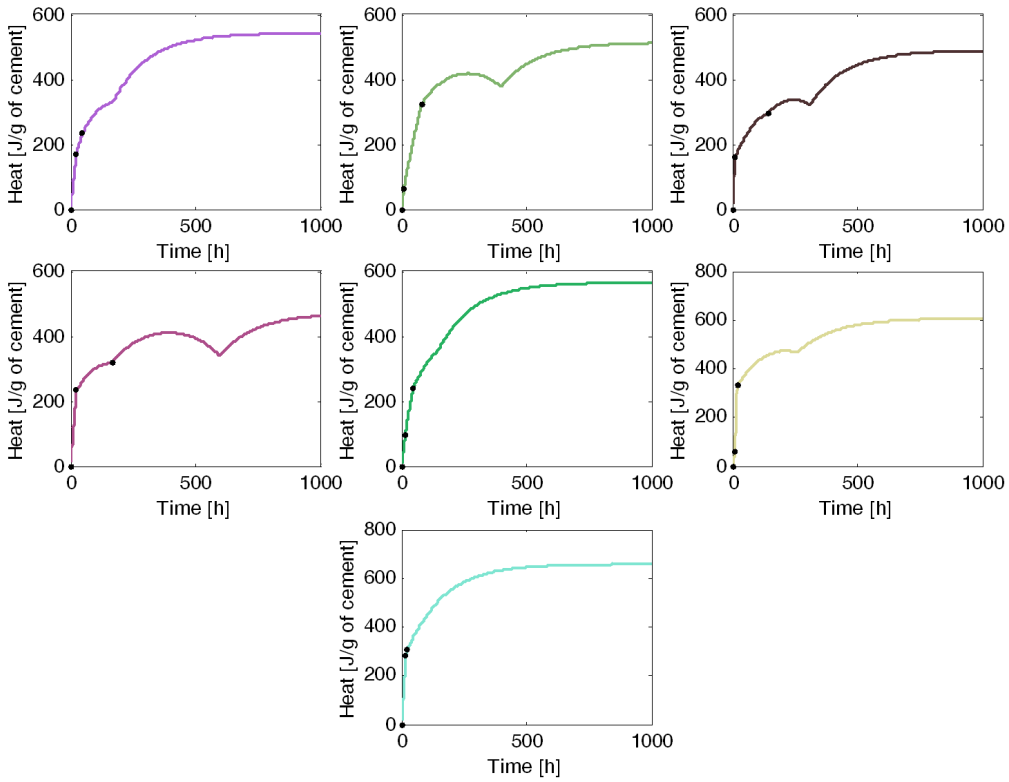


Figure 4.3: Cut of approximation through experiments using exponential correlation function, linear term of composition and exponential regression term in time for **not optimized weights** θ

4.3 Fitting of experimental data

Particular application is shown on experimental data describing the released heat of cement pastes already presented in Tab. 1.2 and Tab. 1.3. First,

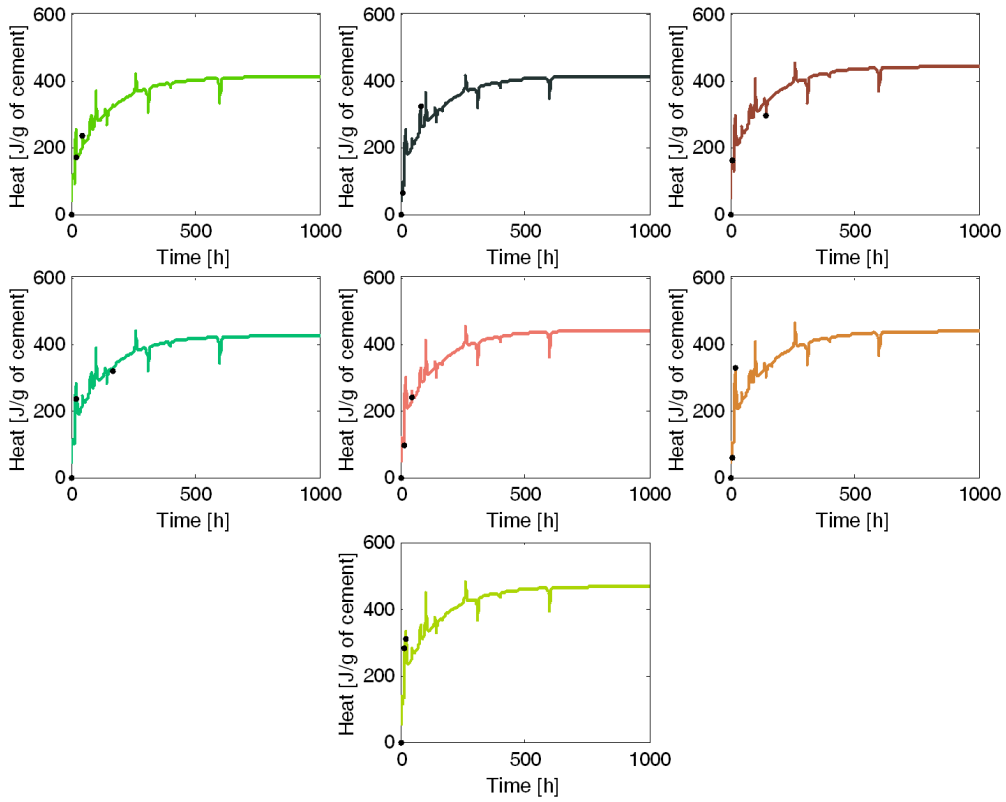


Figure 4.4: Cut of approximation through experiments using exponential correlation function, linear term of composition and exponential regression term in time for **optimized weights θ for minimal MSE**

almost linear dependency within the input data caused by volume unity of the mixture experiment described in Chapter 2 results in ill-conditioned matrices in Eq. (4.2). This obstacle is solved by Principal Component Analysis (PCA) by transforming inputs into the space of principal directions and removing the direction with the smallest eigenvalue, see e.g. [50] for more details. Therefore, our approximation is a real function (hydration heat) of seven inputs – time plus seven original inputs from Tab. 1.2 transformed with PCA to the six dimensions. Next, several combinations of regression and correlation functions have been tested, see Fig. 4.1 and Fig. 4.2. Horizontal axes are for time and vertical axes for hydration heat. Note that zero point [0 h, 0 J/g] has been added to enforce a physically reasonable start of the heat-time relationship.

There are two main requirements on the approximation. We need an interpolation of experimental data to precisely describe the behavior in the vicinity of existing experiments and conversely, the best possible description of the trend in extrapolation. This is of great importance since there is a low number of available data and the range of parameters covered is usually also small. The deficiency of created metamodels for extrapolation purposes is clearly visible from Fig. 4.1 and Fig. 4.2. Whenever the metamodel is far away from given data, the prediction is approaching the mean trend. This means that in distant extrapolation we would obtain a flat surface in the case of a constant regression term and a linear surface in a linear case.

We have tried a dozen combinations of regression descriptions and correlation functions and finally, a combination of an exponential correlation function, a linear regression of mixture parameters and an exponential regression term $(1 - e^{-T})$ for time T gives reasonable regression output, see Fig. 4.3. In Fig. 4.4, the result for optimized weights θ with respect to the minimal MSE is presented. Since the curve of hydration heat history should be (from physical principles) monotonous, the traditional MSE minimization is replaced by minimization of a negative (numerical) derivative of a resulting curve in the time direction. As an optimization algorithm, the Quasi-Newton line-search method available in Matlab Optimization toolbox was used. The optimization algorithm ran 6.5 minutes on AMD Turion MT-37 notebook processor with more than 700 evaluations of the metamodel. The resulting curves are presented in Fig. 4.5. The approximation that almost ideally describes the physical problem has been found, however, the strict monotonicity has not been preserved, see again Fig. 4.5.

4.4 Conclusions

The main advantage of the proposed methodology is that the expected mean Kriging prediction also offers an expected mean squared error (MSE) which serves as a good proxy for the distance from the available experimental data, i.e. MSE is zero at given points and is monotonously growing with the distance from the nearest known values, see again Fig. 4.5. On the other hand,

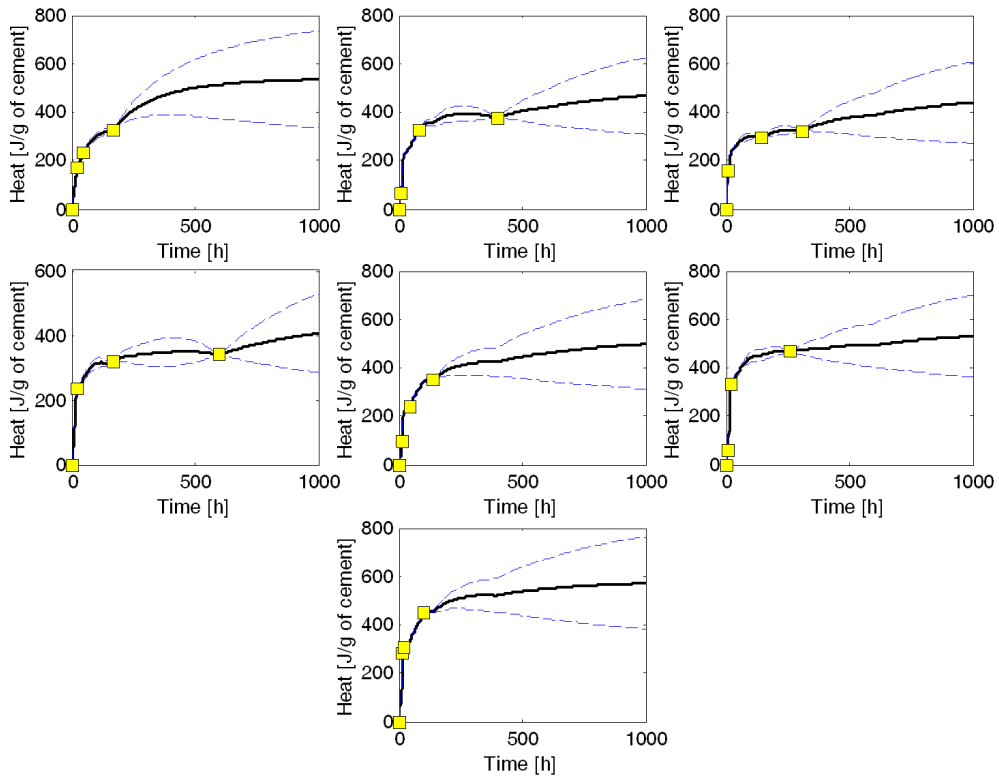


Figure 4.5: Cut of approximation through experiments with expected mean (black continuous line) and MSE bounds (blue dashed lines) for **optimized weights θ for maximal monotonicity**

the main disadvantage is the exhausting search for proper model functions.

Finally, the search for the best closed-form approximation is still black magic and is part of the know-how of every experienced curve fitter. Next chapter shows that Genetic Programming can be used to solve this task using a burden of a computational power.

Chapter 5

GENETIC PROGRAMMING APPROXIMATION

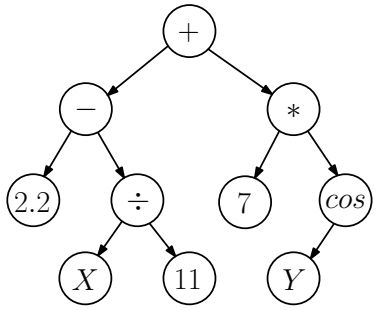
This chapter introduces results of Genetic Programming used for creation of experimental data approximations. Within Genetic Programming trees, the placement of constants still has not been satisfactorily solved. Therefore, the proposed contribution also presents a search for real-valued constants employing Ordinary Least Squares (OLS). Twenty trees as results of twenty independent runs of Genetic Programming are presented. From these results the best six trees are chosen according to specific criteria and the approximations of experimental data are shown. Still, many aspects of Genetic Programming-based symbolic regression are uncovered and especially suppression of the overfitting issues remains unsolved.

5.1 Introduction to Genetic Programming[1]

Genetic Programming (GP) is a relatively new form of artificial intelligence and is inspired by Darwinian biological evolution and genetics. GP is an extension of Genetic Algorithms (GA) [108]. Contrary to GA that uses string of numbers to represent the solution, Genetic Programming deals with tree-structured program (tree) as an individual (see[2] Fig. 5.1). GP searches highly fit computer programs in the space of all possible programs that solve a problem. The trees are compound of nodes that are elements either from a functional set or from a terminal set. Generally, the functional set consists of mathematical operators, for example $\{+, -, *, /\}$ while the terminal set contains variables or constants. The main difference in the functional set and the terminal set is that the terminal set cannot have arguments.

[1]Partially reproduced from: M. Valtrová and M. Lepš. Genetic programming approximation in cement paste experimental performance. In *Engineering Mechanics 2010*, pages 1–7. Institute of Thermomechanics AS CR, Sent for publication, 2010

[2]Reproduced from `http://en.wikipedia.org/wiki/Genetic_programming`



$(2.2 - (\frac{X}{11}) + (7 * cos(Y))$

Figure 5.1: An example of a tree

The evolution of programs is performed through the action of genetic operators and the evaluation of the fitness function [3]. Generally, there are three main genetic operators: reproduction, crossover and mutation. Reproduction is the process of copying individuals according to their fitness value [108]. Trees with fitness lower than the average are killed and the new population is filled with the surviving trees. Meanwhile, the crossover is the process of combining information from two trees that are selected from the whole population. Here, one randomly selected subtree is interchanged with another one and two new offsprings are created (see[3] Fig. 5.2). The last genetic operator is mutation (see again Fig. 5.2). Mutation replaces the subtree of selected individuals with new randomly generated subtrees [108]. These operations with trees are repeated after several generations until the tree with optimum fitness value is obtained. The principle of genetic programming can be seen in Fig. 5.3.

5.1.1 Symbolic regression in Genetic Programming

Problems of symbolic regression require finding a function in a symbolic form that fits a given finite sampling of data points [48]. One of the main challenges in symbolic regression using Genetic Programming is the handling of constants (or real numbers). There are three known possibilities how to solve it. Koza in [48] has expanded the terminal set by adding new special

[3]Reproduced from `http://www.alesdar.org/oldSite/IS/chap6-3.html`

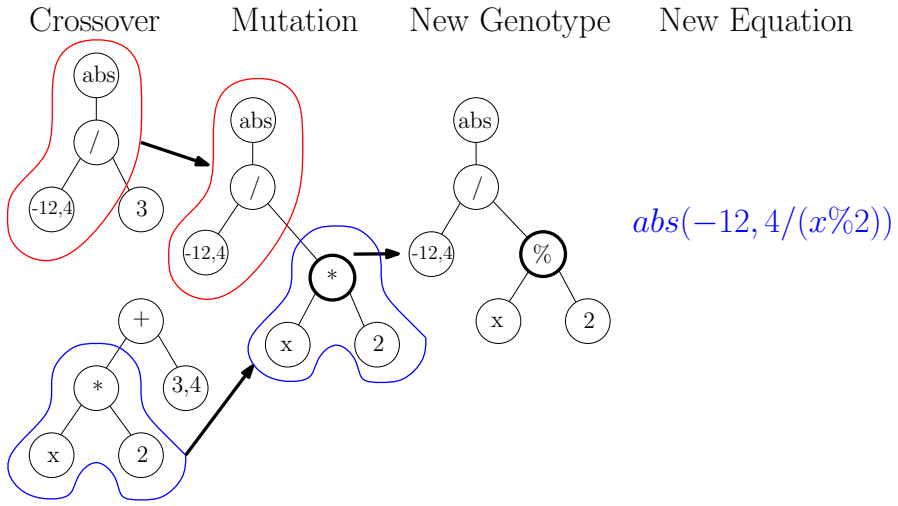


Figure 5.2: Two genetic operators: the crossover and the mutation

terminal called the *ephemeral random constant*. Whenever the ephemeral random constant is chosen for any endpoint of the tree during the creation of the initial random population in generation 0, a random number of a specified data type in a specified range is generated and attached to the tree at that point [48]. These constants remain fixed for all generations.

The second access to solving constants in symbolic regression is presented in paper [3]. The authors used Genetic Programming for the creation of approximation functions obtained by the response surface methodology. In this work the constants are called *tuning parameters*. They are allocated to a subtree depending on the type of the current node and the structure of the subtree according to the algorithm described in [3]. Once the tuning parameters are allocated at different parts of a tree, a nonlinear optimization method is used to compute them.

The last approach of constants' handling has been used, e.g., in [108]. Authors have reported their work regarding GP with polynomials which are applied to fitting a given response surface. They transformed the GP tree into the standard mathematical form by the help of their own translation algorithm. Then the classical regression matrix is created and the Ordinary Least Squares method (OLS) is used to estimate the coefficients. This methodol-

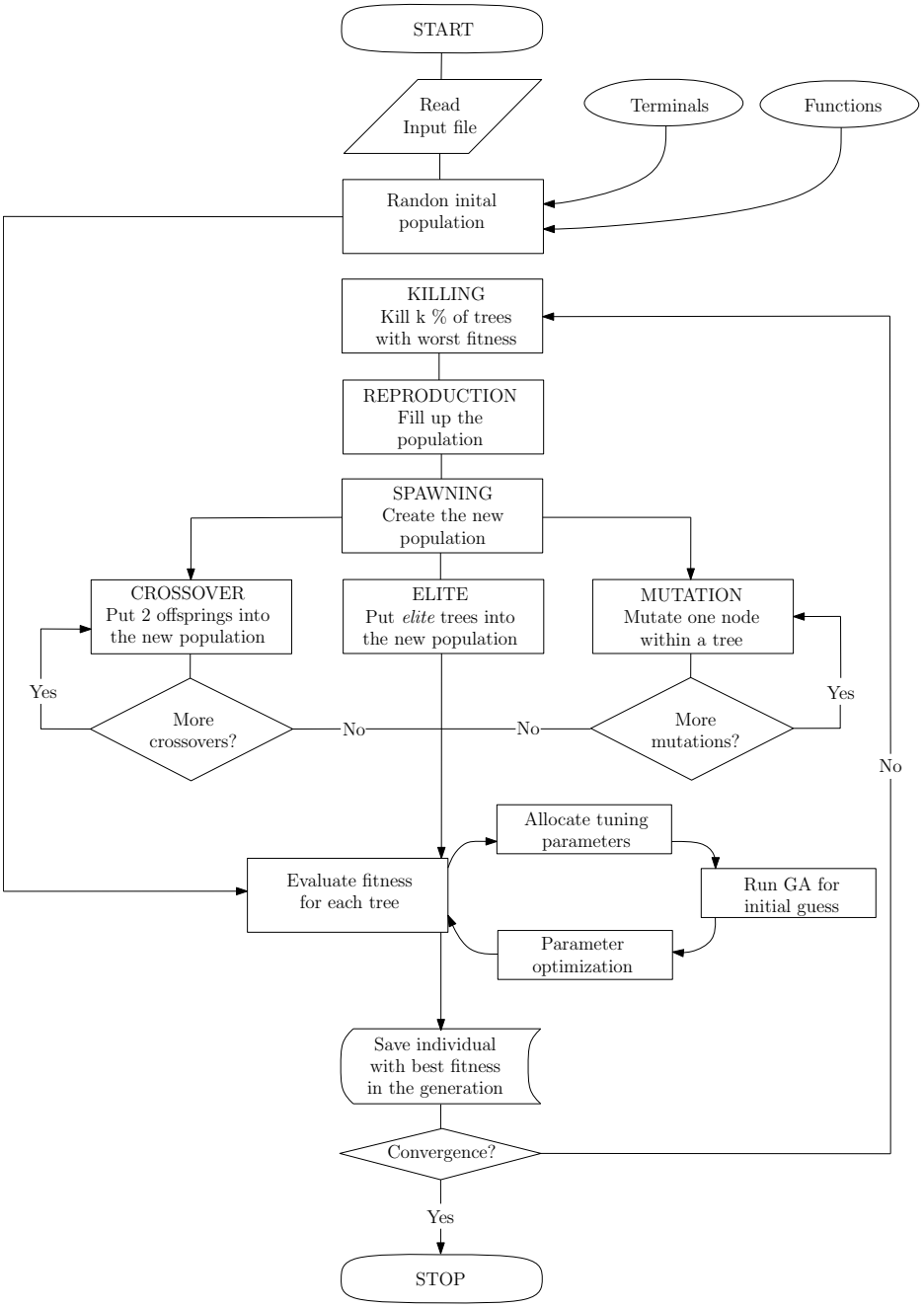


Figure 5.3: The flowchart of GP metodology reproduced from [3]

ogy seems to reduce computational cost in comparison with the nonlinear optimization method.

There are more articles concerning symbolic regression in Genetic Programming. For instance, a paper by [81] presents the analysis of the effect of Multi-Branches Genetic Programming in function approximation problems (i.e. symbolic regression problems). Another article published by [92] describes the usage of Genetic Programming to automate the discovery of numerical approximation formulae. Nevertheless, neither paper specifies the way how they deal with constants within GP trees.

5.2 Application of GP to cement paste hydration data

Because the previous chapter has not brought a sufficiently precise approximation of the hydration heat development, the following part presents Genetic Programming applied at the prediction of hydration heat solely from experimental data sets presented already in Tab. 1.2 and Tab. 1.3. For the sake of simplification, a new notation for seven variables influencing released heat is introduced:

Variable	X1	X2	X3	X4	X5	X6	X7
Parameter	C_3S	C_2S	C_3A	C_4AF	Gypsum	w/c	Fineness

Table 5.1: Definition of the variables X1 - X7

Recall, that according to the affinity model of hydration Eq. (1.7) the value of hydration heat $Q(\cdot)$ is given by

$$Q(\mathcal{X}, t) = Q_{pot}(\mathcal{C}) \cdot DoH(\mathcal{B}(\mathcal{X}), t), \tag{5.1}$$

where the potential hydration heat Q_{pot} is uniquely given by the composition of clinker minerals $\mathcal{C} = \{X1, \ldots, X4\}$ and the time dependent degree of hydration DoH is a phenomenological model influenced by all relevant parameters $\mathcal{X} = \{X1, \ldots, X7\}$ listed in Tab. 5.1. Since particular parameters of the affinity model $\mathcal{B}$ for arbitrary cement are unknown, we have tried to propose a new model based on a GP approximation. The idea is to use an affinity model from Section 3.5 that has been fitted to all available data as an approximation of the DoH, i.e. a time dependent approximation $\overline{DoH}$ which

is independent on the given composition. This ensures the physically correct monotonic behavior of hydration heat $Q(\cdot)$ that has not been obtained in the previous chapter. Then, a general expression of Q_{pot} dependent on all input parameters is needed, i.e. we search for $\overline{Q_{pot}}$ such that approximation

$$Q(\mathcal{X}, t) = \overline{Q_{pot}}(\mathcal{X}) \cdot \overline{DoH}(t) \tag{5.2}$$

fits the given data as much as possible.

5.2.1 Application of GP

We have used a free GPLAB - a Genetic Programming Toolbox for MATLAB by Sara Silva (for more details see `http://gplab.sourceforge.net/`). Since the GPLAB toolbox does not include a solution of real numbers in a symbolic regression within GP, the first step was to solve this problem. In Section 5.1.1 we have presented three possibilities how to treat real numbers. After considering all circumstances (e.g. the way in which the GPLAB is programmed), we decided to transform the GP tree into the standard mathematical form and to apply an Ordinary Least Squares approach. Note that after the last step a GP solution is actually a polynomial with real-valued coefficients that is used for evaluating the fitness of the corresponding tree.

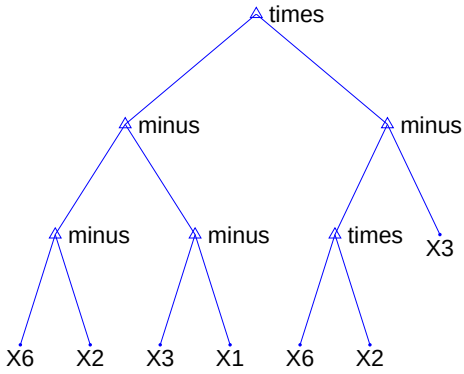
The GPLAB offers many settings of Genetic Programming. Tab. 5.2 shows available options from which several combinations have been tested. In the same table, the final setting that has produced the best performance in short runs is presented. Individual terms are explained in the GPLAB Manual [87].

Name	Available Options	Chosen Setting
Genetic Operators	Crossover Mutation Shrink Mutation Swap Mutation	Crossover Mutation
Initialization Methods	Full Grow Ramped Half-and-Half	Ramped Half-and-Half
Expected Number of Offspring	Absolute Rank85 Rank89	Rank85
Sampling Methods	Roulette SUS Tournament Lexictour Doubletour	Lexictour
Elitism	Replace Keep Best Half Elitism Total Elitism	Replace
Survival of The Individuals	Fixed Popsize Resources Pivotfixe	Fixed Popsize

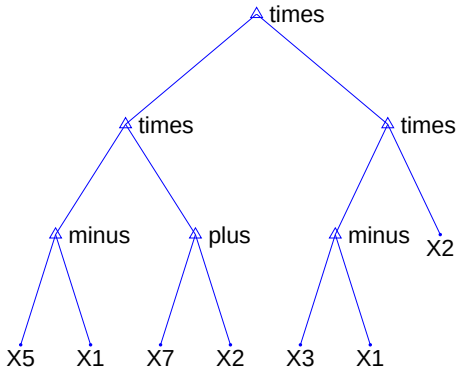
Table 5.2: Available options of Genetic Programming and chosen setting

#	**Tree**	R^2
1	$[X5 * X5 \quad -X1 * X5 \quad X1 * X3 \quad -X3 * X5]$	0.833
2	$[X6 * X6 * X5 \quad -X6 * X5 * X5 \quad X6 * X6 \quad -X6 * X5 \quad X3 \quad X7]$	0.854
3	$[X7^2 * X3 * X4 \quad -X7^2 * X3 * X6 \quad -X7^3 * X4 \quad X7^3 * X6 \quad -X7^2 * X5 * X4 \quad X7^2 * X6 * X5]$	0.567
4	$[3 * X5 * X4X3 * X4 \quad -X7 * X4 \quad -3 * X6 * X5 \quad -X3 * X6 \quad X7 * X6]$	0.813
5	$[X5^2 \quad X6 \quad X4 \quad X5 \quad -X3 \quad X7]$	0.892
6	$[X5 \quad X7 \quad -X3 \quad -X5 * X4 \quad -X5 * X6 \quad -X6]$	0.889
7	$[X6 \quad X4^2 \quad 2 * X4 * X6 \quad X6^2 \quad -X3^2]$	0.890
8	$[X6 \quad X3 \quad -X5 \quad X7 \quad -X5^2 \quad X4]$	0.892
9	$[X7 * X5 \quad X7 * X3 \quad -X3 * X1 * X5 \quad -X3^2 * X1 \quad -X4 * X5 \quad -X4 * X3]$	0.881
10	$[X6^2 * X2 \quad -X6 * X3 \quad -X6 * X2^2 \quad X2 * X3 \quad -X3 * X6 * X2 \quad X3^2 \quad X1 * X6 * X2 \quad -X1 * X3]$	0.895
11	$[2 * X5 * X7^2 \quad X7 * X5^2 \quad -X1 * X7^2 \quad -X7 * X1 * X5 \quad X7^3 \quad -X2 * X7^2 \quad -X7 * X2 * X5]$	0.853
12	$[X6 * X2 \quad X6 * X5 \quad -X4 * X2 \quad -X4 * X5 \quad -X2 * X7 \quad -X5 * X7 \quad X3]$	0.822
13	$[X1 * X7^2 * X4 \quad X1 * X7 * X4 \quad -X1 * X2 * X7 * X4 \quad -X1 * X2 * X4 \quad -X7^3 * X4 \quad -X7^2 * X4 \quad X2 * X7^2 * X4 \quad X2 * X7 * X4]$	0.327
14	$[X2 * X5 * X7 * X3 \quad -X2 * X7 * X1 * X5 \quad X2^2 * X5 * X3 \quad -X2^2 * X5 * X1 \quad -X2 * X1 * X7 * X3 \quad X2 * X1^2 * X7 \quad -X1 * X2^2 * X3 \quad X1^2 * X2^2]$	0.895
15	$[X1 * X7 \quad X7 * X6 \quad X5 * X7 \quad X7 * X4 \quad X1 * X5 \quad X6 * X5 \quad X5^2 \quad X4 * X5]$	0.895
16	$[X7 * X6 \quad X6 * X4 \quad -X6 * X5 \quad -X4^2 \quad X4 * X5 \quad X7^2 \quad -X5 * X7]$	0.801
17	$[X3^2 \quad X3 * X6 \quad -X3 * X7 \quad -X3 * X1 \quad X5 * X3 \quad X6 * X5 \quad -X5 * X7 \quad -X1 * X5]$	0.895
18	$[2 * X6^2 * X4 \quad X6 * X4 * X7 \quad -2 * X6^2 * X5 \quad -X6 * X5 * X7 \quad -2 * X3 * X4 * X6 \quad -X3 * X4 * X7 \quad 2 * X5 * X3 * X6 \quad X5 * X7 * X3]$	0.895
19	$[X7 * X6^2 * X1 \quad X7 * X6^3 \quad X6^2 * X5 * X1 \quad X6^3 * X5 \quad -X6 * X1 * X7^2 \quad -X7^2 * X6^2 \quad -X6 * X7 * X1 * X5 \quad -X6^2 * X5 * X7]$	0.895
20	$[X5 * X7 * X6 \quad -X5^2 * X7 \quad X5 * X1 * X6 \quad -X5^2 * X1 \quad X4 * X7 * X6 \quad -X4 * X7 * X5 \quad X4 * X1 * X6 \quad -X4 * X1 * X5]$	0.895

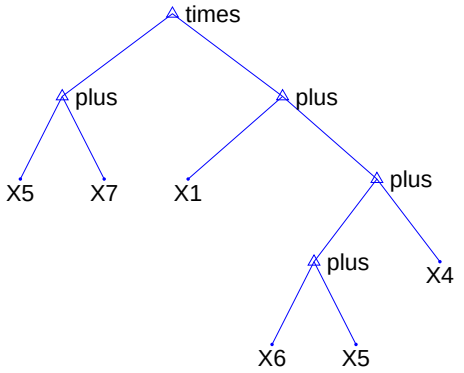
Table 5.3: Best trees out of 20 independent runs (without regression coefficients) with appropriate R^2 coefficients ($R^2 = 1$ means perfect fit, see Section 3.5)



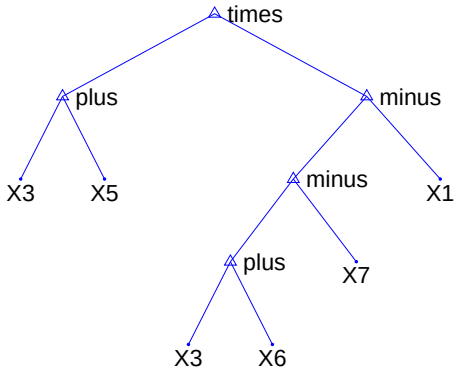
(a) Tree 10



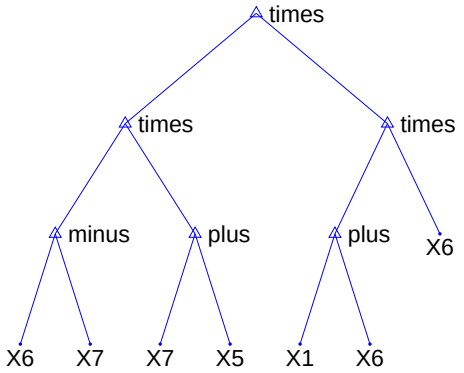
(b) Tree 14



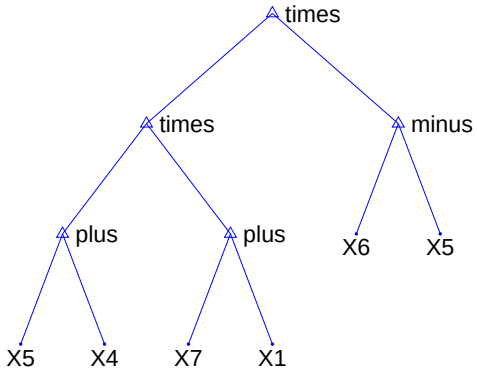
(c) Tree 15



(d) Tree 17



(e) Tree 19



(f) Tree 20

Figure 5.4: Chosen trees

The function set contained mathematical operators $\{+, -, *\}$, the terminal set consisted of variables only. To avoid a bloat, a phenomenon consisting of an excessive code growth without the corresponding improvement in fitness [87], the maximum of 14 nodes was set. Finally, the population size of 500 individuals and the maximum number of generation of 100 was used.

5.2.2 Results

One optimization run was repeated 20 times for the sake of statistical relevance. In the last ten runs we have increased the maximum number of generation to 150. Tab. 5.3 presents best trees (without regression coefficients) along with appropriate coefficients of determination R^2 (the more coefficient of determination is closer to one the better a model is, see Section 3.5).

Table 5.4: Coefficients of chosen trees

Tree Number	Figure	Coefficients
10	Fig. 5.4 (a)	$-8.11e+003$ $-1.33e+004$ $2.16e+004$ $2.09e+004$ $3.14e+004$ $-5.15e+004$ $1.65e+004$ $-2.50e+003$
14	Fig. 5.4 (b)	$9.24e+005$ $1.09e+005$ $-1.64e+009$ $-1.95e+008$ $6.85e+004$ $8.14e+003$ $-1.18e+008$ $-1.42e+007$
15	Fig. 5.4 (c)	$1.45e+001$ $-3.76e+001$ $2.36e+002$ $-1.25e+001$ $-1.40e+005$ $4.20e+005$ $-2.63e+006$ $1.42e+005$
17	Fig. 5.4 (d)	$-2.46e+005$ $1.65e+005$ $-3.18e+002$ $2.39e+005$ $-8.62e+004$

Continued on Next Page...

Table 5.4 – Continued

Tree Number	Figure	Coefficients
		$-1.89e+005$
		$3.75e+002$
		$-3.27e+005$
19	Fig. 5.4 (e)	$-6.07e+001$
		$1.82e+002$
		$1.03e+006$
		$-2.10e+006$
		$-3.72e-001$
		$6.62e-001$
		$3.45e+003$
		$-5.91e+003$
20	Fig. 5.4 (f)	$-1.12e+003$
		$-3.33e+004$
		$-1.42e+005$
		$1.00e+007$
		$6.55e+002$
		$1.63e+004$
		$1.62e+005$
		$-3.58e+006$

Six best trees were chosen according to coefficients of determination, see again Tab. 5.3. These trees are plotted in Fig. 5.4 and appropriate coefficients are shown in Tab. 5.4. We have also created approximation of experimental data as shown in Fig. 5.5. We present only the tree 10 approximation of experimental data because other approximations are almost the same and the differences are not remarkable.

5.3 Conclusions

The goal of this chapter was to create an approximation of experimental data. The fitted response surfaces are almost perfect; however, a detailed examination of results shows that no tree contains all important input variables (X1, X2, X3, X6 and X7). One of the missing variables X1 – X3 can be replaced by X4 variable since X1 – X4 are linearly dependent constituting a volume unity, see Chapter 2. But still, inclusion of this relationship into GP will surely not bring substantial improvement. Rough enumeration of obtained trees over the allowable domain shows classical signs of overfitting. Therefore, more data are needed or more strict selection mechanism within GP based on the length and/or complexity of regression trees have to be

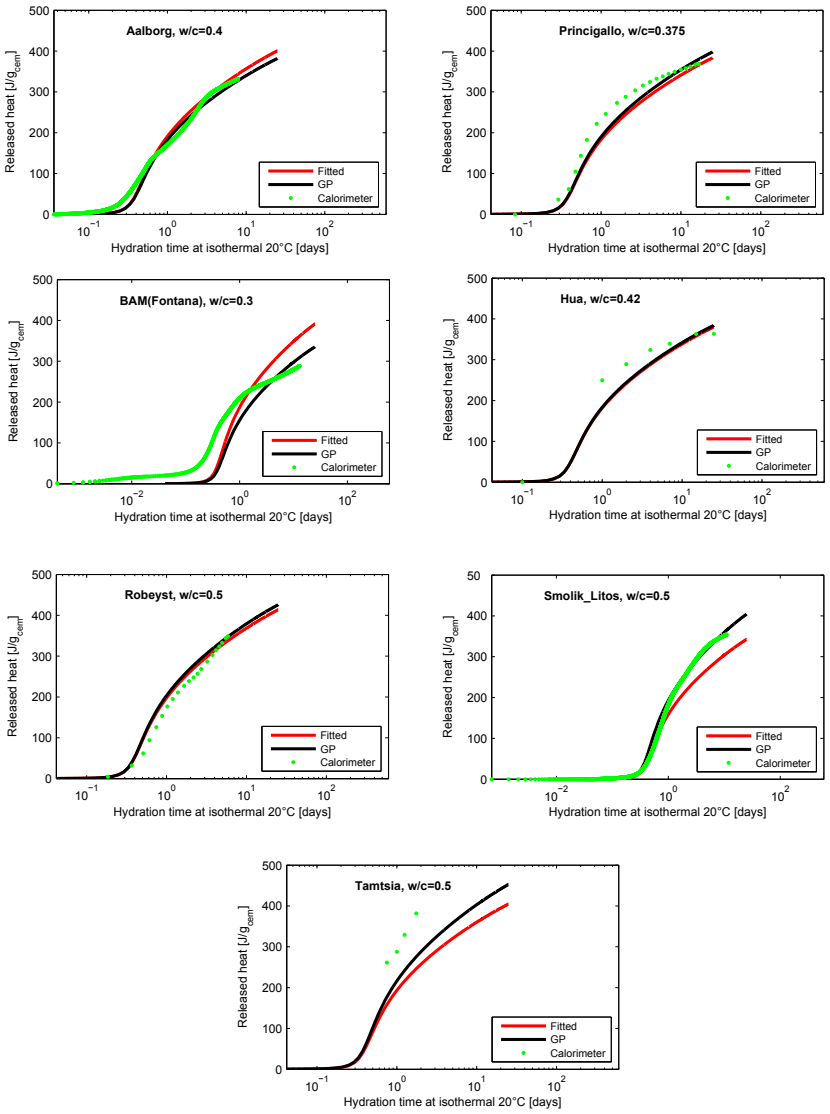


Figure 5.5: Approximation of experimental data. Key: Fitted = *DoH* fitted to all experimental data multiplied by original Q_{pot} form Eq. (1.6); GP = new approximation found by GP

employed. Finally, a problem of constants' placement within GP trees has been successfully solved; however, the Ordinary Least Squares approach limits its dramatically the functions list. Therefore, in a near future our next step will be aimed at the ephemeral random constant methodology.

Chapter 6

SENSITIVITY ANALYSIS AND OPTIMIZATION OF CEMENT PASTE PERFORMANCE

6.1 Introduction[1]

This chapter is aimed at utilizing tools presented in previous chapters and to show potential outcomes offered by soft computing methods. From the area of computer experiments, it may seem that the *sensitivity analysis* (SA) is usually only a process that is preceding the optimization as to minimize a number of parameters and hence computational demands. A frequent term *screening* is usually used for this process, see e.g. references [21] and [66]. However, the general area of sensitivity analysis is broader and can be divided into three groups [83]: (i) *factor screening* mentioned above, (ii) *local SA* most frequently used within the shape optimization domain via computing derivatives of the objective functions with respect to shape parameters [86, 13, 14] and (iii) *global SA* studying the influence of inputs uncertainty to the uncertainties at the outputs [84, 36]. For the latter, so-called *sampling-based SA* [37] is usually applied. Within this chapter, the sampling-based SA will be used to investigate properties of the cement paste hardening process. Particularly, a Spearman's rank correlation coefficient (SRCC) that is able to describe nonlinear monotonic relationship between parameters and the outputs of the studied system will be introduced, see e.g. [54, 50] for more details on the methodology used.

Then, one type of optimization can be seen as a special case of a sensitivity analysis performed only on a small subset of solutions asking a question how these solutions are influenced by inputs. However, the term optimization is wide covering all sciences. In material sciences from which the design of

[1]Partially reproduced from: Z. Bittnar, M. Lepš, and V. Šmilauer. *Soft Computing in Civil and Structural Engineering*, chapter Soft computing in concrete mix optimization, pages 227–246. Stirling: Saxe-Coburg Publications, 2009.

the mixture experiment for the cement paste comes from, the optimization is usually aimed not only at designing one particular composition, but also at the detailed analysis of the influence of individual components influencing the studied optimal properties. Particularly, concrete as a multiscale material inherits several properties from the cement paste. The level of cement paste plays an important role in the design of tailored material since its composition and time evolution can be controlled, monitored, and influenced. A higher level of mortar or concrete is rather relevant for aggregates, being usually not the weakest element in mechanical performance or durability.

The optimization of concrete properties has been a long time the domain of experiments and experts, see e.g. reference [110] for an example of an expert system. Statistical mixture design methods may provide certain guidance to select extreme combinations and to interpolate among experimental results [89]. The leading idea is to optimize not directly the problem itself, but only some approximation of obtained experimental data. One of the first applications of such a methodology is the nice paper [49], where Nelder-Mead simplex optimization algorithm [72] was applied to optimize response surfaces (RSM) [68] of obtained data in a multi-objective fashion by using a desirability approach. For a more detailed description of this type of methodology, a reader is referred to the technical report [88]. From a soft computing perspective, e.g. papers [2] and [107] present an approximation of experimental results by RSM and Artificial Neural Networks, respectively, and the resulting systems are optimized by Genetic Algorithms.

Virtual modeling [93] is another approach to complement an experimental point of view. It is a promising tool not only for a design itself but for the verification of certain assumptions in a numerical way. One of the first applications of the virtual models in cement research is a paper [7], where the sensitivity analysis were done by a virtual model for concrete diffusivity and result were compared to experimental data. The next section presents similar methodology utilizing already presented CEMHYD3D model, see Section 1.2.1.

The second part of this chapter is devoted to *the combination of both the data and the virtual model*. To the best author's knowledge, such an optimiza-

tion approach that directly incorporate measured data and the virtual model together has not been published yet. The advantage in comparison to other methods is that Kriging model presented in Chapter 4 is precise interpolation and therefore, the value of the optimized model is equal to measured data in given data points.

6.2 Sampling-based sensitivity analysis

The objectives of this part include sampling-based sensitivity analysis between input parameters of cement paste and its response in terms of hydration heat and Young's modulus. Such approach is hardly to be achieved experimentally due to the hundreds of evaluations which are time-consuming, or exhibit statistical nature of cement paste and testing device. The reproducibility in virtual tests is guaranteed although the link microstructure-property does not have to exist or can be even wrong. Therefore virtual modeling is exploited particularly for well-established and validated simulations, i.e. for the heat evolution and Young's modulus of cement paste.

As was presented in Chapter 2, the design of the cement composition is a mixture experiment [11]. Therefore, the amount of clinker minerals form unity which complicates the creation of the individual mixtures. For the purposes of this chapter, the classical bounding box approach has been used, see Section 2.4. The bounds for individual parameters are taken from Tab. 1.1. Typically, two hundred input data sets were randomly generated using LHS method which guaranteed low mutual correlation among input parameters and covered the range of feasible input combinations. Since four clinker minerals have to form a unity, weak correlation among them was reported.

6.2.1 Role of parameters influencing the heat of hydration

Bentz, Waller & de Larrard [8] used CEMHYD3D in the simulation of temperature rise under adiabatic conditions. It is well established that heat of hydration originate from individual chemical reactions in which the kinetics of clinker minerals determine heat release rates. The effect of input parameters on the overall heat is examined either in terms of the degree of hydration

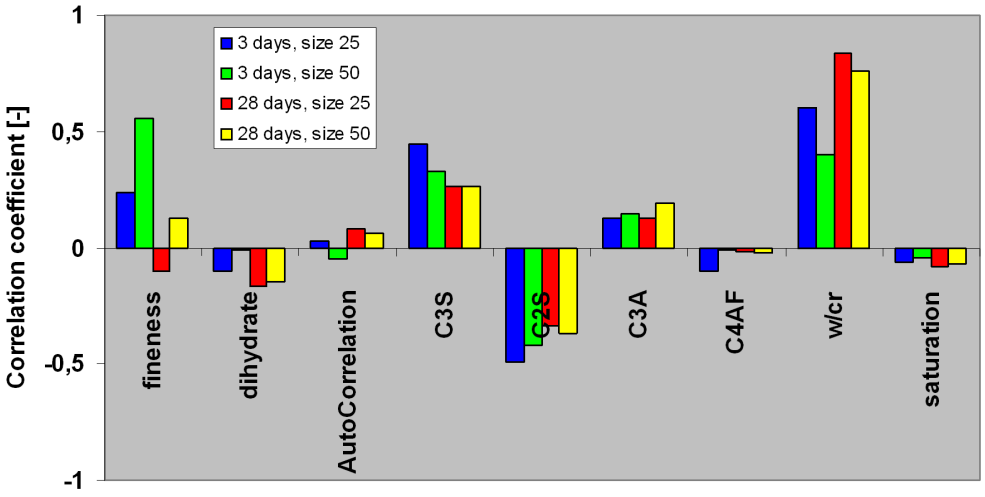


Figure 6.1: Correlation between input parameters and hydration heat at two degrees of hydration and two hydration times

(DoH) or elapsed hydration time, see Fig. 6.1.

Regardless of the above-mentioned evaluation criteria, there are negligible effects in the amount of gypsum, type of autocorrelation file, C_4AF content, and saturated/sealed curing conditions on released heat. Decreasing C_3A content and increasing C_2S amounts have the only significant impact when microstructures are hydrated to the same hydration degrees but not times.

When the released heat is quantified at 3 and 28 days of hydration, the effect of input parameters is more complex. Released heat increases due to higher fineness, especially at early ages, since higher surface area of cement grains accelerates the overall kinetics. The effect of C_3S and C_3A content is about the same, which seems to be contradictory to experiments, where Portland cements with higher C_3A are known to release more heat at early stages. It must be borne in mind that increasing the C_3A amount itself results in the decrease of the other clinkers and that C_3S content is always more dominant over C_3A in Portland cements. Therefore there are two contradicting mechanisms which defeat each other, but the mechanism is uncovered in the comparison against the achieved hydration degree. An increase in w/c

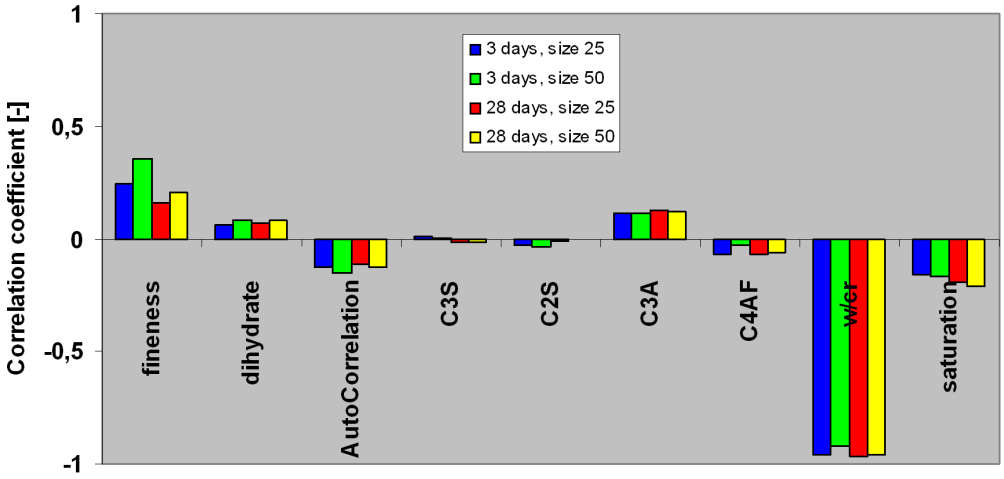


Figure 6.2: Correlation between input parameters and Young's modulus at two degrees of hydration and two hydration times

has an impact on higher released heat since hydration is not slowed down by limited capillary space. This effect is more significant at 28 days.

6.2.2 Role of parameters influencing Young's modulus

Determination of the Young's modulus is based on numerical linear elastic homogenization, utilizing fast Fourier transform (FFT) [67]. Periodic microstructure 50 × 50 × 50 μm from CEMHYD3D model is filtered through solid percolation routine [101] and sampled to a grid of 50 × 50 × 50 Fourier points. The filtering has an effect up to an approximate degree of hydration of 0.3 for the range of input parameters in Tab. 1.1. The average time consumption for FFT-based homogenization is around 10 minutes on 3.2 GHz CPU. Intrinsic elastic properties are taken from [99], including water filled and empty capillary porosity, C-S-H$_{LD}$ and C-S-H$_{HD}$.

The stochastic statistical sensitivity between input parameters and Young's modulus is depicted in Fig. 6.2. The w/c ratio is the only governing parameter, followed by cement fineness. Obviously, decreasing w/c reduces the capillary porosity while increasing the modulus. More homogeneous composites, having the same volume fractions, are known to exhibit higher stiffness

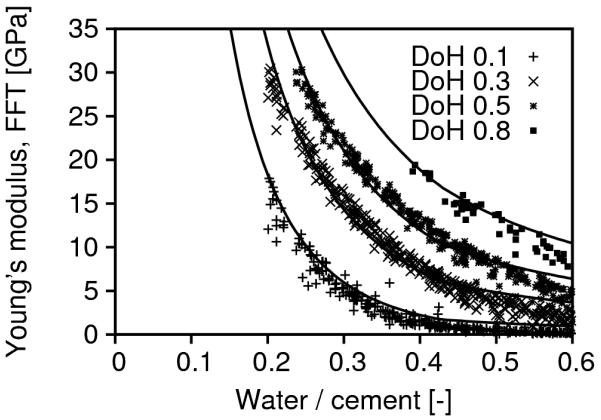


Figure 6.3: Relationship between Young's modulus and w/c ratio and the fit to a power function

which can be interpreted as increasing cement fineness where smaller grains cause more uniformly distributed phases.

The w/c is the most influencing input parameter for the Young modulus so the results may be plotted against w/c only. The situation is depicted in Fig. 6.3. Over 250 sets of different microstructures were generated, left to hydrate, filtered with solid percolation routine and homogenized by FFT-based method. Only 50 microstructures were selected for the degree of hydration of 0.8. The evolution of Young's modulus in Fig. 6.3 can be approximated with a power function. See e.g. reference [102] for its derivation.

6.3 Multi-objective optimization of the mixture composition

A majority of realistic optimization problems do not rely on one objective function only but require simultaneous optimization of several objective functions. For example, one objective function can be merged with the amount of released heat at a given time and an optimal set of input parameters can be found. On the other hand, such combination can be detrimental to other properties, such as stiffness, shrinkage, or crack formation. Several contradicting goals are typical in multi-objective optimization and a compromise has to be found partially to satisfy all of them.

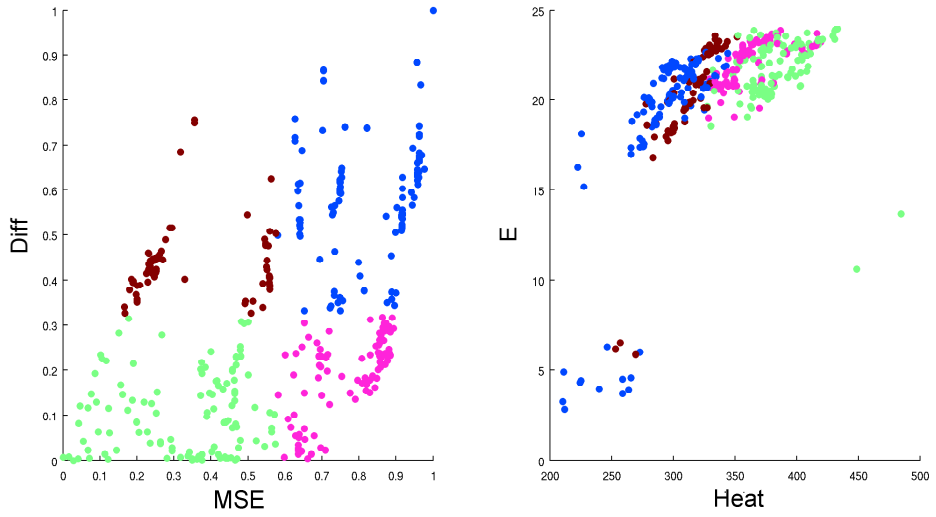


Figure 6.4: Two consecutive Pareto fronts: Difference among CEMHYD3D and Kriging approximation vs. MSE (both normalized, left) and Young's modulus vs. released heat (right)

Therefore, the scalar concept of optimality has to be replaced with Pareto optimality in the multi-objective optimization, see e.g. [55] or [98] for more details. Pareto optimal solution presents a set, for which no better set can be found while minimizing (or maximizing) *all* of the objective functions. The goal is to find such a set of input parameters which maximize the Young modulus while minimizing hydration heat at the time of 28 days of hydration. Moreover, the Kriging approximation from Chapter 4 is used as an estimation of accuracy of the CEHYD3D prediction. We are supported by two proxies - the value of a mean square error (MSE) of the approximation tells us how far we are from the nearest experimental data point whereas the difference between the Kriging and CEMHYD3D result (we will use abbreviation DIFF hereafter) expresses either lack of Kriging fit or inability of CEMHYD3D to properly describe experimental results.

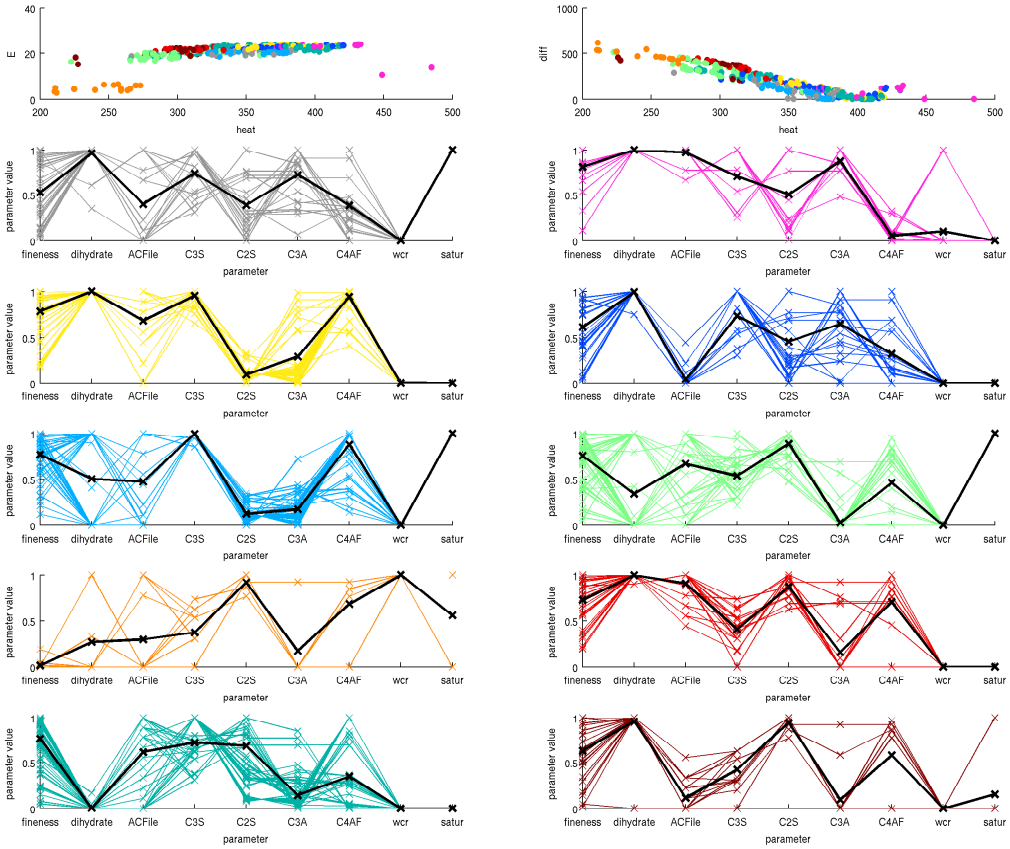


Figure 6.5: Pareto fronts from Fig. 6.4 (top) and ten clusters found by *k-means* cluster analysis: horizontal axes are for parameters, vertical axes are normalized using bounds from Tab. 1.1. Black curves represent means of clusters

Thee multi-objective optimization method used was based on an evolutionary algorithm called Pareto Archived Evolution Strategy (PAES) [47]. Each individual in optimization represents a unique set of input parameters. The PAES algorithm chooses one individual from the population and applies mutation. The offspring is evaluated for dominancy and is rejected or given back into the population archive. The situation after 1000 cycles is depicted

in Fig. 6.4, where two consecutive Pareto fronts are depicted.

The "errors" in Fig. 6.4 (left) are divided into four groups for the sake of convenience. The division lines are made in medians within both normalized proxies, MSE and DIFF, respectively. This division formed four groups: the left lower corner (green dots) represents solutions which are closed to experimental data and the predictions from CEMHID3D and Kriging are close by. These solutions are characterized by relatively high Young's moduli as well as high hydration heat. An opposite case is the right upper corner (blue dots) where we are missing both experimental data and agreement between our two models. These solutions dominate lower grades of hydration heat whereas two clusters with totally different Young's moduli can be found here. Two remaining parts, left upper corner with high DIFF and low MSE (brown dots) and right lower corner with high MSE and low DIFF (magenta dots), are filling the space between two previous limit cases.

Next, we investigate properties of resulting solutions using a k-means clustering analysis [43] available in Matlab. As can be seen from Fig. 6.5, different compositions can be found. For instance, the left cluster with low Young's moduli and low hydration heat (orange color) is composed of high w/c, high C_2S and low fineness. Oppositely, the whole middle part of the Pareto fronts is composed with minimal level of the w/c ratio. Finally, all these results can serve as a basis for new mixture compositions aimed at tailored properties.

6.4 Conclusions

In this chapter, it has been shown that the soft computing techniques are easily applicable to concrete engineering. The most fundamental quantity - hydration heat - was studied in detail together with the estimation of the Young's modulus. A cellular automata-based hydration model CEMHYD3D is readily applicable for sensitivity analysis and multiscale simulation as a virtual model of much more expensive real tests. It is important to mention that the hydration model underestimates hydration heat, roughly after a week of hydration. Here, the comparison is possible due to calibrated Kriging model to real data sets.

Kriging metamodel developed in Chapter 4 approximated experimental data of hydration heat well and enables an efficient combination of the virtual and data-based model. It should be emphasized that the metamodel does not rely on any physical process and can be applied to any data sets. The combination of these two models then offers reliable optimization that can show weaknesses in both the insufficient cover by the data as well as incompatibility and potential deficiencies among models.

Chapter 7

CONCLUSIONS

Soft computing (SC) is a relatively new research area, however its definition is not clear. Usually contains methodologies that do not solve given problem directly. However, categorization based on such definition is fuzzy. Imagine a list of similarities of several interpolation methods as presented in [39] shown in Fig. 7.1.

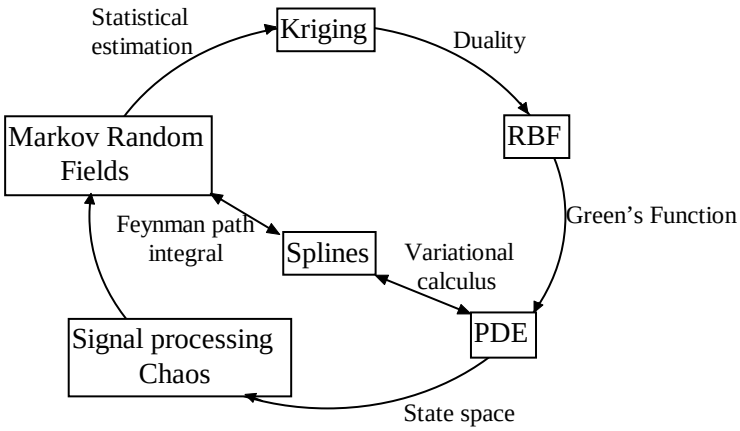


Figure 7.1: Similarity among several interpolation methods, reprinted from [39]. Key: RBF = Radial Basis Functions, PDE = Partial Differential Equations

Kriging and Radial Basis Functions are tackled as a part of SC. However, e.g. spline approximation can be used for inexact pre-evaluations and then will be a part of SC. Also a nice quotation by Peter Convey and Roger Highfield in the book *Frontiers of Complexity* describes a relation of classical branches of sciences to SC:

> Natural selection can be seen as cheating for scientists who want to find discoveries only by perfect deduction.

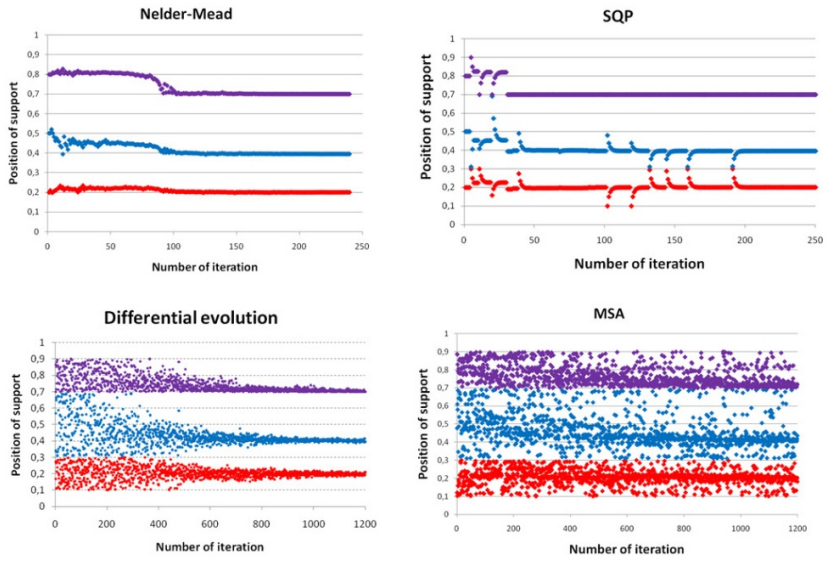


Figure 7.2: Comparison of convergence histories for four optimization algorithms for a structural benchmark. Figure provided by courtesy of Jaroslav Kabeláč. Key: Nelder-Mead = simplex method [72], SQP = Sequential Quadratic Programming, Differential Evolution = simple evolutionary algorithm presented in [90], MSA = Modified Simulated Annealing [60]

From this view, SC is commonly seen as a simpler and easy way of doing research. Therefore, a big honor for the soft computing community comes with the publication of the paper [85] in the Science journal. The article is devoted to the GP that has been applied to rediscovery of known analytical solutions for dynamical systems.

From author's own perspective and experience, the application of soft computing methods for the given problem is a part of multi-objective decision making. If we compare computational demands of several optimization methods in solving continuous optimization problems, see e.g. comparison in Fig. 7.2, the gradient-based methods will usually lead. But sometimes they can completely fail, see e.g. a chapter [105] for details on problems usually encountered in optimization practice. Another objective is time spent on the development of the code. For instance, in the case presented above in Fig. 7.2, the development of the MSA algorithm was 14 days in contrast to the Sequential Quadratic Programming method that cost almost three

months.

In the last few decades the growing price of material and labor leads to huge developments in the field of SC, since anything that can lead to the minimization of the price of real experiments was justifiable. However, since the beginning of the millennia, the price of electrical energy needed for computers grows such that the price of computationally demanding calculations like GP is continuously increasing and can limit future developments.

BIBLIOGRAPHY

[1] Simplex volumes and the Cayley-Menger determinant. www pages: http://www.mathpages.com/home/kmath664/kmath664.htm, 2009.

[2] K. Sobolev A. Amirjanov. Genetic algorithm for cost optimization of modified multi-component binders. *Building and Environment*, 41:195–203, 2006.

[3] L. F. Alvarez, V. V. Toropov, D. C. Hughes, and A. F. Ashour. Approximation model building using genetic programming methodology: Application. Department of Civil and Environmental Engineering, University of Bradford, UK, 2000.

[4] P. Audze and V. Eglais. New approach for planning out of experiments. *Problems of Dynamics and Strengths*, 35:104–107, 1977. Zinatne Publishing House.

[5] S. J. Bates, J. Sienz, and D. S. Langley. Formulation of the audze-eglais uniform latin hypercube design of experiments. *Adv Eng Softw*, 34(8):493–506, 2003.

[6] D. P. Bentz. *CEMHYD3D: A Three-Dimensional Cement Hydration and Microstructure Development Modeling Package. Version 3.0.* Building and Fire Research Laboratory Gaithersburg, 2005.

[7] D. P. Bentz, E. J. Garboczi, and E. S. Lagergren. Multi-scale microstructural modeling of concrete diffusivity: Identification of significant variables. *Cement, Concrete, and Aggregates, CGAGDP*, 20(1):129–139, 1998.

[8] D. P. Bentz, V. Waller, and F. de Larrard. Prediction of adiabatic temperature rise in conventional and high-performance concretes using a 3-d microstructural model. *Cement and Concrete Research*, 28(2):285–297, 1998.

[9] Ch. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[10] E. Cantú-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, 2001.

[11] N. Chantarat. *Modern Design of Experiments Methods for Screening and Experimentations with Mixture and Qualitative Variables*. PhD thesis, The Ohio State University, 2003.

[12] L. Chen and M. Holst. Efficient mesh optimization schemes based on Optimal Delaunay Triangulations. *Computer Methods in Applied Mechanics and Engineering*, To be published, 2010.

[13] K. K. Choi and N. H. Kim. *Structural Sensitivity Analysis and Optimization 1: Linear Systems*. Springer, 2005.

[14] P. W. Christensen and A. Klarbring. *An Introduction to Structural Optimization*. Springer, 2009.

[15] T. M. Cioppa and T. W. Lucas. Efficient nearly orthogonal and space-filling latin hypercubes. *Technometrics*, 49(1):45–55, February 2007.

[16] C. A. C. Coello. Evolutionary multiobjective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36, February 2006.

[17] J. A. Cornell. Experiments with Mixtures: A Review. *Technometrics*, 15(3):437–455, 1973.

[18] J. A. Cornell. Experiments with Mixtures: An Update and Bibliography. *Technometrics*, 21(1):95–106, 1979.

[19] K. Crombecq, I. Couckuyt, D. Gorissen, and T. Dhaene. Space-filling sequential design strategies for adaptive surrogate modelling. In B. H. V. Topping and Y. Tsompanakis, editors, *Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*. Civil-Comp Press, Stirlingshire, UK, 2009.

[20] D. R. Cruise. Plotting the composition of mixtures on simplex coordinates. *Journal of Chemical Education*, 43(1):30, 1966.

[21] A. Dean and S. Lewis. *Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics*. Springer, 2006.

[22] L. Devroye. *Non-Uniform Random Variate Generation*. Springer, 1986.

[23] J. Dréo, A. Pétrowski, P. Siarry, and E. Taillard. *Metaheuristics for Hard Optimization: Methods and Case Studies*. Springer, 1st edition, December 2005.

[24] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, October 7, 2008.

[25] M. T. M. Emmerich. *Single- and multi-objective evolutionary design optimization assisted by gaussian random field metamodels*. PhD thesis, Universität Dortmund, 2005.

[26] E. M. R. Fairbairn, N. F. F. Ebecken, C. N. M. Paz, and F.-J. Ulm. Determination of probabilistic parameters of concrete: solving the inverse problem by usign artificial neural networks. *Computers & Structures*, 78(1–3):497–503, 2000.

[27] K-T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments*. Chapman & Hall/CRC, 2006.

[28] J. Fish and T. Belytschko. *A First Course in Finite Elements*. John Wiley & Sons, Ltd, 2007.

[29] E. J. Garboczi, J. W. Bullard, and D. P. Bentz. Virtual testing of cement and concrete – USA 2004. *Concrete International*, 26(12):33–37, 2004.

[30] I. G. Georgoudas, G. Ch. Sirakoulis, and I. Th. Andreadis. An Intelligent Cellular Automaton Model for Crowd Evacuation in Fire. In *19th IEEE International Conference on Tools with Artificial Intelligence*, pages 36 – 843. IEEE Computer Society, 2007.

[31] T. Goel, R. T. Haftka, W. Shyy, and L. T. Watson. Pitfalls of using a single criterion for selecting experimental designs. *International Journal for Numerical Methods in Engineering*, 75(2):127–155, 2008.

[32] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[33] L. F. González, E. J. Whitney, J. Périaux, M. Sefrioui, and K. Srinivas. Multidisciplinary aircraft conceptual design and optimisation using a robust evolutionary technique. In Neittaanmäki et al. [71].

[34] B. Sendhoff H.-G. Beyer. Robust optimization – a comprehensive survey. *Comput. Methods Appl. Mech. Engrg.*, 196:3190–3218, 2007.

[35] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2nd edition, 1998.

[36] J. C. Helton, J. D. Johnson, W. L. Oberkampf, and C. J. Sallaberry. Sensitivity analysis in conjunction of with evidence theory representations epistemic uncertainty. *Reliab Eng Syst Safe*, 91(10-11):1414–1434, 2006.

[37] J. C. Helton, J. D. Johnson, C. J. Sallaberry, and C. B. Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliab Eng Syst Safe*, 91(10-11):1175–1209, 2006.

[38] M. Hofwing and N. Strömberg. D-optimality of non-regular design spaces by using a Bayesian modification and a hybrid method. *Structural and Multidisciplinary Optimization*, Sent for publication, 2010.

[39] P. Hornby, F. G. Horowitz, and F. Path. Closely related alternatives to kriging interpolation. Unpublished manuscript, 1996. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.9697`.

[40] C. Hua, P. Acker, and A. Ehrlacher. Analyses and models of the autogenous shrinkage of hardening cement paste. I. modelling at macroscopic scale. *Cement and Concrete Research*, 25(7):1457–1468, 1995.

[41] B. Husslage. *Maximin Designs for Computer Experiments*. PhD thesis, Tilburg University, 2006.

[42] L. Ingber. Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling*, 18(11), 1993.

[43] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[44] E. Janouchová and A. Kučerová. Competitive comparison of optimal designs of experiments for sampling-based sensitivity analysis. *Computers & Structures*, Sent for publication, 2010.

[45] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[46] M. K. Karakasis and K. C. Giannakoglou. On the use of surrogate evaluation models in multi-objective evolutionary algorithms. In Neittaanmäki et al. [71].

[47] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[48] J. R. Koza. *Genetic Programming*. A Bradford book, sixth edition, 1998.

[49] S. E. Krampe. *Computer Applications in the Polymer Laboratory*, chapter Analysis and Optimization of Constrained Mixture-Design Formulations. American Chemical Society, 1986.

[50] A. Kučerová. *Identification of nonlinear mechanical model parameters based on softcomputing methods*. PhD thesis, Ecole Normale Supérieure de Cachan, Laboratoire de Mécanique et Technologie, 2007.

[51] A. Kučerová, M. Lepš, and J. Skoček. Large black-box functions optimization using radial basis function networks. In B. H. V. Topping, editor, *Proceedings of Eighth International conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*, Stirling, United Kingdom, 2005. Civil-Comp Press.

[52] A. Kučerová, M. Lepš, and J. Zeman. Back analysis of microplane model parameters using soft computing methods. *Computer Assisted Mechanics and Engineering Sciences*, 14(2):219–242, 2007. (Special issue of The International Symposium on Neural Networks and Soft Computing (NNSC-2005)).

[53] J. Lee and P. Hajela. Application of classifier systems in improving response surface based approximations for design optimization. *Computers & Structures*, 79:333–344, 2001.

[54] D. Lehký. *Inverse Stochastic Analysis of Concrete Structures (in czech)*. PhD thesis, VUT v Brně, 2005.

[55] M. Lepš. *Single and Multi-Objective Optimization in Civil Engineering with Applications*. PhD thesis, Czech Technical University in Prague, 2005.

[56] M. Lepš. Load-balancing of master-slave evolutionary algorithm for parameters identification. In *Proceedings of the First International Conference on Parallel, Distributed and Grid Computing for Engineering*. Stirling: Civil-Comp Press Ltd, 2009.

[57] S. N. Lophaven, H. B. Nielsen, and J. Sondergaard. Aspects of the Matlab toolbox Dace. Technical Report IMM-REP-2002-13, Informatics and Mathematical Modelling, Technical University of Danemark, 2002. 44 pages, `http://www.imm.dtu.dk/~hbn/publ/TR0213.ps`.

[58] H. Maaranen, K. Miettinen, and A. Penttinen. On initial populations of a genetic algorithm for continuous optimization problems. *J. of Global Optimization*, 37:405–436, March 2007.

[59] S. W. Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, 1995.

[60] S. W. Mahfoud and D. E. Goldberg. Parallel recombinative simulated annealing - A genetic algorithm. *Parallel Computing*, 21(1):1–28, 1995.

[61] R. Mahnken. *Encyclopedia of Computational Mechanics Part 2. Solids and Structures*, chapter Identification of Material Parameters for Constitutive Equations. John Wiley & Sons, Ltd., 2004.

[62] G. Matheron. Principles of geostatistics. *Economic Geology*, 58:1246–1266, 1963.

[63] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 3rd edition, 1999.

[64] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Dordrecht, 1999.

[65] D. C. Montgomery. *Design and Analysis of Experiments, 5th Edition*. Wiley, 5 edition, June 2000.

[66] M. D. Morris. Input screening: Finding the important model inputs on a budget. *Reliability Engineering and System Safety*, 91:1252–1256, 2006.

[67] H. Moulinec and P. Suquet. A fast numerical method for computing the linear and nonlinear mechanical properties of composites. *Comptes Rendus de l'Academie des Sciences Serie II*, 318(11):1417–1423, 1994.

[68] R. H. Myers and D. C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. New York: Wiley, 1995.

[69] E. Myšáková and M. Lepš. Comparison of space-filling design strategies. In *Engineering Mechanics 2010*, Sent for publication, 2010.

[70] H. Nakayama, K. Inoue, and Y. Yoshimori. Approximate optimization using computational intelligence and its application to reinforcement of cable-stayed bridges. In Neittaanmäki et al. [71].

[71] P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, P. Périaux, and D. Knörzer, editors. *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*, Jyväskylä, 24–28 July 2004 2004.

[72] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[73] A. M. Neville. *Properties of Concrete*. John Wiley & Sons, Inc., 1997.

[74] NRMCA. Concrete in practice. Technical report, National Ready Mixed Concrete Association, 1978–2007.

[75] P-O. Persson and G. Strang. A simple mesh generator in MATLAB. *SIAM Review*, 46(2):329–345, June 2004.

[76] M. Petelet, B. Iooss, O. Asserin, and A. Loredo. Latin hypercube sampling with inequality constraints. *AStA Advances in Statistical Analysis*, To be published, 2010.

[77] B. Pichler, R. Lackner, and H.A. Mang. Back analysis of model parameters in geotechnical engineering by means of soft computing. *International Journal for Numerical Methods in Engineering*, 57(14):1943–1978, 2003.

[78] A. Princigallo, P. Lura, K. van Breugel, and G. Levita. Early development of properties in a cement paste: A numerical and experimental study. *Cement and Concrete Research*, 33(7):1013 – 1020, 2003.

[79] D. Quagliarella. Airfoil design using Navier-Stokes equations and an asymmetric multi-objective genetic algorithm. In G. Bugeda, J.-A. Désidéri, J. Périaux, M. Schoenauer, and G. Winter, editors, *Evolutionary Methods for Design, Optimization and Control: Applications to Industrial and Societal Problems*, Eurogen 2003. International Center for Numerical Methods in Engineering (CIMNE), September 2003.

[80] N. Robeyst, E. Gruyaert, and N. De Belie. *Advances in Construction Materials 2007: Ultrasonic and calorimetric measurements on fresh concrete with blast-furnace slag*, chapter VI, pages 497–504. Springer Berlin Heidelberg, 2007.

[81] K. Rodríguez-Vázquez and C. Oliver-Morales. Function approximation by means of multi-branches genentic programming. IIMAS-UNAM, Circuito Escolar, Ciudad Universitaria, Mexico, 2004.

[82] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

[83] A. Saltelli, K. Chan, and E. M. Scott. *Sensitivity analysis*. NY:Wiley, New York, 2000.

[84] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis. The Primer*. JWS, 2008.

[85] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *SCIENCE*, 324, 2009.

[86] O. Sigmund and M. P. Bendsœ. *Topology Optimization: Theory, Methods and Applications*. Springer-Verlag, 2nd edition, 2003.

[87] S. Silva. *GPLAB Manual*. ECOS - Evolutionary and Complex Systems Group, Portugal, 3^{rd} edition, April 2007.

[88] M. J. Simon. Concrete mixture optimization using statistical methods: Final report. Technical Report FHWA-RD-03-060, National Institute of Standards and Technology, 2003.

[89] M. J. Simon, E. S. Lagergreen, and K. A. Snyder. Concrete mixture optimization using statistical mixture design methods. In *Proceedings of the PCI/FHWA*, pages 230–244, 1997.

[90] R. Storn and K. Price. Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, University of Berkeley, 1995.

[91] A. Strauss, K. Bergmeister, D. Novák, and D. Lehký. Stochastische Parameteridentifikation bei Konstruktionsbeton für die Betonerhaltung. *Beton- und Stahlbetonbau*, 99(12):967–974, 2004. [in German].

[92] M. Streeter and L. A. Becker. Automated discovery of numerical approximation formulae via genetic programming. Department of Computer Science, Worcester Polytechnic Institute, Worcester, 2003.

[93] P. Stroeven, L. J. Sluys, Z. Guo, and M. Stroeven. Virtual reality studies of concrete. *Forma*, 21:227–242, 2006.

[94] B. T. Tamtsia, J. J. Beaudoin, and J. Marchand. The early age short-term creep of hardening cement paste: Load-induced hydration effects. *Cement and Concrete Composites*, 26:481–489, 2004.

[95] V. V. Toropov, S. J. Bates, and O. M. Querin. Generation of extended uniform latin hypercube designs of experiments. In B. H. V. Topping, editor, *Proceedings of the Ninth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*. Civil-Comp Press, Stirlingshire, UK, 2007.

[96] M. Valtrová. Computation aspects of Kriging in chosen engineering problems. Master's thesis, CULS Prague, 2009.

[97] E. R. van Dam, G. Rennen, and B. Husslage. Bounds for maximin latin hypercube designs. *Operations Research*, 57:595–608, May 2009.

[98] Z. Vitingerová. *Evolutionary Algorithms for Multi-Objective Parameter Estimation*. PhD thesis, CTU in Prague, To be published, 2010.

[99] V. Šmilauer. *Elastic properties of hydrating cement paste determined from hydration models*. PhD thesis, CTU in Prague, 2006. `http://mech.fsv.cvut.cz/~smilauer/`.

[100] V. Šmilauer. *Multiscale Modeling of Hydrating Concrete*. Saxe-Coburg Publications, Stirling, To appear, 2010.

[101] V. Šmilauer and Z. Bittnar. Microstructure-based micromechanical prediction of elastic properties in hydrating cement paste. *Cement and Concrete Research*, 36(9):1708–1718, 2006.

[102] V. Šmilauer, M. Lepš, and Z. Vitingerová. Multiobjective optimization of cement paste performance. In *Engineering Mechanics 2008*, pages 961–967. Ústav termomechaniky AV ČR, 2008.

[103] J. F. Wang, J. Periaux, and M. Sefrioui. Parallel evolutionary algorithms for optimization problems in aerospace engineering. *Journal of Computational and Applied Mathematics*, 149:155–169, 2002.

[104] Z. Waszczyszyn and L. Ziemianski. *Parameter identification of materials and structures*, chapter Neural networks in the identification analysis of structural mechanics problems, pages 265–340. Springer-WienNewYork, 2005.

[105] T. Weise, M. Zapf, R. Chiong, and A. J. Nebro. *Nature-Inspired Algorithms for Optimisation*, volume SCI 193, chapter Why Is Optimization Difficult?, pages 1–50. Springer-Verlag Berlin Heidelberg, 2009.

[106] S. Wolfram. *Cellular Automata and Complexity*. Perseus Publishing, 2002.

[107] I-Cheng Yeh. Computer-aided design for optimum concrete mixtures. *Cement & Concrete Composites*, 29:193–202, 2007.

[108] Y. S. Yeun, Y. S. Yang, W. S. Ruy, and B. J. Kim. Polynomial genetic programming for response surface modeling. Part 1: A methodology. *Structural and Multidisciplinary Optimization*, 29:19–34, 2005.

[109] L. A. Zadeh. Fuzzy logic, neural networks, and soft computing. *Commun. ACM*, 37(3):77–84, March 1994.

[110] M. F. M. Zain, M. N. Islam, and I. H. Basri. An expert system for mix design of high performance concrete. *Advances in Engineering Software*, 36:325–337, 2005.

Appendix A

COMPUTATION OF SIMPLEX VOLUME

Because we know the coordinates of simplex vertices, we use the formula which requires these (and only these) inputs [1].

The computation of a volume of a simplex in 2D (3 vertices):

$$V_2 = \frac{1}{2!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} \\ 1 & x_{1(2)} & x_{2(2)} \\ 1 & x_{1(3)} & x_{2(3)} \end{vmatrix}$$

The computation of a volume of a simplex in 3D (4 vertices):

$$V_3 = \frac{1}{3!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} & x_{3(1)} \\ 1 & x_{1(2)} & x_{2(2)} & x_{3(2)} \\ 1 & x_{1(3)} & x_{2(3)} & x_{3(3)} \\ 1 & x_{1(4)} & x_{2(4)} & x_{3(4)} \end{vmatrix}$$

The computation of a volume of a simplex in nD ($n+1$ vertices):

$$V_n = \frac{1}{n!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} & \dots & \dots & x_{n(1)} \\ 1 & x_{1(2)} & x_{2(2)} & \dots & \dots & x_{n(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1(n+1)} & x_{2(n+1)} & \dots & \dots & x_{n(n+1)} \end{vmatrix}$$

In the notation $x_{a(b)}$ a is a variable (dimension), b is a design point.

Appendix B

LIST OF (0-5)-DIMENSIONAL SIMPLEX ELEMENTS

dim.	**name**	**vertices**	**edges**	**faces**	**cells**	**4-faces**	**5-faces**	**sum**
0	Point	1						1
1	Line segment	2	1					3
2	Triangle	3	3	1				7
3	Tetrahedron	4	6	4	1			15
4	Pentachoron	5	10	10	5	1		31
5	Hexateron	6	15	20	15	6	1	63

Table B.1: List of (0-5)-dimensional simplex elements.