CAD-based Evolutionary Design Optimization with CATIA V5

Oliver König*, Marc Wintermantel

EVEN - Evolutionary Engineering AG, Zurich, Switzerland

Abstract

Modern CAD systems offer the most simple and natural way to represent complex mechanical structures heavily constrained through a variety of functional, aesthetic, and manufacturing demands. We present two concepts for evolutionary design optimization directly relying on CAD-based design encodings of arbitrary mechanical structures. One relies on the associative-parametric description of the design. Its full potential is shown by the successful optimization of a race car rim, a complex structure heavily constrained through manifold requirements. In the second concept, we present an Evolutionary Algorithm that directly relies on the representation of mechanical structures in a CAD system, on CAD features as ribs or webs. This approach is implemented in CATIA V5, using its C++ component application architecture (CAA) and the Evolving Objects (EO) framework. We demonstrate the superior performance of this concept in comparison to a natively in CATIA implemented Simulated Annealing optimizer by means of a minimum compliance test problem. Finally, the weight of an end plate of a fuel cell stack subject to a stress constraint is optimized resulting in a new and very appealing design.

Keywords: Evolutionary Design, Evolutionary Algorithms, Structural Optimization, Computer Aided Design, CATIA V5, CAA V5

1 Introduction

Within the scope of this paper the combination of modern knowledge-based Computer Aided Design (CAD) systems with Evolutionary Algorithms (EAs) is presented. CAD systems build the backbone of modern product development processes by offering the most simple and natural way of representing complex mechanical structures that are heavily constrained through a variety of functional, aesthetic, and manufacturing demands. We present two concepts for evolutionary design optimization. One relies on the associative-parametric description of the design. A

*Contact: Dr.sc.Techn. Oliver König, EVEN - Evolutionary Engineering AG, Leonhardstr. 25, CH-8092 Zurich, E-Mail: koenig@even-ag.ch, Web: http://even-ag.ch

Table 1: Four important forms of Evolutionary Algorithms

Algorithm	Abbreviation	References
Genetic Algorithms	**GA**	[10, 8, 9]
Evolutionary Programming	**EP**	[7]
Evolution Strategies	**ES**	[15, 16, 22, 1]
Genetic Programming	**GP**	[12]

second is directly based on the representation of mechanical structures in CAD systems.

The paper is structured to first give an introduction to the general concepts of Evolutionary Algorithms in Section 2. Then, the representation and the handling of mechanical structures in the CAD system CATIA V5 are discussed in Section 3. Section 4 presents two concepts for the adaptation of EAs for CAD-based design optimizations. The optimization of a race car rim discussed in Section 5 highlights the potential of using EAs within associative-parametric CAD designs. The paper concludes with an application of the CAD-feature based EAs in Section 6 and a discussion of the results in Section 7.

2 Basics of Evolutionary Algorithms

Evolutionary Algorithms use an analogy with natural evolution to perform search by evolving solutions to problems, usually working with a large collection of solutions at a time. The common underlying idea behind the techniques is to have a given population of individuals (solutions), whereas the environmental pressure causes natural selection and thereby the fitness of the population is improving.

In structural optimization, EAs are used in many different forms. Commonly, they are divided into four categories, see Table 1. However, they are all based on similar evolutionary principles. Thus, we will use a more modern terminology also used by [4] and [21]: we generally speak just of Evolutionary Algorithms. All the above listed strategies can be considered as specializations of general EAs, which we will describe below.

EAs use two separate spaces, i.e. the search space and the solution space, respectively. The search space contains coded solutions (*genotype*) to the problem, whereas the solution space contains actual solutions (*phenotype*). EAs maintain a population of $P \in \mathbb{N}$ individuals, each individual consisting of a genotype and a corresponding phenotype. A simple EA works as outlined in Figure 1. A population of a given number of individuals is randomly initialized. The fitness of each individual of the population is determined by evaluating the objective function and assigning the resulting fitness value *(Evaluation)*. The fitness value from the evaluation is then used to determine how many copies of each individual are placed into a temporary area often termed the mating pool. For the reproduction process parents are picked from the mating pool with some pressure of preferring parents with better fitness values *(Selection)*. Offspring are generated by applying the crossover operator that randomly allocates genes from each parent's genotype to each offspring's genotype. Then, mutation is occasionally applied to the offspring. A new population is built

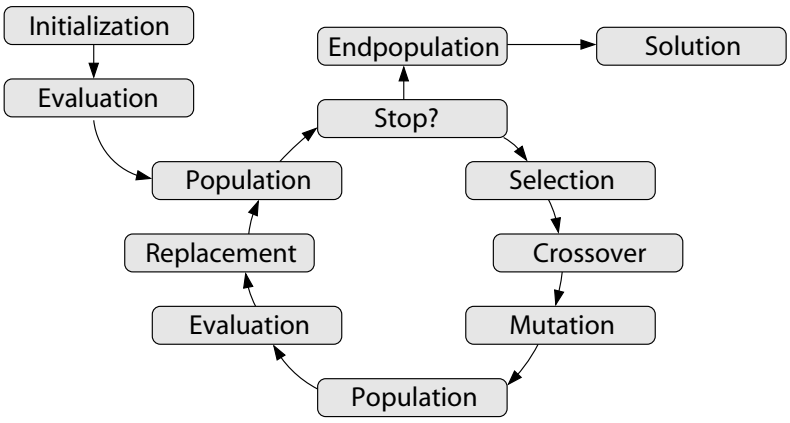


Figure 1: Basic scheme of an Evolutionary Algorithm

by deterministic or stochastic choice of parents and offspring (*Replacement*). Subsequently, the new fitness values of the individuals of the population are evaluated. The entire process of evaluation, reproduction and replacement continues until a given *stopping criterion* is achieved. This can for example be defined with the values of a time or a generation counter or the best fitness of the population.

3 Handling of mechanical structures in CATIA V5

With regard to the presented concepts for the combination of EAs and CAD systems in Section 4, the representation of mechanical structures in the CAD system CATIA V5[1] is discussed. The commercial program CATIA V5 was chosen because it represents the most modern CAD system available regarding programming concepts and data structures. In the following it will also be outlined how the mechanical behavior of a mechanical structure can be analyzed with this software. The explanations given are based on the manuals of CATIA V5 and on the documentation of its C++ interface CAA.

3.1 Representation of mechanical structures

A complex mechanical structure, e.g. a car or an airplane, can be represented in a `Product` document in CATIA V5. A `Product` document consists of different components which can be `Product` documents themselves or which can be single mechanical structures represented in `Part` documents. `Part` documents hold four containers:

Product container. It manages the integration of a `Part` document into the `Product` document.

Specification container. It contains the actual design representation of the mechanical object. The design is defined by a list of mechanical features being hierarchically grouped in a specification tree.

[1] http://www.ibm.com/catia

Scope container. This container is concerned about generic naming concepts providing stable and flexible ways to reference topological geometry objects in the specification container.

Geometrical container. Mechanical features handled in the specification container essentially capture the design intent of the user. From this features the actual shape of a mechanical object is then computed using an underlying modeler, also called update mechanism, and the topological results are stored in the geometrical container.

Figure 2 shows such a `Part` document of a simplified shaft structure with its list of mechanical features in the specification tree and with its topological result as simple solid body. Being interested in the representation of structures, the list of

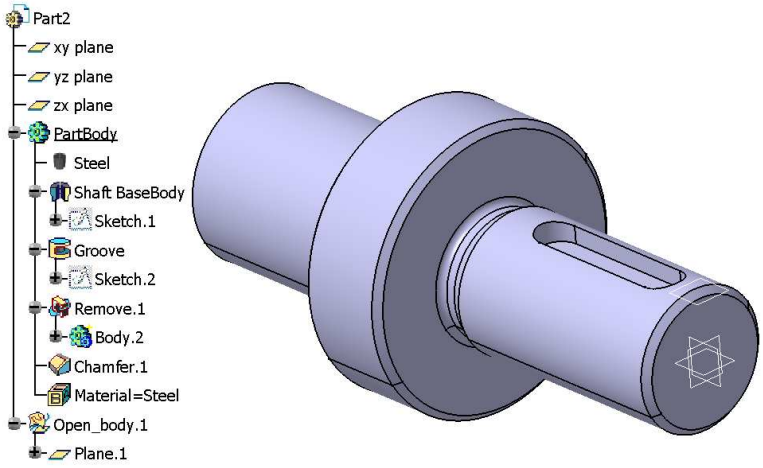


Figure 2: CATIA representation of a mechanical structure.

mechanical features in the specification container is further analyzed. There are two kinds of mechanical features:

Structural features. They structure the `Part` document by aggregating mechanical features. Every structural feature represents a certain topological object as it is defined at this stage of the update process. Looking at the example shaft structure in Figure 2, structural features are `Part2`, `PartBody`, `Body2`, all sketches, and `Open_Body.1`.

Geometrical features. These are features to which the update mechanism associates a topological result. Again looking at Figure 2, `Shaft BaseBody`, `Groove`, `Remove.1` and `Chamfer.1` are geometrical features.

If the shaft structure is updated, the program first creates the basic shaft body from the defining `Sketch.1` and adds it to the empty `PartBody`. Then the `Groove` subtracts some material from the basic solid. Another body for the nut is created and is immediately subtracted from the `PartBody`. Finally, some sharp edges are removed by the feature `Chamfer.1`.

The presented representation of a mechanical object is needed to provide a parametric-associative CAD. The mechanical features are finally defined through

parameters or through direct contact with neighbor objects. After changing a parameter or a defining entity, an automatic update of the structure shall succeed and result in an adapted topological body.

3.2 Analyzing the mechanical behavior of structures

The CAD system CATIA provides a variety of possibilities to analyze the mechanical behavior of structures:

- After assigning a material specification to a body, CATIA can directly display properties such as weight, center of gravity, or moment of inertia.
- Dimensions, distances, areas etc. of the resulting structures can directly be measured.
- Through a fully integrated FE tool sensors can be applied to the structure measuring e.g. the stress level at a certain location, evaluating the maximum deformation, or calculating eigenfrequencies.

Fitness values for an evolutionary optimization can easily be computed from this measures and sensors.

3.3 Parameter optimization workbench in CATIA V5

The Product Engineering Optimizer (PEO) workbench in CATIA offers an easy-to-use parameter optimization interface. Any real-valued parameters accessible from within CATIA V5 can be selected as parameters to be optimized. They are typically provided with an upper and a lower limit for the optimization as can be seen in Figure 3. Additionally, an optimization objective has to be specified which must

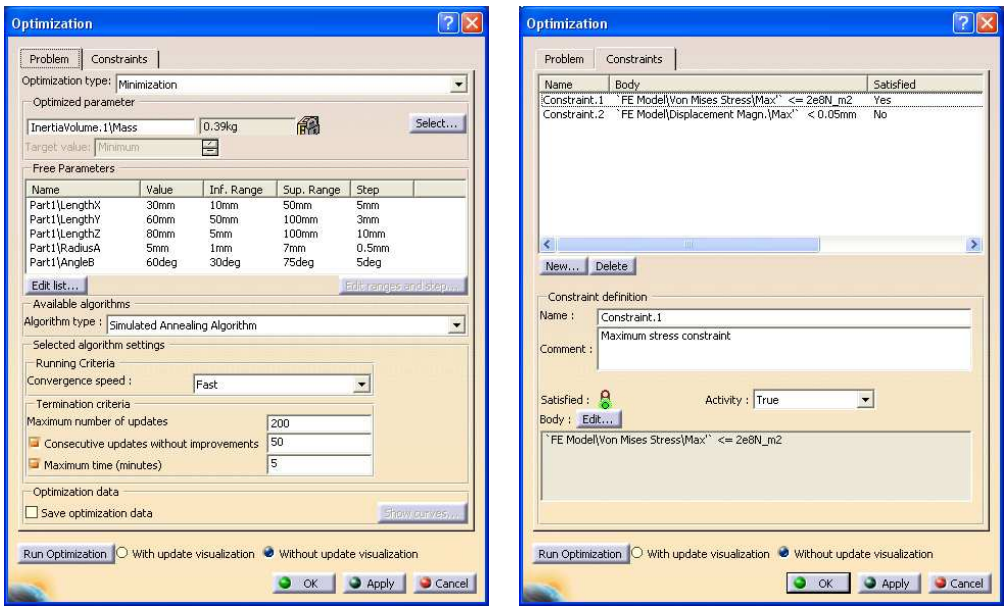


Figure 3: User interface of the CATIA Product Engineering Optimizer workbench.

be a parameter from within CATIA V5 as well. Further, it has to be defined whether this parameter should be minimized, maximized, or if a target value has to be achieved. Typically, the parameter selected as design objective is an analysis result, e.g. a maximum von Mises stress value from FEA, or a sensor value defined somewhere on the structure, e.g. a minimum distance between two objects. Finally, constraints for the optimization can be defined by comparing parameter values with given limits as can be seen in Figure 3.

As optimization engines the PEO provides three basic algorithms:

Conjugate Gradients: This optimization engine is based on classical mathematical programming techniques [6, 13]. For smooth search space topologies it is very efficient, but it is not able to get out of local search space extrema. Consequently, it is mainly suited for optimization tasks providing a smooth convex search space.

Simulated Annealing: This is a stochastic optimization procedure inspired by the thermal relaxation in solid state physics [14]. It can be applied to almost any optimization problem. The search domain is sampled more and more locally during the search process controlled by the annealing strategy. The performance of this procedure is strongly dependent on the kind of problem it is applied to.

Design of Experiments: Actually, this is not a real optimization method. A step size for each parameter has to be provided, hence defining a regular grid over the search space. The points of this grid are systematically explored and the corresponding parameter dependencies can be evaluated.

Finally, it is important to note that in the PEO workbench an optimization problem defined through a list of parameter values, an objective, and some constraints can be set up independently from the algorithm used. The optimization problem is then passed to an optimization engine evaluating the given problem appropriately, where own optimization engines can be implemented.

4 EAs for CAD based design optimization

Within this Section, it is explored how Evolutionary Algorithms can be adapted to take advantage of the available representations of mechanical structures in modern CAD systems. Therefore, two concepts to map a structure defined in the specification tree of CATIA V5 to a genotype of an EA are introduced.

4.1 Genotype of CAD parameters

A first possibility to utilize the CAD system for evolutionary design optimization consists of mapping all the parameters used to define the different features in the specification tree to a genotype. Standard evolutionary operators are then applied on this list of parameter genes, and new solutions (phenotypes) are simply created by updating the CAD structure. The optimization of a race car rim discussed in Section 5 represents an application for this approach.

4.2 Genotype of CAD features

EAs operate most effective, if the properties of the parents are directly inherited to the offspring (see [23] for a detailed discussion of this topic). With other words, individuals being similar in the genotype space should also be similar in the phenotype space. The plain parameter encoding approach described above does not guarantee such a correlation, since a component of the mechanical structure in the phenotype space is not necessarily represented by a single parameter in the genotype space. This reduces the correlation between the parents and their offspring and can harm the efficiency of the evolutionary operators and thus the efficiency of the whole optimization process.

In the following, we present a novel concept, which better accounts for the specific properties of the representation of mechanical structures in CAD systems. Considering the different information containers used by CATIA V5 to model a mechanical part as outlined in Section 3.1, the **Specification Container** and the **Geometrical Container** attract attention. They perfectly provide the necessary separation between genotype space and phenotype space. Consequently the goal must be to directly use them in the Evolutionary Algorithm. The real components of a mechanical structure, e.g. ribs or webs, are typically modeled as features in the specification tree of the CAD system. These features should be defined in a modular way correctly representing the structure of the design to be optimized. For optimization, a genotype consisting of a list of features can be built, where one feature-gene can consist of sub-features and is finally defined through a set of parameters. An example of such a gene is the `Groove` feature in Figure 2 which is defined through parameters defining its length, depth, and width. Therefore, such a genotype consists of a heterogeneous list of different features with a fixed order. For the implementation this genotype can also be seen as a heterogeneous list of grouped parameters, where every group denotes a feature in the CAD system. This perception is advantageous because finally there is again a simple list of parameters with fixed length and order to be handled by the EA.

To make the CAD-feature genotypes perform effective, it is crucial to introduce adequate evolutionary operators. In the following, crucial points concerning such operators suited for the handling of CAD-feature genes are discussed.

Mutation operators. Mutation operators should increase diversity in a population by random variation of single genes. For the CAD-feature genotype it makes therefore sense to apply mutation on CAD-feature genes rather than on parameter gene level. Therefore, if mutation is called for a single gene, a whole feature or substructure of the CAD model is mutated. This is superior to changing the same amount of information regardless to the CAD features. With mutation on CAD-feature gene level, a whole component of the structure is modified at once by an operator call, whereas mutating only one single parameter per call will often not effect correlated and substantial changes in the structure.

Crossover operators. The same thoughts as for the mutation operator basically also affect the design of adequate crossover operators, but now the inheritance of knowledge from the parents to their offspring is desired. This can be achieved efficiently by exchanging entire components of the structure using n-point type

crossover. If reasonable superpositions of CAD-features are defined, arithmetic-type crossovers can also be applied on CAD-feature level.

4.3 Knowledge-Based Initialization

Initialization operators using **knowledge-seeding** mechanisms can easily be applied when representing structures in CAD systems. Thereby, already existing solutions, typically based on some conceptual ideas of the designer, are incorporated in the CAD model by manually adjusting the defining parameters of the different features. The CAD system CATIA V5 even provides an own container named `DesignTable` to manage parameter sets of a specific CAD model. To seed these solutions into the start population for the evolutionary optimization, the parameter sets only have to be mapped to individuals, so that a single set represents one individual. In the majority of cases this knowledge seeding mechanisms accelerate the optimization process drastically, given that the designer's intuition provides at least some useful concepts.

4.4 Fitness evaluation

In order to complete the presented optimization concept, a fitness function $F(\vec{p})$ has to be defined giving an adequate rating for each possible design with respect to the demands. F directly depends on the phenotype $\vec{p}$, i.e. the CAD structure represented by the Geometrical Container in CATIA. Furthermore, the CATIA optimization workbench provides a reasoned interface to define the design objective and constraints by selecting arbitrary parameters defined in the model, see Section 3.3. We propose to define the fitness function to be a weighted sum

$$F(\vec{p}) = \sum_i w_i D_i(\vec{p}), \tag{1}$$

where $D_i(\vec{p})$ represents the rating for an objective or a specific constraint and w_i is the corresponding relative weight. In order to avoid that one of these terms becomes very large and therefore dominant, only bounded functions $D_i(\vec{p})$ are used that are scaled to the interval $[0, 1]$.

5 Parameter optimization of a race car rim

This application highlights how EAs can be successfully applied to a highly heterogeneous and complex structural optimization problem. The structure to optimize is heavily constrained through manifold requirements from regulatory, manufacturing, and functional demands. Therefore the optimization task is not to evolve a completely new rim design, but rather to adjust parameters of the existing rim structure, fully compliant with the requirements, in an optimal way. In that context, the application demonstrates how parametric-associative CAD systems can be integrated in automated optimization processes. Additionally, this complex problem presents a typical example where the parameterization has to be chosen carefully so that useful results can be obtained within available computation times.

5.1 Problem description

Motorcar racing is a field where limits are pushed in many ways. The wheels form via the tires the important link to the ground into which most acceleration forces are transferred. The performance of a racing car rim as shown in Figure 4 depends on several factors: weight should be as low as possible, the moment of inertia along the rotation axis has to be small, and the stiffness should be high at the same time. The **weight** of the rim does not only influence the performance as a part of the cars

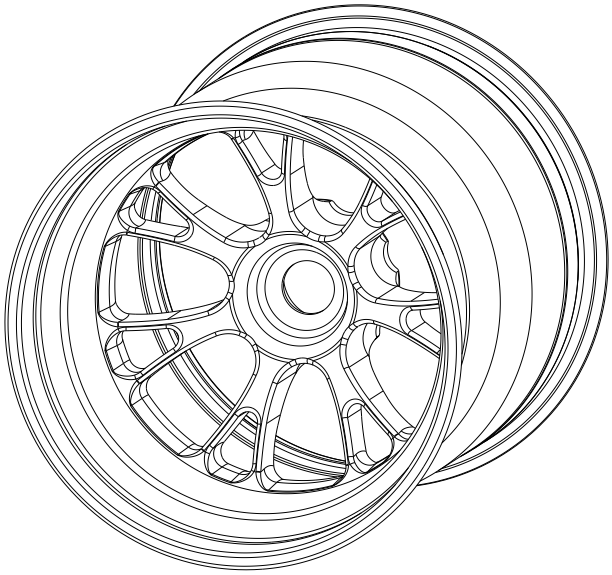


Figure 4: CAD-Model of the racing car rim.

overall weight. Since the wheels belong to the so called unsprung mass, a low rim weight improves the mechanical grip of the car especially on bumpy road surfaces. The **moment of inertia** along the wheels rotational axis should be minimal for several reasons. Low moments of inertia allow faster acceleration and deceleration of the wheels and therefore of the whole car. Furthermore, the moment of inertia leads through the gyro effect to higher steering forces as well as a higher inertia of the car with respect to direction changes. Finally, the **stiffness** of the rim is of high importance in turns at high speed. Vertical loads of $5700N$, resulting from the car's weight and aerodynamical descending forces, as well as maximum lateral forces of $7000N$, as a result of the centripetal forces, build up a bending moment on the rim. Additionally, **strength requirements** must be fulfilled. Plastic yield must not occur in use, whereas some parts reach temperatures well above $200°C$. Manufacturing starts with a forging blank, the rim's bed is shaped by CNC-lathe, and the spokes form the interspace of CNC-milled pockets. The forging blank's shape is not to be changed, as this would exceed costs. FIA[2] regulations affect the bead diameter as well as dimensions of the lower rim-bed.

For the optimization presented in this chapter, **maximum bending stiffness** is

[2]Fédération International d'Automobiles (http://www.fia.com)

defined as main design objective from the rim manufacturer. Nevertheless, the desired properties

- Low mass
- Low rotational moment of inertia
- Sufficient margin of safety for mechanical stresses

of the existing design should also be matched or even surpassed by the optimized design. Furthermore, the optimization must also take into account the functional, regulatory, and manufacturing requirements discussed earlier in this section. Altogether, this constitutes a highly constrained optimization problem, which can not be tackled with classical mathematical optimization techniques.

Since all original data is confidential, arbitrary new geometries for the rim and the forging blank, as well as modified load cases are created and optimized for the presentation in this paper

5.2 CAD model and parameterization

In the following, a parameterization of the rim structure implemented in the parametric associative CAD system CATIA V5 is presented. The goal is to implicitly include as many of the manifold requirements as possible into the parameterization without excluding relevant designs from the solution space. Moreover, the number of optimization variables allowed is determined through the available computational resources.

Considering computational limitations. EAs require a certain minimum number of evaluations. When the evaluation of one individual includes the creation of a new CAD-model followed by a Finite Element Analysis (FEA) on that model, CPU-time becomes a hard restriction. Measuring the time needed for one evaluation showed that it would be around 20 minutes consisting of roughly 5 minutes for the creation of the CAD-model, and about 15 minutes for the FEA (using a Sun Blade 1000, 750 MHz, 1 GB RAM). Being able to run 6 workstations in parallel during 20 hours limits the number of evaluations to about 360.

From experience, one can say that an optimization should run at least over about a dozen of generations to make the evolutionary strategies work on a level beyond pure stochastical search. Further, in order to obtain reasonable optimization performance (with respect to pure stochastical search), the number of parameters to be optimized should not exceed the population size. These rough assumptions limit the number of optimization variables for the actual task to 30. Therefore a design parameterization with only 30 parameters has to be found that is as general as possible.

Geometric constraints. There are three different classes of constraints geometrically restricting the solution space that shall be included implicitly into the CAD parameterization:

Manufacturing constraints. They include the shape of the forging blank which is not to be changed. Moreover a $1\,mm$ distance to the blank's outer contour has to be kept to allow properly machined surfaces. In addition, manufacturing

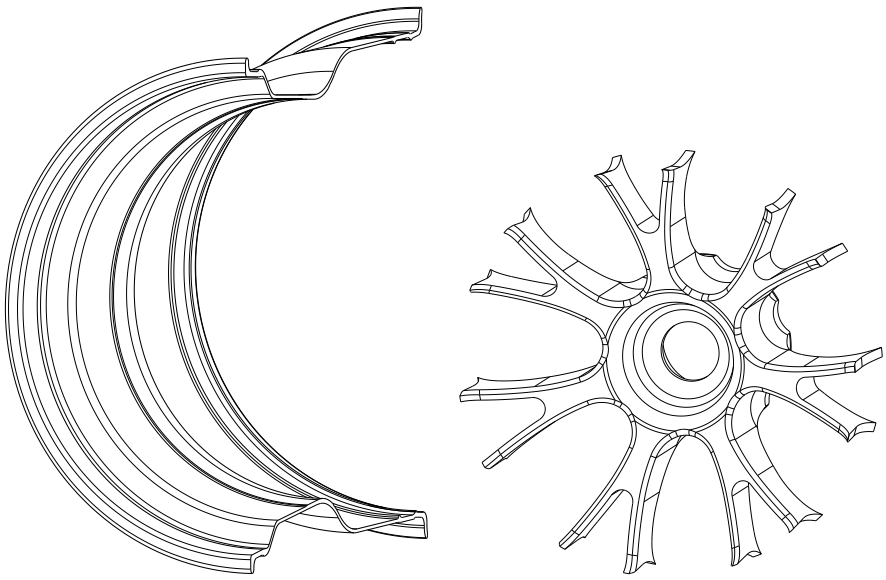
techniques are not to be changed. Finishing is restricted to a CNC-lathe and a CNC-mill. The minimum wall thickness of the rim's bed of $2\,mm$ is also a manufacturing constraint.

Assembly constraints. They include contact areas to the car's suspension as well as to the nut holding the wheel and the tire. An additional constraint is a $3\,mm$ distance to the brake assembly positioned on the inside of the rim.

Regulatory constraints. Finally, there are FIA-regulations applying to the rim. The maximum bed diameter is limited to $330\,mm$. The minimum depth of the lower bed[3] is $13.57\,mm$ and the maximum distance from the outside surface is $43.3\,mm$.

Structure of the CAD Model. Actually the CAD model used is quite simple. Four substructures (features) build up the basic geometry of the CAD model as shown in Figure 5:

- A rotational body for the rim's bed.
- A second rotational body for the spokes.
- Two pockets that remove the spokes' interspace.



(a) Half of the rim bed.

(b) Spoke body with the two pocket features already subtracted.

Figure 5: Structure of the CAD model of the rim.

All these features are built on fully parametric two-dimensional sketches. In addition to these basic features, various chamfers are applied, chosen completely parametric as well. The basic design is taken from an existing racing car rim.

[3]for definition of terms see [18], [17]

5.3 Parameterization

For performance reasons as outlined before, a parameterization including implicitly as many of the mentioned constraints as possible has to be found, without excluding relevant feasible solutions.

The bed is parameterized by 9 wall thicknesses including the inner and the outer bead as shown in Figure 6. With this approach the manufacturing constraint of a

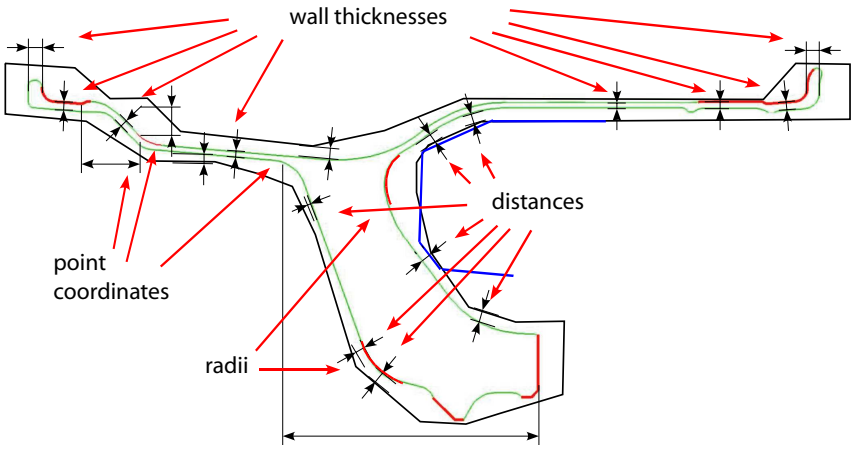


Figure 6: Optimization variables for spoke-body and bed contour.

minimum wall thickness of 2 mm can directly be included. The bed's outer contour can not be altered in some sections. The outer bead diameter and some shoulder diameters are restricted by FIA regulations. The inner bed's maximum diameter is limited to the shoulder's diameter to allow the assembly of the tire. The lower bed's minimum depth and the maximum distance from the rim's outer face are also subject to FIA regulations. Therefore the coordinates of the point in question also form parameters allowing to comply with those restrictions. For the remaining degrees of freedom of the bed, contour lines of the forging blank and the brake assembly limit the geometric design space. To make sure the minimum distances of 1 mm to the forging blank and 3 mm to the braking contour can be approached as closely as possible without crossing them, the contour lines form construction elements in the sketch and the bed's contour is directly dimensioned to those contours, setting the distances as parameters.

The rotational body for the spokes is parameterized in a way similar to the rim's bed. Front and rear contour are dimensioned to the blank's contours. The parts of the contour-forming interfaces to the suspension and the nut are non-parametric. The interface between spokes and bed depends on the shape of both features. The sketch for the bed contains the spokes' rear and front contour lines as construction elements. The second sketch for the rotational body of the spokes is referenced to those construction elements and the interface line with reference dimensions. This way, the parameterization for both features is done in only one sketch and update loops are avoided. At the same time, two separate features are needed, because the base part between the spokes is not defined by prismatic pockets but the contour of the bed. The pockets are removed from the spoke body going beyond the outer diameter. The remaining spokes are then added to the rim bed. The sharp angled base of the spokes is then smoothened out by two parametric fillet features.

The two pockets are parameterized in a third sketch (see Figure 7). A base line

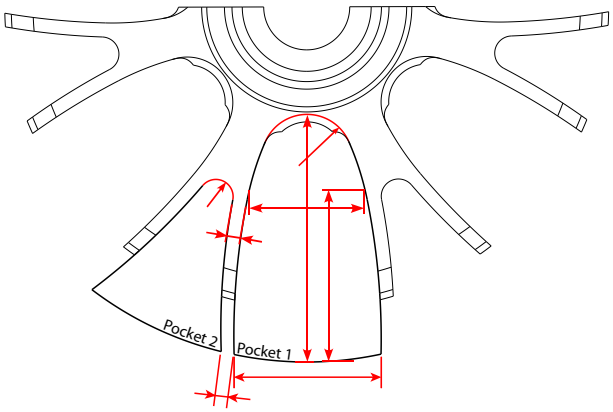


Figure 7: Optimization variables for the pockets.

with a constant radius is defined. Parameters for the larger pocket include height and width at the base as well as at a continuous transition point, and the radius of the upper rounding. The gap between the two pockets, forming the spoke, is parameterized by two widths, one at the base and one at the same transition point. This leaves only the upper rounding as a free parameter of the small pocket.

Altogether, the rim is parameterized with 36 float parameters. For every gene, adequate lower and upper limits are estimated by varying the parameters in the CAD model, and testing whether the update process in the CAD program still succeeds, and whether feasible solutions are created. However, the former is not realistic in practice for all possible variations in the defining parameter set. For a small percentage of evolved individuals the CAD update fails.

As a result, a CAD geometry for an individual at hand (expressed through the phenotype $\vec{p}$) is created, and its weight $W(\vec{p})$ and moment of inertia $I(\vec{p})$ are computed using CATIA built-in functionalities. The geometry is stored in CATIA V4 format, which can be imported to the Finite Element program ANSYS in order to carry out the Finite Element analysis. There the bending stiffness $D(\vec{p})$ and the margins of safety for mechanical stresses $S(\vec{p})$ are evaluated for the given loads. The effective fitness $F(\vec{p})$ of a single solution is then computed as a weighted sum of all these different demands.

5.4 DynOPS - Dynamic Optimization Parameter Substitution

DynOPS (Dynamic Optimization Parameters Substitution) is a parallelized and object-oriented evolutionary optimization program entirely written in C++ code. It has been developed at our chair, a detailed discussion can be found in [11]. The program concatenates simulation software controllable by ASCII input files over file transfer with Evolutionary Algorithms. As optimization engine, Evolutionary Algorithms based on the component-made EO (Evolving Objects) library are implemented. On the other side, the most common way to interact with commercial

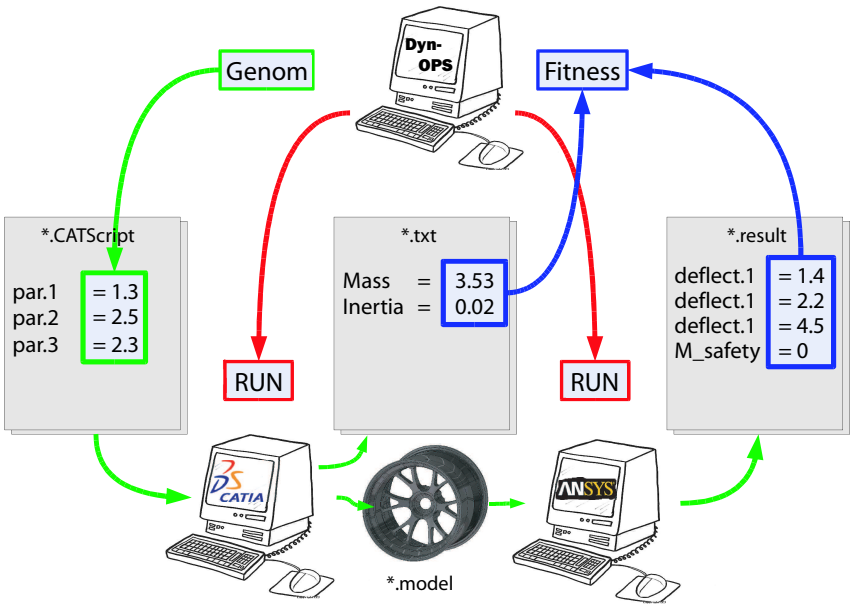


Figure 8: General functionality of the optimization process with DynOPS

(simulation) software is to use their batch mode, i.e. running simulations through input-files.

As indicated in Figure 8, DynOPS is, so to say, the master-mind within this optimization. It performs the following sub-steps of a run:

- Reading genetic information from the EA.
- Creating batch files for the external programs, one set per individual to be evaluated.
- Parallel distribution and start of the program sequence needed for evaluation.
- Waiting for the simulation programs to finish and reading of results of the evaluations.
- Passing results to EA.

In our case batch-file-types used are $*$.CATScript and $*$.cmd to run CATIA and ANSYS. Apart from parameters, DynOPS also handles names and paths of files that form interfaces between different programs, making sure no files are erroneously overwritten. This is especially important when running the optimization in parallel. Further on, DynOPS automatically assigns error fitness values to individuals that could not be evaluated in the external simulation programs. This occurs, especially in the beginning of an optimization, quite often if e.g. the CAD program can not evaluate a consistent 3D geometry from the parameter set given.

5.5 Results

As mentioned before, due to confidentiality reasons, only exemplary results for an optimization run based on modified geometries and loads can be shown in this public document. The Evolutionary Algorithm is run for 15 generations with a population

Table 2: Comparison of objective values of optimized and original design.

Properties	Original	Optimized	Improvement
Bending compliance [deg]	0.257	0.206	19.8%
Mass [kg]	3.51	3.27	6.8%
Inertia [$\mathrm{kg} \cdot \mathrm{m}^2$]	0.063	0.059	6.3%

size of 30. This corresponds to an overall CPU-time of approximately 150 hours or 25 hours when running 6 workstations in parallel.

Figure 9 shows the fitness evolution of the best individual over the 15 generations. In Table 2 the objective values of the optimized rim are compared with the ones

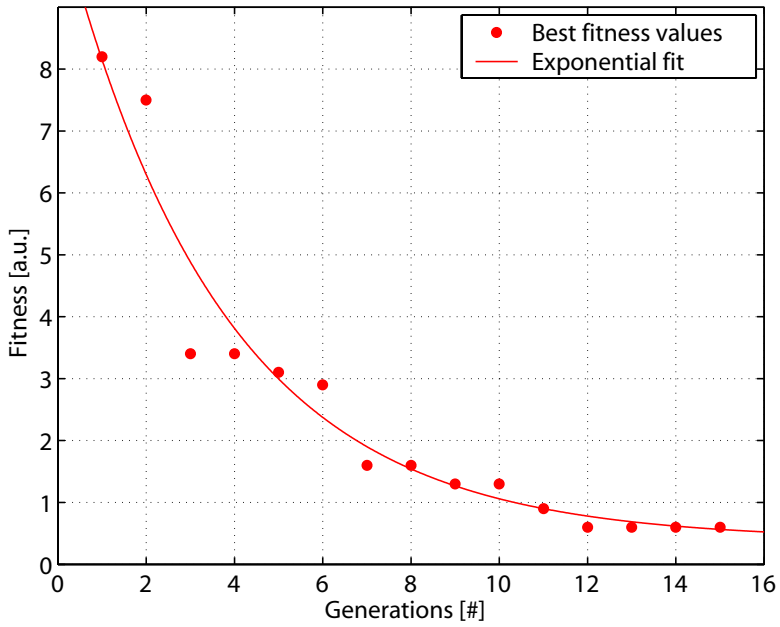
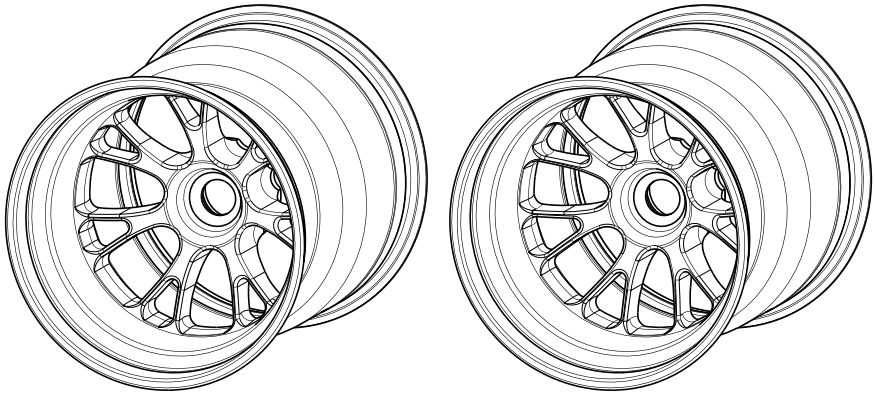


Figure 9: Progress of best fitness values for the rim optimization.

of the original design. Although having evaluated only a relative small number of individuals and generational steps, clear improvements can be noticed in all aspects considered within fitness calculation. The main design objective *bending stiffness* is improved by 19.8% and all the constraints are fulfilled. These significant improvements are quite surprising when considering that only slight changes in the design can be noticed in Figure 10 and 11. Most of these design changes can hardly be motivated by pure engineering intuition and therefore indicate the potential of automated optimization algorithms in high-performance applications where it is necessary to approach the limits very closely.



(a) original (b) optimized

Figure 10: Original and optimized rim design.

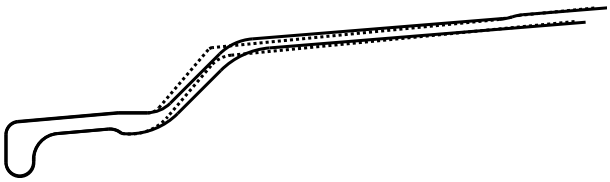


Figure 11: Lower bed shape: optimized and original (dashed).

6 CAD-feature based optimization of a fuel cell end plate

This problem sample applies the concepts regarding evolutionary design optimization with a CAD-feature genotype as proposed in Section 4.2. The novel approach is verified and compared with a CATIA native optimization engine based on Simulated Annealing (SA). Other experiments will show the superior performance of CAD-feature driven EAs in comparison with plain parameter representations. For computational reasons, these experiments are processed with a verification model. It is based on the same CAD-model and the same genotype as the fuel-cell end plate, but the optimization goal is chosen differently to speed up fitness evaluation. Only this simplification allows proper performance comparisons of the algorithms taking into account the stochastical nature of EAs and Simulated Annealing.

However, the potential of the approach will finally be demonstrated by solving the engineering problem of minimizing the weight of an end plate of a fuel cell stack under strength constraints.

6.1 Problem description

The weight of an end plate of a fuel cell stack shall be minimized. The problem arose in fuel cell research projects[4] [19, 20] at the Swiss Federal Institute of Tech-

[4]http://www.powerpac.ethz.ch

nology Zurich. The end plates for the fuel cell stacks developed in these projects were designed by Tribecraft AG[5] [5], which also provided the detailed problem description for the optimization presented in this section. A fuel cell stack, as shown in Figure 12, consists of bipolar plates, two collectors, electrical insulation, and the end plates that are connected with tie bolts. The fuel and cooling supply line runs through the upper end plate. The weight of such an end plate is minimized sub-

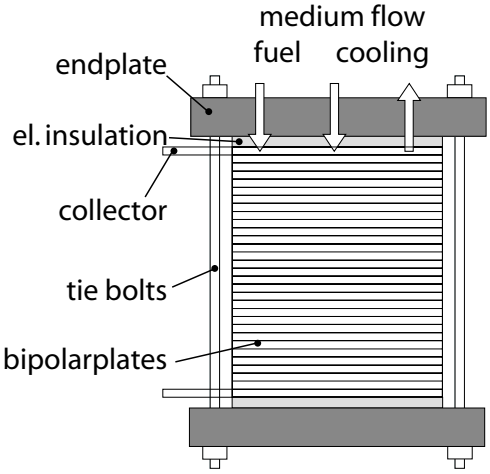


Figure 12: Conceptual model of a fuel cell stack.

ject to a stress constraint and manufacturing requirements, whereas the structure is foreseen to be manufactured by extrusion molding. The objective is to increase the power density, i.e. power per weight, of a fuel cell stack. In a second optimization procedure, described in [5], the bottom surface of the end plate can be cambered independently to guarantee a constant pressure distribution on the fuel cell stack in built-in state. This sequential partitioning of the optimization problem allows to first minimize the weight of the plate under a stress constraint without concern for the stiffness of the plate.

For the verification model mentioned above, the same CAD-model and genotype is used to optimize a bridge-like structure. The objective of this optimization is to minimize its compliance subject to a constant mass given in percent of the fully-filled design domain.

6.2 Mechanical structures to be optimized

Verification model. The mechanical structure to be optimized with a CAD-based representation is a bridge-like 3D girder. Figure 13 illustrates the design domain in which the structure can evolve (indicated by dashed lines), the mechanical boundary conditions, and the CAD model itself. The relevant mechanical data are specified in Table 3. The girder shall be a symmetric and extrudable rib structure assembled through a lower and an upper plate that is loaded by a constant pressure f at the bottom of the lower plate. A support at the right end and the symmetry conditions in the middle complete the mechanical model. For the results presented, the structure shall fill $f_{tm} = 20\%$ of the design domain with material.

[5] http://www.tribrecraft.ch

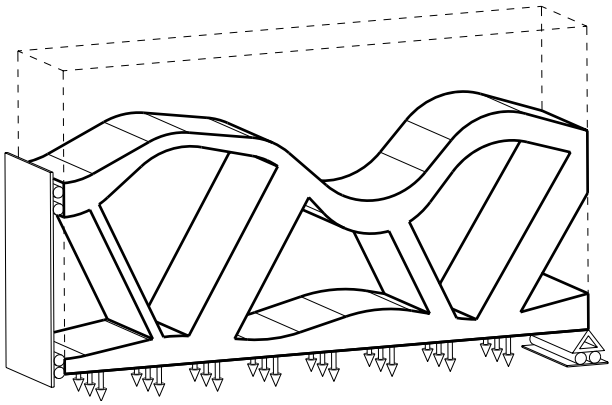


Figure 13: CAD model of the bridge-like structure with mechanical boundary conditions and given design domain.

Table 3: Specifications of the verification model.

Parameter	Denotation	Value	Unit
Dimensions	$l \times w \times h$	$73.7 \times 15 \times 40$	$[mm]$
Pressure	f	$1.44 \cdot 10^6$	$[\frac{N}{m^2}]$
Elastic modulus	E	$7 \cdot 10^{10}$	$[\frac{N}{m^2}]$
Material density	ρ	2710	$[\frac{kg}{m^3}]$
Target mass fraction	f_{tm}	20	$[\%]$

End plate of a fuel cell stack. Figure 14 shows a quarter model of the end plate with mechanical boundary conditions. The relevant specifications for this structure are given in Table 4. The end plate is loaded on the bottom surface by

Table 4: Specifications of the fuel cell end plate.

Parameter	Denotation	Value	Unit
Dimensions	$l \times w \times h$	$166 \times 147.4 \times 40$	$[mm]$
Stack pressure	p_s	$1.6 \cdot 10^6$	$[\frac{N}{m^2}]$
Bolt force	F_b	8800	$[N]$
Elastic modulus	E	$7 \cdot 10^{10}$	$[\frac{N}{m^2}]$
Material density	ρ	2710	$[\frac{kg}{m^3}]$
Stress limit	σ_{max}	$3 \cdot 10^8$	$[\frac{N}{m^2}]$

the given stack pressure p_s. The reaction forces, hold through four tension bolts, are also applied as forces on the bolt contact faces. This ensures a proper load introduction into the structure and does not restrict the rotation of the bolt areas. In addition, symmetry conditions are applied, and the end plate is statically definite restrained to prevent rigid-body motions.

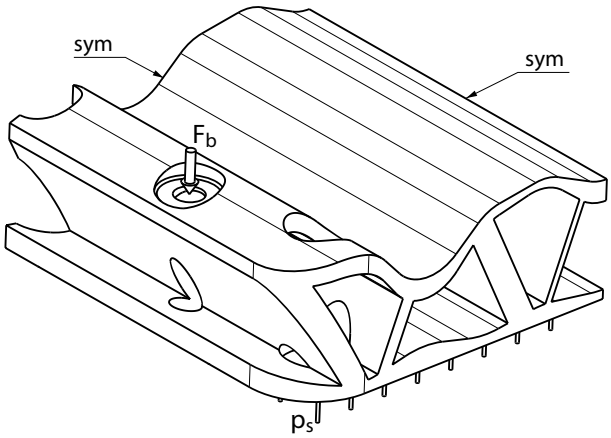


Figure 14: Quarter model of fuel cell end plate with mechanical boundary conditions.

6.3 CAD build-up and feature genotype

The CAD model is assembled from three types of CAD features: the lower plate, the upper plate, and four ribs. Figure 15 details how these entities are defined. The lower plate is bounded through a planar functional face at the bottom and through assembled face segments at the top. These segments are defined through equidistant sampling points defining six optimization variables $t_{l1}...t_{l6}$ of the lower plate. Additionally, the edges of the top faces are chamfered. The upper plate is defined through two sets of sampling points, i.e. six equidistant height parameters $h_{u1}...h_{u6}$ defining the bottom face and six thickness variables $t_{u1}...t_{u6}$ defining the top face of the upper plate. Again, the edges of these faces are chamfered. Finally, each rib is defined through three optimization variables: a lower position x_{li}, an upper position x_{ui}, and a thickness t_{ri}. For all these optimization parameters a range and a step size σ for Gaussian mutation or initialization are assigned as listed in Table 5. The genotype for the Evolutionary Algorithm is defined as follows. The ribs are

Table 5: Ranges and step sizes for the optimization variables.

Parameters	Range [mm]	Step [mm]
$t_{l1}...t_{l6}$	$[1, 7]$	1
$h_{u1}...h_{u6}$	$[1, 33]$	4
$t_{u1}...t_{u6}$	$[1, 7]$	1
$x_{l1}...x_{l4}$	$[0.1, 73.3]$	8
$x_{u1}...x_{u4}$	$[0.1, 73.3]$	8
$t_{r1}...t_{r4}$	$[1, 7]$	1

chosen as CAD features to be optimized; one rib is defined by three parameters that are represented in one gene. For the lower and upper plate sampling positions are chosen as genes. That means for the lower plate each thickness represents a gene, and for the upper plate the height and the thickness at a sampling position form a gene. This leads to the following genotype, where each gene is marked through

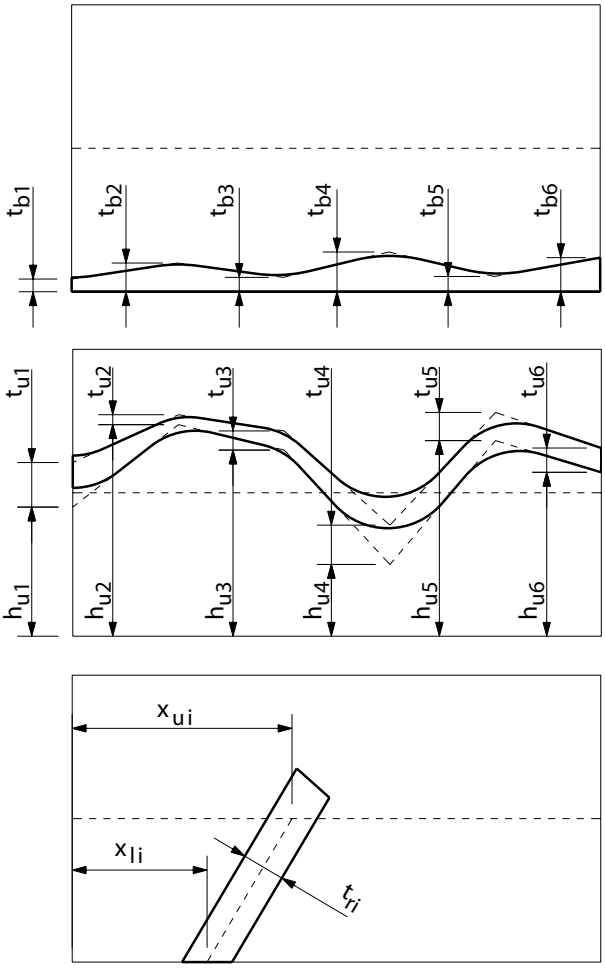


Figure 15: CAD features: lower plate, upper plate and a rib.

accolades:

$\{t_{l1}\}$ $\{h_{u1}, t_{u1}\}$ $\{t_{l2}\}$ $\{h_{u2}, t_{u2}\}$ $\{t_{l3}\}$ $\{h_{u3}, t_{u3}\}$ $\{t_{l4}\}$ $\{h_{u4}, t_{u4}\}$ $\{t_{l5}\}$ $\{h_{u5}, t_{u5}\}$ $\{t_{l6}\}$ $\{h_{u6}, t_{u6}\}$ $\{x_{l1}, x_{u1}, t_{r1}\}$ $\{x_{l2}, x_{u2}, t_{r2}\}$ $\{x_{l3}, x_{u3}, t_{r3}\}$ $\{x_{l4}, x_{u4}, t_{r4}\}$.

This representation implicitly fulfills the manufacturing requirement, i.e. extrusion molding. For the end plate, the holes for medium flow and the tension bolts are made in a subsequent machining process, and the global vertical edges are rounded. Since material can only be removed from the extruded base block, planar horizontal faces have to be machined into the upper plate for a well defined load introduction from the bolts to the end plate. The position of these contact faces must be adapted to the varying slope and curvature of the top face of the upper plate, whereas for some solutions this face even can be split in two subregions.

6.4 Fitness evaluation

Fitness functions are again computed as weighted sum of the different demands.

Fitness of the bridge-like verification model. Fitness values are computed from the compliance design objective and a mass penalty. If subject to fixed loads, the compliance of a mechanical structure is proportional to its elastic energy as discussed [3, 2]. In the Finite Element workbench of CATIA the elastic energy can directly be accessed through a sensor. The mass constraint is implemented as upper limit constraint.

Fitness of the fuel cell end plate. For the end plate, the design objective is to minimize its mass. The maximum von Mises stress needed for the penalty term of the fitness function is captured by a sensor in the Finite Element workbench of CATIA. Computation of stresses using the Finite Element method is sensible to the fineness of the mesh and to a proper load introduction. Thus, a relatively fine mesh is defined using 10-node tetrahedrons with quadratic form functions. In addition, the load introduction of the bolt forces is done using the *Smooth Virtual Part* feature of CATIA which smoothly introduces a concentrated load on an area using beam spiders.

6.5 Integration of EAs into CATIA using CAA V5

A proper implementation of the CAD-feature genotype used within this application requires direct access to the CAD representation of mechanical structures. Therefore, the C++ Component Application Architecture[6] (CAA V5) offered by CATIA V5 is used.

The Product Engineering Optimizer (PEO) workbench in CATIA (see Section 3.3) provides a very convenient graphical interface to set up a design optimization. Therefore it is self-evident to directly implement the new optimization strategy in this prepared environment. A design optimization problem, interactively

[6] http://www.caav5.com

defined through the user by specifying optimization variables, an optimization objective, and constraints, is then transferred to an EA-type formulation. Figure 16 pictures the integration of the EA into CATIA. The design objective and the con-

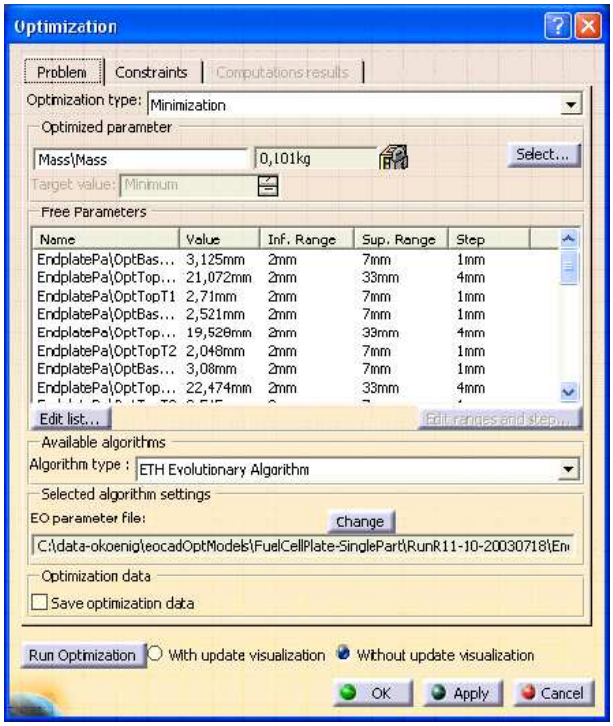


Figure 16: Integration of the generic EA into CATIA's optimization workbench.

straints, also specified in the graphical interface by the user, are directly mapped to a fitness function.

With the interface presented, setting up an evolutionary design optimization becomes possible for standard users of CAD systems without specialized 'evolutionary' know-how.

6.6 Demonstration of an optimization run

For all evolutionary optimizations of the verification problem the population size is set to $n_{pop} = 60$ and the runs are terminated after $n_{gen} = 120$ generations. Start populations for this structure are initialized randomly, typically resulting in structures as shown in Figure 17 with an average mechanical compliance of $C = 2.22\ J$ and an average mass of $m = 0.034\ kg$. The evolutionary process over 120 generations can be tracked in Figure 18. Since the currently best solution is kept in the population through an elitism mechanism, the best fitness values are monotonically decreasing, whereas the average fitness values of the populations sometimes raise temporarily in order to escape local minima or because of single error-fitness values. The final population after 120 generations is presented in Figure 19. The individuals of the final generation have an average compliance of $C = 0.099\ J$ and an average mass of $m = 0.025\ kg$ complying with the target mass fraction of 20%. Finally, the best individual ever found ($C = 0.0947J$, $m = 0.024\ kg$) is pictured in Figure 20. It

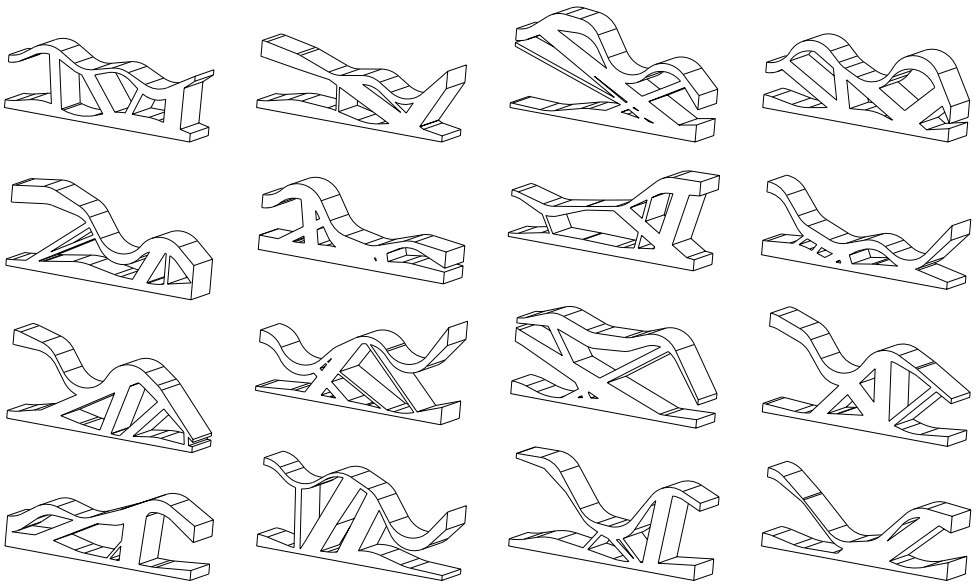


Figure 17: Sorted variety of individuals in random initial population.

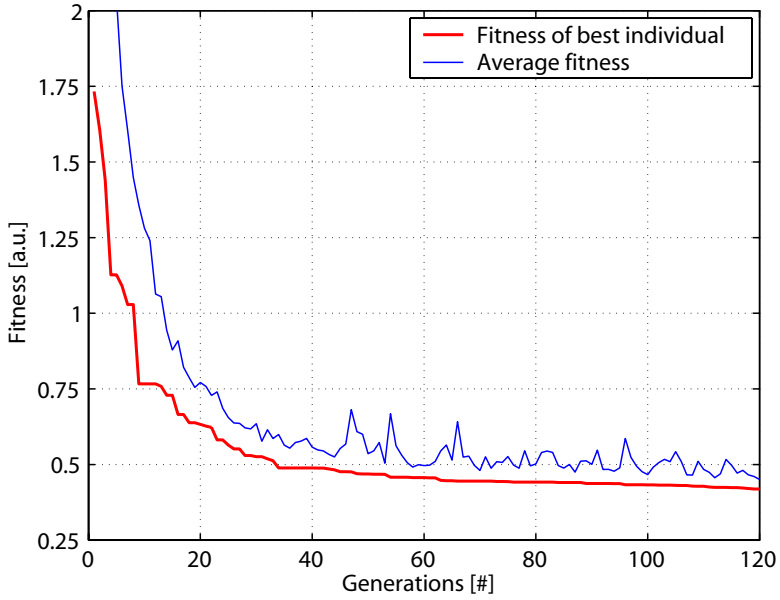


Figure 18: Average and best fitness progress over 120 generations.

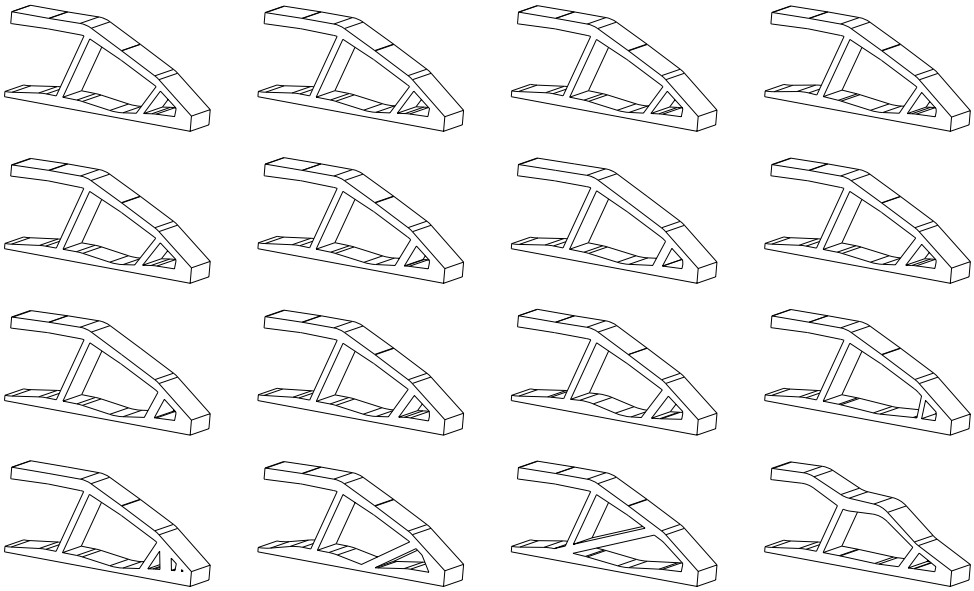


Figure 19: Sorted variety of individuals in generation 120.

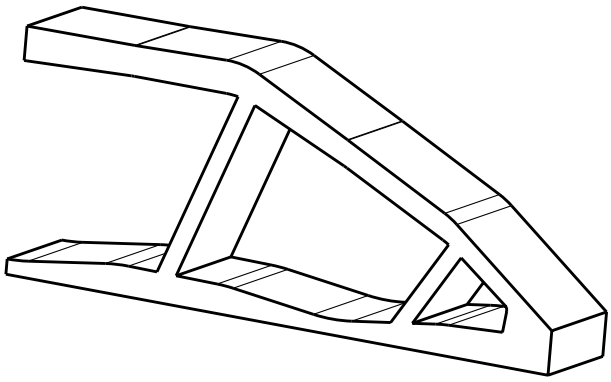


Figure 20: Best individual ever found after 120 generations

definitely matches with the intuitive expectations of a bridge design. Only two of the four ribs given in the original representation survived as independent features, whereas the other two ribs reinforce the right edge of the design. Generally, the application of the CAD-feature approach on the verification problem produces very appealing results.

6.7 Comparison with a Simulated Annealing algorithm

In order to demonstrate the performance of the CAD-feature driven EA, a comparison with the Simulated Annealing algorithm natively implemented in the Product Engineering Optimizer of CATIA V5, see Section 3.3, is presented. The comparison is based on seven identical runs of each algorithm in order to take into account the stochastic nature of Evolutionary Algorithms as well as of Simulated Annealing strategies. First, seven EA optimizations of the verification model are run, identically to the one presented in Section 6.6 with a random initial population. For the SA optimization runs, the same problem set-up in the PEO can be used, consisting of the bounded parameters with step sizes, the compliance objective, and the mass constraint. Since the SA algorithm starts from a single design solution, the seven SA runs are started from the best individuals of the initial populations of the seven EA runs. Furthermore, the SA convergence speed setting *infinite* in CATIA is chosen, because it provides the best results for the problem at hand.

Figure 21 presents the averaged best fitness values from 700 evaluations for both algorithms, i.e. approximately twelve generations when using EAs. As already ex-

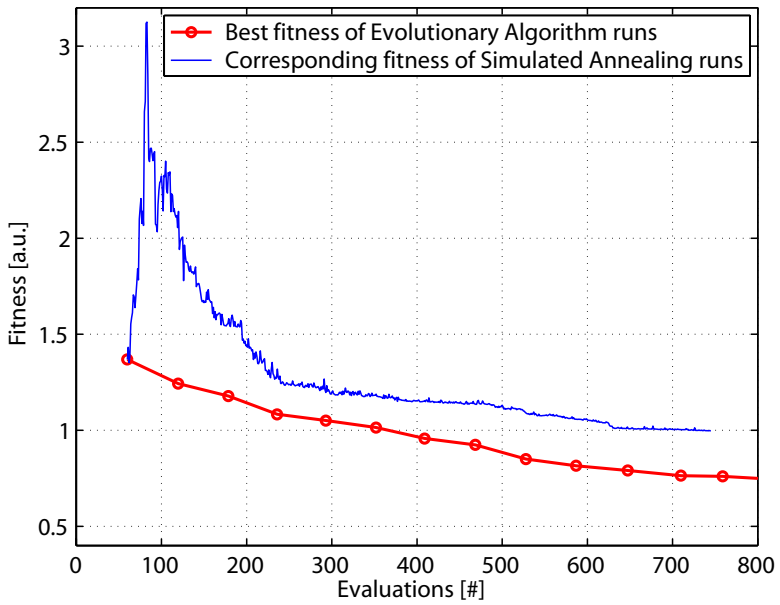


Figure 21: Comparison of averaged best fitness values of seven EA and seven SA optimizations.

plained, both curves start after 60 evaluations from the random initial populations or from the best individuals of this initial pool of solutions, respectively. Optimization runs with Simulated Annealing crash shortly after 700 evaluations due to a

memory leak in CATIA, hence limiting the comparison to this range. For Evolutionary Algorithm runs the memory leak could be bypassed by restarting CATIA and the optimization after every fifth generation. The increase of fitness at the beginning of the SA runs seems to be caused by an initial overestimation of the compliance objective resulting in too heavy designs. Since details of the CATIA SA implementation are not known, this aspect can not be further investigated. Nevertheless, Figure 21 shows that the CAD-entity based Evolutionary Algorithm approach clearly outperforms the CATIA native Simulated Annealing concept for the problem at hand.

6.8 Convergence speed of CAD-feature driven EAs

As a last verification experiment, EA runs with the CAD-feature genotype are compared to EA runs with simple parameter based representation. Results of the CAD-feature genotype with custom EA operators are taken again from the seven optimization runs used in Section 6.7. Then, a genotype consisting of a simple list of the optimization parameters as listed in Table 5 is built. Seven EA optimization runs with this genotype and standard operators with adequate settings are finally evaluated to produce averaged fitness results of the parameter based representation.

Figure 22 illustrates how the different genotypes affect the performance of the Evolutionary Algorithm. Since random initialization is identical for both, the fitness values coincide for the initial population. Then, for the first stage of the optimization until approximately generation number 50, the plots reveal that the CAD-feature based runs perform better. This can be explained by the fact that the global placement and adjustment of the different features in the structure is done in the beginning of the optimization. In fact, the expected increase in efficiency using CAD-feature genotypes arises in this phase of the optimization through a faster decay of fitness values. In a second stage of the optimization starting from generation number 50, the fitness of the structures can only be gradually improved by fine tuning single parameters as for example the thickness of a rib. Thereby, the genotype seems not to influence the performance of the algorithm anymore, since the fitness values are similar for both representations. Looking at the entire optimization process, the observed increase in performance is rather small, because the problem at hand consists of only a few CAD-features, whereof some ribs even disappear during the optimization. For more complex structures it can be expected that the presented CAD-feature genotype improves the efficiency of EA runs more significantly.

6.9 Results for the fuel cell end plate

The optimization is run with a population size of $n_{pop} = 60$ over $n_{gen} = 135$ generations, corresponding to approximately 8100 CAD updates and FEM analyses. Such a run takes about 65 hours computing time, made possible by restarting CATIA and the optimization each third generation because of the memory leak in the CAD system.

Figure 23 shows the progression of the mass of the best individual over all generations resulting in a weight of $m = 0.385kg$ for the best individual ever found. The best design ever found is presented in Figure 24. The Finite Element analysis

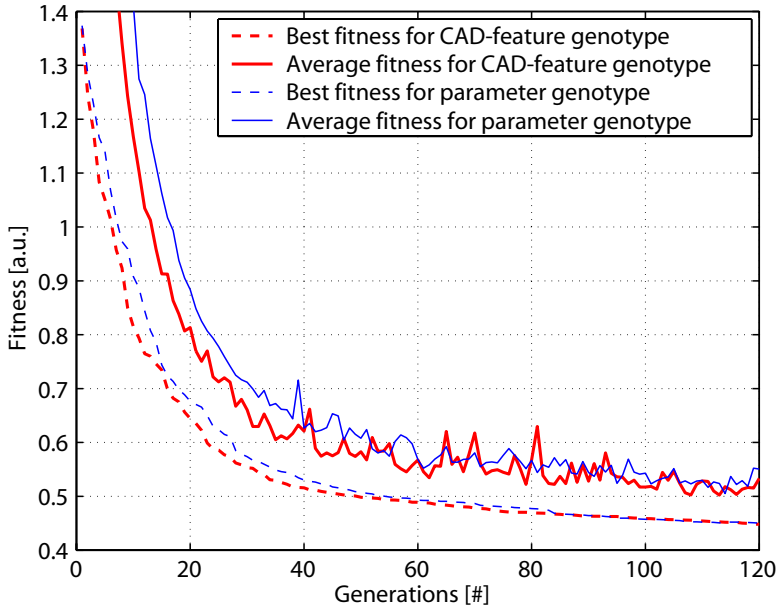


Figure 22: Comparison of averaged fitness values of seven EA runs each with CAD-feature genotypes and with plain parameter genotypes.

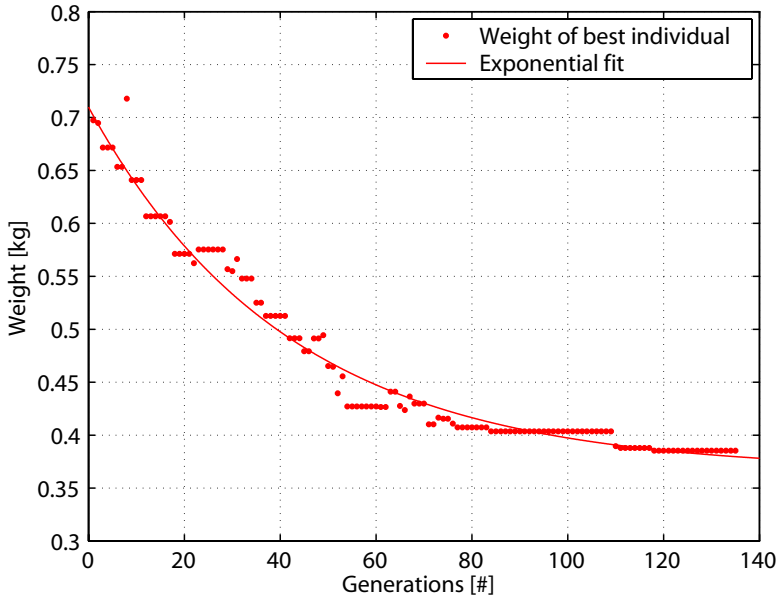


Figure 23: Progress of weight of the fuel cell end plate over 135 generations.

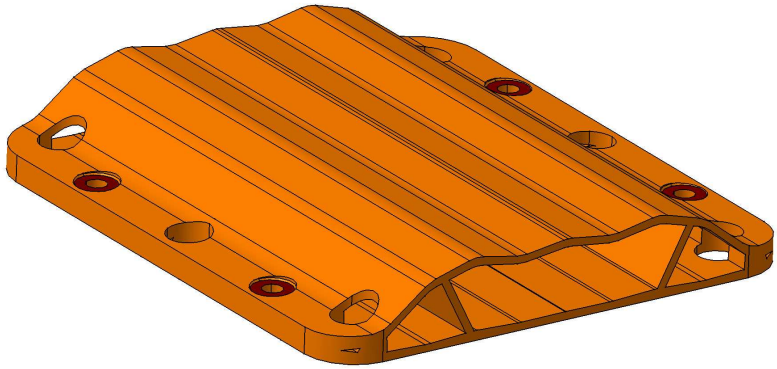


Figure 24: Best end plate design ever found after 135 generations.

presented in Figure 25 is evaluated with a finer discretization than used for the optimization. However, the stress constraint is observed very well, apart from a few

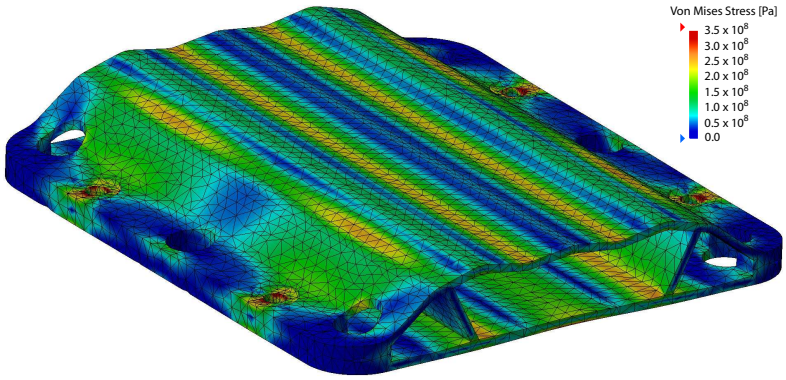


Figure 25: Von-Mises stresses of best plate design ever found. The plot also shows true scale deformations of the loaded structure.

FE introduced singularities at the bolt load introduction faces.

In the resulting structure, only one independent rib survives, the other three are melted into the load introduction domain. It is noteworthy that this load introduction zone is reduced to a fully filled block. It seems that only this way the occurring stresses can be kept under the given limit of σ_{max}. Furthermore, a corrugated upper plate is proposed by the optimization. Comparisons with an endplate with a planar upper plate showed that this stiffness reduction in the upper plate decreases the stresses over the upper end points of the ribs, and is therefore mandatory for this design.

Recapitulating, one can say, that the CAD-feature optimization of the fuel cell end plate resulted in a very lightweight design that fulfills the strength constraint and is manufacturable by extrusion molding. For a structure ready for production, the cambering of the bottom face has to be done and a potential influence on the stress level in the lower plate should be investigated.

7 Conclusions

We efficiently adapt an existing CAD-system as a parameterization interface for its use with Evolutionary Algorithms.

The parameter optimization of a race car rim demonstrates that EAs can tackle complex and highly-constrained problems where only a very limited number of evaluations is affordable. Starting from a rim, well designed through engineering experience, the optimization has successfully exploited the limits of this structure by adjusting 36 parameters of the geometry. Experience and know-how developed in years is therefore not thrown over board but forms the foundation of a new dimension of what has been engineering everyday life ever since: Trial and Error.

The novel approach based on a CAD-feature genotype is verified on a minimum-compliance problem leading to very appealing results. The superior performance of this CAD feature driven EA, if compared to the CAD system's natively included Simulated Annealing optimizer, is proven. Furthermore, a comparison between parameter-based and CAD-feature driven EAs showed slight advantages for the novel concept. Finally, the weight minimization of an end plate of a fuel cell stack under strength constraints is addressed, yielding surprising results.

Overall, the approach to implement EAs directly based on the representation of mechanical structures in CAD systems shows great potential to exploit the limits of modern products far beyond human intuition.

Acknowledgments

This work was supported by the Swiss National Foundation (SNF-project 21-66879.01).

References

[1] Th. Bäck. *Evolutionary algorithms in theory and practice.* Oxford University Press, 1995.

[2] M.P. Bendsøe. *Optimization of structural topology, shape, and material.* Springer-Verlag, Berlin, Heidelberg, New York, 1995.

[3] M.P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71:197–224, 1988.

[4] Peter J. Bentley, editor. *Evolutionary design by computers.* Morgan Kaufmann Publishers, Inc, San Francisco California, 1999.

[5] J. Evertz and M. Günthart. Structural concepts for lightweights and costeffective end plates for fuel cell stacks. In *2nd European PEFC Forum,Lucerne,Switzerland*, 2003.

[6] R. Fletcher and C.M. Reeves. Function minimization by conjugate gradients. *Comp, J.*, 7:149–154, 1964.

[7] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial intelligence through simulated evolution.* John Wiley, New York, 1966.

[8] D.E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1989.

[9] P. Hajela. Stochastic search in structural optimization: Genetic algorithms and simulated annealing. In *Structural Optimization: Status and Promise*, volume 150, pages 611–637. Progress in Astronautics and Aeronautics, 1992.

[10] J.H. Holland. *Adaptation in natural and artificial systems.* University Michigen Press, Ann Arbor, MI, 1975.

[11] O. König. *Evolutionary Design Optimization: Tools and Applications.* PhD thesis, Swiss Federal Institute of Technology, Zürich, 2004. Diss. ETH No. 15486.

[12] J. Koza. *Genetic programming: On the programming of computers by means of natural selection.* MIT Press, Cambridge, MA, 1992.

[13] M.J.D Powell. Nonconvex minimization calculations and the conjugate gradient method. In D.F. Griffiths, editor, *Lecture Notes in Mathematics*, volume 1066, pages 122–141. Springer-Verlag, New York, 1984.

[14] R.E. Randelman and G.S. Grest. Optimization by simulated annealings. *J. Stat. Phys.*, 45:885–890, 1986.

[15] I. Rechenberg. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution.* Fromann-Holzboog, Stuttgart-Bad Cannstatt, 1973.

[16] I. Rechenberg. *Evolutionsstrategie '94.* Fromann-Holzboog, Stuttgart-Bad Cannstatt, 1994.

[17] J. Reimpell. *Reifen und Räder.* Vogel Buchverlag, Würzburg, 1988.

[18] J. Reimpell. *Fahrwerktechnik Grundlagen.* Vogel Buchverlag, Würzburg, 1995.

[19] Martin Ruge. *Entwicklung eines flüssigkeitsgekühlten Polymer-Elektrolyt-Membran-Brennstoffzellenstapels mit einer Leistung von 6,5kW.* PhD thesis, ETH Zurich, 2003. VDI Verlag GmbH, Reihe 6, Nr. 494, Fortschritt-Berichte VDI; Diss. ETH No. 14901.

[20] Daniel Schmid. *Entwicklung eines Brennstoffzellenstapels fuer portable Aggregate unterschiedlicher Leistungsbereiche.* PhD thesis, ETH Zurich, 2003. VDI Verlag GmbH, Reihe 6, Nr. 500, Fortschritt-Berichte VDI; Diss. ETH No. 15066.

[21] Marc Schoenauer and Zbigniew Michalewicz. Evolutionary computation: an introduction. *Control and Cybernetics*, 26(3):307–338, 1997.

[22] H.-P. Schwefel. *Evolutionsstrategie und numerische Optimierung.* PhD thesis, TU Berlin, 1975.

[23] Marc Wintermantel. *Design-Encoding for Evolutionary Algorithms in the Field of Structural Optimization.* PhD thesis, Swiss Federal Institute of Technology, Zürich, 2003. Diss. ETH No. 15323.