Search for alternative car concepts with OptiSLang

Dr.-Ing. Johannes Will, Dipl-Ing. Jörg Riedel

DYNARDO GmbH, Germany, Weimar

Prof. Dr.-techn. Christian Bucher

Bauhaus Universität/DYNARDO GmbH, Germany, Weimar

Dipl.-Ing. Ingo Raasch

BMW AG, Germany, München

Summary

During the preliminary design phase, the car body is to be optimized. The demands on the car body are formulated by numerous performance constraints derived from crash, strength or driving comfort requirements. At the beginning of an optimization procedure, often one or more design drafts are available representing the previous knowledge from known functional designs. These designs are being parameterized and by this become utilizable for parameter optimization. In this example, the design space is described by an average of 1500 design variables. Because of very restrictive performance demands and numerous construction constraints, only extremely small areas of admissible and, regarding the weight, promising car dimensioning exists in the 1500 dimensional design space.

It is of high interest to find out whether there exist any competitive car concepts in the design space that lie beyond known car concepts or load paths, respectively. In this case, the optimization task does not only consist of an weight optimization of known car concepts, but includes the search for alternative vehicle concepts. In the here presented example, the search for reliable vehicle dimensioning has been successful thanks to special sampling strategies followed by genetic optimization algorithms. Subsequently, it has been investigated by means of cluster analyses whether there are vehicle concepts representing alternative complete vehicle concepts, and appropriate starting points for local optimization were determined. For selected designs, up to twenty clusters have been searched for weight optimal variants by means of mathematical optimization.

Due to the combination of genetic strategies for the search for admissible vehicle dimensioning and mathematical optimization for weight optimization on the islands of admissible vehicle dimensioning, the weight could be significantly further reduced then by mathematical optimization strategies only.

Keywords

multidisciplinary optimization, genetic algorithms, mathematical optimization, cluster analysis

0. Introduction

It is of high interest to find out whether there exist any competitive car concepts in the design space that lie beyond known car concepts or load paths, respectively. In this case, the optimization task does not only consist of an weight optimization of known car concepts, but includes the search for alternative vehicle concepts. A thus formulated optimization task often leads to mathematically ill-conditioned problems. Weight optimization and the observance of many constraints are often quite contradictory. Therefore, the optimization task is subdivided into the identification of islands of admissible designs (or, if the case may be, alternative vehicle concepts) on one hand, and the weight optimization on the islands, on the other hand. Genetic strategies are especially applicable for the task of island search. Genetic strategies are charecterized by a high robustness in mathematically ill-conditioned optimization tasks. Additionally, previous construction knowledge can be integrated on various levels. Provided that the selection, mutation and reproduction mechanisms are chosen appropriately, genetic search strategies can search large spaces and do not finish the search in the first local optimum. Once islands of admissible designs are found and appropriate gradients can be determined, mathematical optimization methods are highly efficient for weight optimization on such an island.

All structural mechanics calculations are performed with NASTRAN. The mathematical Optimization by means of gradient solvers are performed using NASTRAN SOL200. The genetic optimization as well as the cluster analysis are performed using OptiSLang.

1. Optimization during the preliminary design

During the early design state, the design spaces are to be investigated searching for alternative vehicle concepts. Additionally, the full vehicle models are to be weight optimized. In the course of these investigations, several important performance constraints regarding NVH, stiffness or crashworthiness have to be observed. This is realized by investigating eight static load cases (in order to analyze stiffness and crash behaviour) as well as several dynamical loadcases containing modal analyses and frequency analyses. Additionally, more than three thousand construction constraints have to be met, such as sheet thickness skips, height/length ratios, or constructed space. A total of 1544 design variables can be modified in the full vehicle model. These are e.g. profile measures (height/width) or sheet thicknesses. All variables are regarded as continuous.

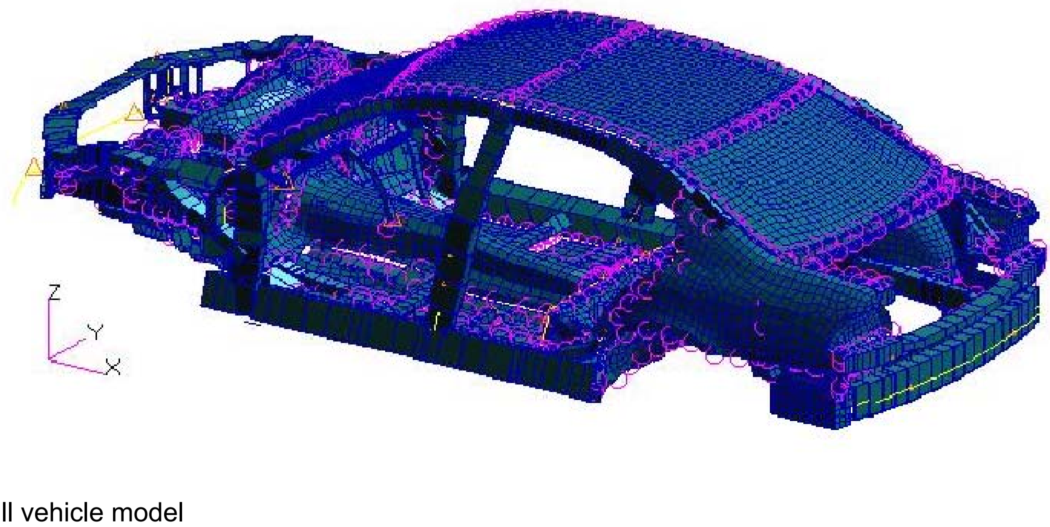

Fig. 1 Full vehicle model

The classical conflict in optimization tasks of this kind is that the aim of weight minimization and the observance of the constraints are somewhat contradictory. Additionally, it is to be assumed that all initial designs lie within the inadmissible space, while only very small admissible areas do exist in the high dimensional design space that are both interesting regarding the weight and fulfilling the performance constraints. The existence of multiple “islands” of admissible designs representing several basic variants of admissible full vehicle designs is presumed. Therefore, the search for these islands is an important part of the optimization task. The second part consists of the weight optimization on one island. Hence, in the following, algorithms for global as well as local optimization are employed.
In the course of the design process, a time frame of about two weeks is previewed for the complete optimization task.

1.1 Previous approach

Starting from Best Practice designs, numerous optimization runs with different start values have been executed by means of mathematical optimization. In order to perform a systematical search in the design space, DOE methods have been tested to define start values. About 100 starting values have been selected out of the multitude of possible sampling points in the high dimensional space. Starting from these values, it has been tried to find admissible weight optimized designs using a gradient optimizer. By this strategy, an admissible weight optimal design weighing 1096 kg has been found. It is displayed in fig. 9 to 14 as a comparison.

At BMW, long lasting experiences exist regarding the calibration of control parameters for gradient optimization, as well as on how to start an optimization in admissible or inadmissible regions, respectively. The formulation of the objective function is depending on several factors: on one hand, whether the starting point is situated in an admissible or an inadmissible region, and, on the other hand, whether the aim of the optimization is to find admissible designs or to reduce weight. Additionally, it is advantageous to integrate the constraints into categories and to extend the objective functions with the maximum value of these categories and additional punishing terms. All these experiences have been used to obtain the previously shown solution of the task by means of mathematical optimization.

In theory, the gradient based methods may lead to the closest “dominant” local minimum. That is due to numerical reasons, the nonlinearities, and the finite step width. Therefore, these optimization methods are called local approaches. Nevertheless, in the course of the iterations also a different local optimum of the objective function may be met. Therefore, it is impossible to define the covered path of a gradient optimization as the confidence region of an optimum. With this strategy, gradient optimization is a “random” search strategy with a high probability that the found state is admissible. That leads to considerable numerical expenses. In this case, one function evaluation took about 10 minutes, leading to a 60 to 100 times longer gradient optimization. Therefore it seems advisable to consider global optimization by means of genetic algorithms.

1.2 Motivation of the hybrid optimization strategy

Even if, in some cases, convenient gradients can be determined and the gradient solver finds admissible design spaces, it can be assumed that gradient solvers are successful only on an “island”. Therefore, we suggest a genetic search strategy for the global search on potentially different islands. As mentioned above, the constraint equations and the aim of weight minimization are contradictory. Therefore, the operators of the genetic optimization are calibrated in order to priorily search for islands of optimal designs. At the same time, as little as possible weight should be added. Still a significant potential for weight optimization of the found admissible designs is expected. In the next step, the weight optimization on the islands is executed by means of gradient optimizers, mainly to rise efficiency. As the case may be, multiple valid designs may be available for gradient optimization. Therefore the set of valid designs is to be investigated by means of cluster analysis in order to find alternative vehicle concepts as well as appropriate starting values for weight optimization.

It is of vital importance for the performance of the genetic search algorithm to use as much of the available know-how as possible. For the start values, this may be the best practice designs; for later generations, the information of previous optimization steps. This know-how is integrated in the start

generation, the constraints and the objective terms as well as in the operators of the genetic search strategy.
It showed in the course of the work that the creation of the start generation by means of stochastic sampling methods in the original space, i.e. 1500 optimization variables with upper and lower bound, is not convinient. This approach leads solely to start designs with too much weight to be interesting. Hence, in the design space defined by the boundaries of the optimization variables, there is only an extremely small subspace interesting for weight optimization. It has become necessary to find a different way to generate an appropriate start generation. Based on the present design state, a start generation is created by means of hierarchical mutation and is shifted into a region of low weight.

1.3 Global optimization by means of genetic search strategies

Of course, the expected numerical expense of one single calculation of all load cases is of crucial importance for the choice of the genetic operators and the size and number of generations. In moderate design spaces, i.e. with about 10 to 100 variables, at least 50 design evaluations, distributed into 5 generations with 10 designs each, are necessary to reach significant design improvement as far as our experience goes. If this condition is fulfilled, the genetic strategies can develop something that might be called “genetic intelligence” which leads to design improvements that go beyond random improvements as achieved with Monte Carlo approaches. In the present case, one single calculation of all loadcases on one CPU did last less than ten minutes. Furthermore, a Linux cluster with 20 CPU’s was available. Therefore, a total amount of 10,000 runs has been considered feasible, and the genetic operators have been calibrated accordingly. For the first step of the global optimization, a generation size of 50 and a number of 100 generations have been chosen. The start generation has been chosen relatively large in order to increase the probability of finding multiple design islands.

Considering such a large number of optimization variables, significant correlations between the variables can be expected. Therefore, the hierarchical structure of the variable tree has been taken into account in the initialization of the start generation. Herein, a hierarchy of eleven assembly groups and 44 components has been taken into account, such as front end, cowl, bottom, or rear, on one hand, and bumper, engine mounting, or shock absorber mounting, on the other hand.

The 50 designs of the start generation have been initialized using Latin Hypercube Sampling around a reference design, utilizing the hierarchy. The following mutations have been used:

- the assembly groups have been sampled with a standard deviation of 0.2 around the reference design. This corresponds to a scattering around the reference design in the 2-sigma-value of +-40%.
- additionally, the components have been sampled with a standard deviation of 0.5 around the reference design. This corresponds to a scattering around the reference design in the 2-sigma-value of +-10%.

Following this, the start generation was scaled into the weight range of about 1100 kg, in order to find islands with as little weight as possible.

1.4 Objective function and constraint for global optimization

In genetic algorithms, the objective function is also called “fitness“ and consists of the objective function of the optimization task as well as constraint terms.

The weight and the constructed space are included in the objective function. The constraints equally considered concern the stiffness (deformation), the eigenfrequencies, the distances between the eigenfrequencies, the stresses as well as acceleration values from harmonic analysis. All constraints of one demand class (e.g. constraints regarding eigenfrequencies) are combined. If all constraints of one class are fulfilled, this results in a sum less than 1.0. If one or more constraints are violated, the sum takes a value greater than 1.0. All constraints are introduced into the fitness function in the form of scaled punishing terms. The scaling factor is chosen so that a fulfillment of a constraint is rated with a value of 1. In case of violation, the scaled values increase up to 400. Additionally, a constant punishing term is included into the fitness function in order to reliably distinguish admissible from inadmissible designs in case of violation of one or more constraints.

The operators of genetic strategy contain elitism, selection, reproduction, and mutation. They are chosen such that the design space is mainly searched for admissible designs by the genetic optimization strategies.

The constructive constraints are controlled subsequent to the creation of the design variants and "repaired" in case of violation. Thus it is ensured that only those designs fulfilling the construction constraints are evaluated. The repaired values of the design variables are entered into the calculation before the solver run.

2. Global and local optimization

2.1 Global Optimization – search for islands by means of genetic optimization strategies

The following adjustment of the genetic optimization strategy have been chosen:

- In each generation, 50 indivuduals are calculated, the total number of generations being 100.
- The island search begins with a start generation that is initialized around the reference design using the knowledge of assembly group and component hierarchy and scaled into an interesting weight region.
- Following each generation calculation, two elite designs are determined by means of roulette wheel logic and 99% scaling of the roulette wheel. These elite designs are entered unchanged into the next generation.
- After each generation calculation, the ten worst designs are determined by means of roulette wheel logic and erased from the generation.
- The parent pairs are allotted by means of roulette logic as well from the gene pool of the current generation.
- The genetic interchange is executed using a uniform crossover probability of 8 %.
- For the mutation, a standard deviation of 0.03 in the normalized range from lower to upper bound of the optimization variables is used. This mutation rate is decreased linearly to 0.01 in the course of 100 generations.

After about 15 generations, i.e. 750 calculation runs, the genetic algorithm had grown a sufficiently good genetic memory. Thus, successful genes and search directions could be identified, leading to a continuous improvement of the constraints. After 67 generations resulting in a total of 3348 solver runs, the optimization was stopped. Up to this point, not one admissible design fulfilling all constraints could be found. The interpretation of the optimization revealed the following trends:

The stiffness constraint shows a good gradient and is reliably fulfilled after the 60th generation, that means 3000 solver runs (fig. 2). The exceedance of the eigenfrequency constraints trends to decrease, but still is slightly violated (fig. 3).
The weight clearly trends to rise during the optimization and stagnates from generation 60 at a level of about 1120 kg (fig. 4). Additionally, the constructed space is clearly increased (fig. 5) as well as the first bending mode (fig. 6) and the first torsion mode (fig. 7).

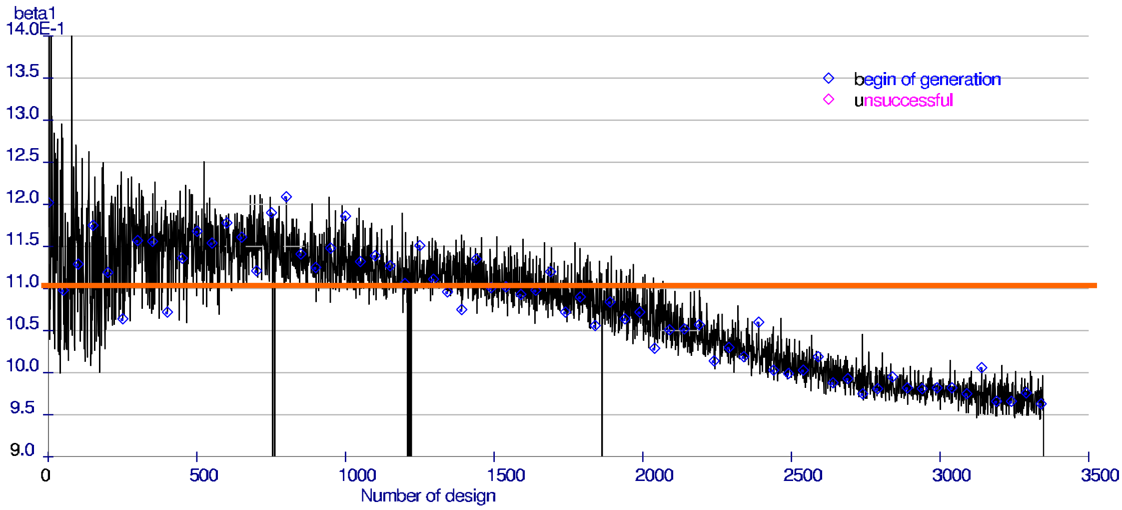

Fig.2 History of the stiffness constraint (starting from a value of 1.0, indicated by the bold red line, the constraint is fulfilled)

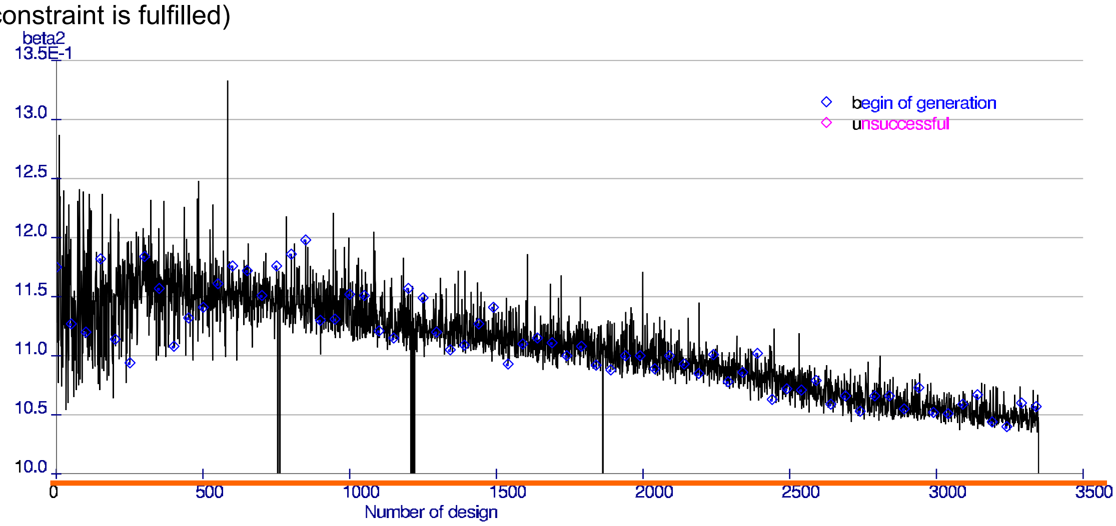

Fig. 3 History of the eigenfrequency constraint (starting from a value of 1.0, indicated by the bold red line, the constraint is fulfilled)

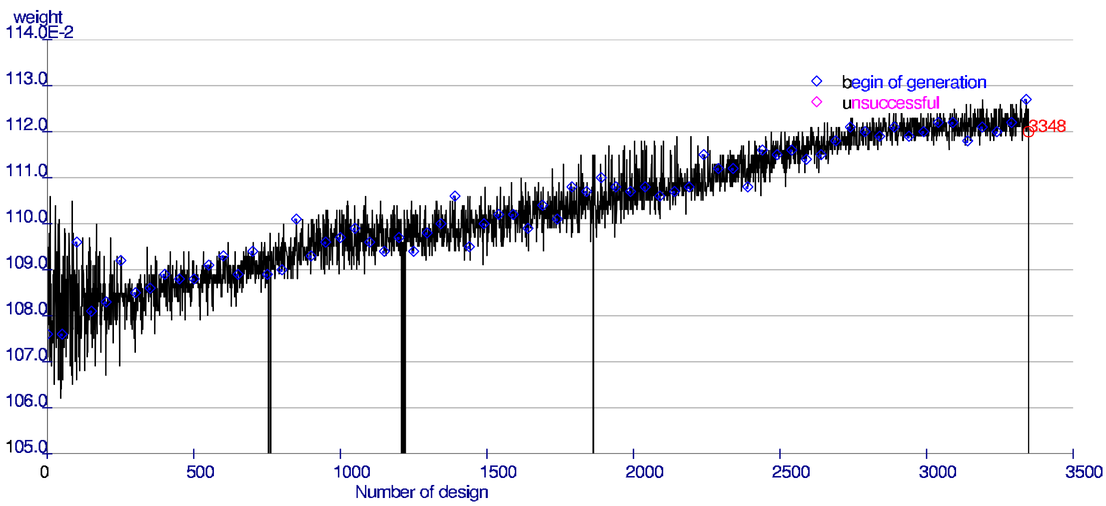

Fig. 4 History of weight

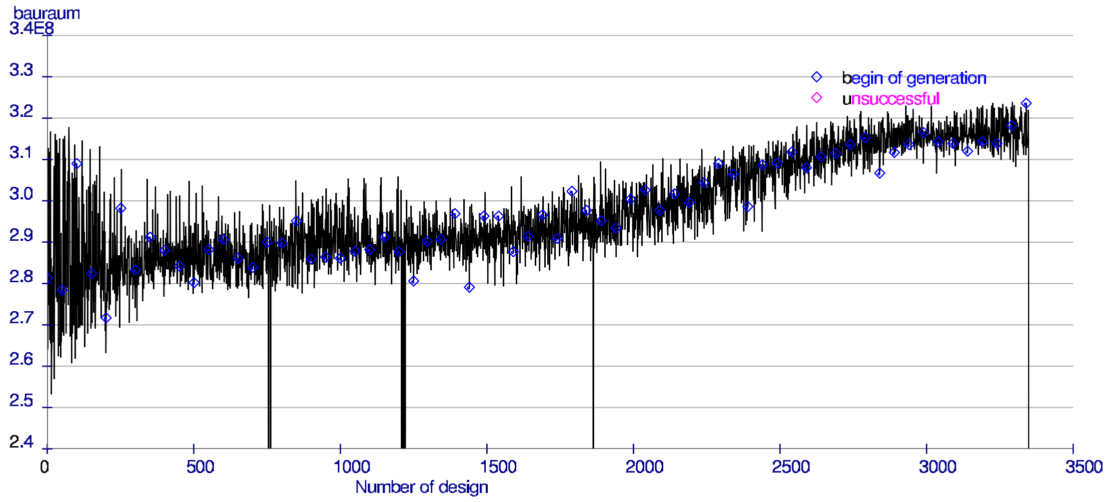

Fig. 5 History of constructed space

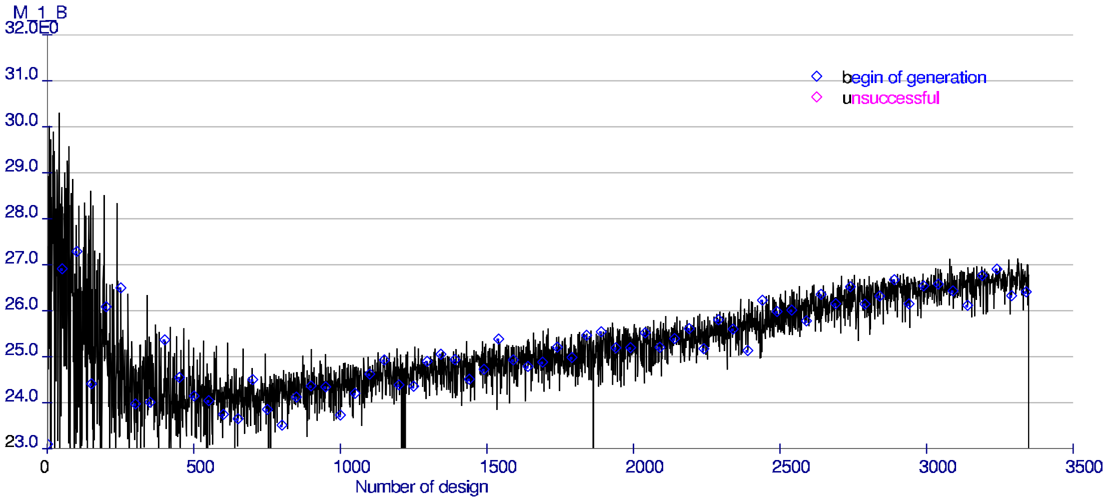

Fig. 6 History of the first bending mode

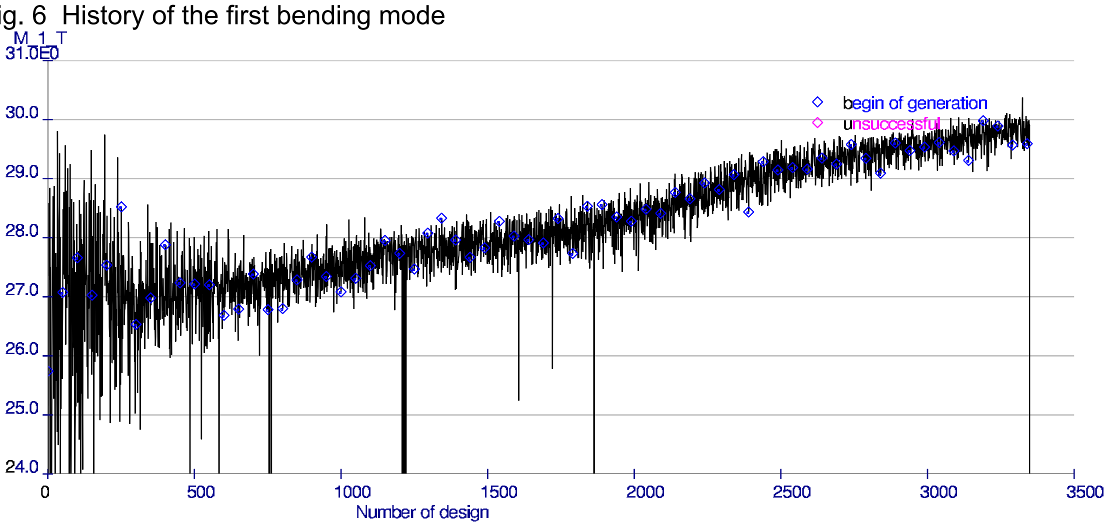

Fig. 7 History of the first torsion mode

2.2 Restart of the genetic optimization

The knowledge derived from the previous optimization is integrated into the start generation, and the genetic operators are adjusted:

- 100 individuals per generation and a total of 100 generations are calculated.
- The start population consists of 100 designs that were chosen from those designs of the previous optimization that did best fulfill the constraints.
- After each generation, five elite individuals are determined by means of roulette wheel logic and a 90 % scaling of the roulette wheel. These are entered unchanged into the next generation.
- After each generation, ten worst designs are determined by means of roulette wheel logic and deleted from the generation.
- Again by means of scaled roulette wheel logic, the parents are determined from the gene pool of the current generation.
- The genetic crossover is performed with a uniform crossover probability of 15 %.
- A standard deviation of 0.05 from lower to upper boundary in the normed space of the optimization variables is used for mutation. In the course of the calculation of the 100 generations, this mutation rate is decreased linearly to 0.03.

With the adapted genetic optimization strategy leading to higher genetic crossover and mutation, an increase of the convergence speed of the constraint fulfillment was aimed at. At the same time, the stronger elitism served to concentrate the genetic optimization around the previously determined best designs. The generation size was increased in order to minimize the probability, that a convergence to local optima could be enforced. As can be derived from the following results, the adjustment of the genetic operators was successful regarding speed as well as probability (which describes the amount of admissible designs). The Best_Design could be found in generation 28 (Design_2717) weighing 1148 kg, with a constructed space of 3.38e8 mm^4 . In total, 1018 admissible designs have been found. As the genetic optimization concentrates on the search for islands, the weight is still considerably high (1150 kg).

In the history plot of the objective function considering only the influence of weight and constructed space, it can be seen that the chosen genetic strategy finds islands with a good trend. The first admissible design could be found in the solver run 1219 with a weight of 1148 kg. Starting from generation 30, the variance of the objective function does increase considerably. That means that the chosen genetic strategy creates too much mutation. Only a weak trend towards design improvement can be detected in the following.

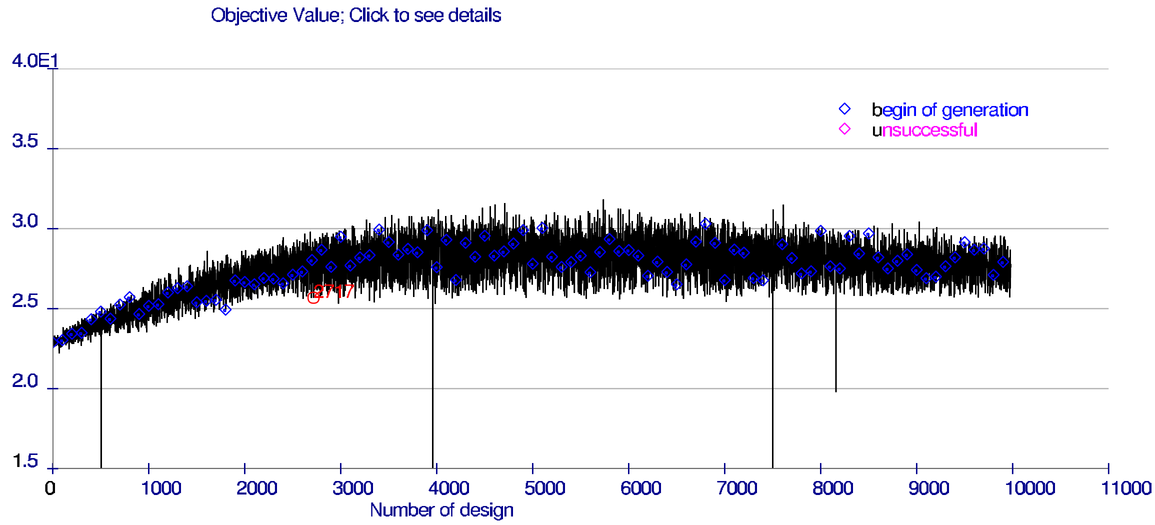

Fig.8 History of the objective function

2.3 Cluster analysis and local optimization

By means of cluster analysis, a total of ten clusters is determined from the subspace of the 1018 admissible designs. They are searched depending on a radius in the normed space. A cluster's center of gravity is constituted by the two most similar designs of each cluster. The cluster algorithm determines the two most similar designs from the bulk of admissible designs. Then, it detects all designs within the cluster radius. These designs form cluster 1. Equally, cluster 2 and all further clusters are determined from the remaining space of admissible designs. The cluster determination is repeated with gradually larger radius up to the point where the clusters coincide. This happens because the cluster membership depends on the radius. Thus, the largest possible cluster radius leading to the smallest possible subdivision of the admissible designs into clusters is determined. In the following, local optimizations by means of gradient optimization are started from ten selected designs. For each cluster, it is always the design most reliably fulfilling the constraints that is selected as a starting point for the gradient optimizer. These designs are assumed to have the largest weight optimization potential. In the following table, the most successful optimization in cluster 4 is evaluated. An admissible design weighing as little as 1056 kg could be found by means of cluster analysis and local optimization using NASTRAN Sol200.

	Weight [kg]	Constraint Stiffness OptiSLang	Constraint Eigenfrequencies OptiSLang	Maximum constraint NASTRAN
OptiSLang Cluster 4	1160	0.89	0.989	
NASTRAN_Optimization from the gravity center of Cluster 4	1056			1.000

Tab. 1 Optimization results using NASTRAN Sol200 for Cluster 4

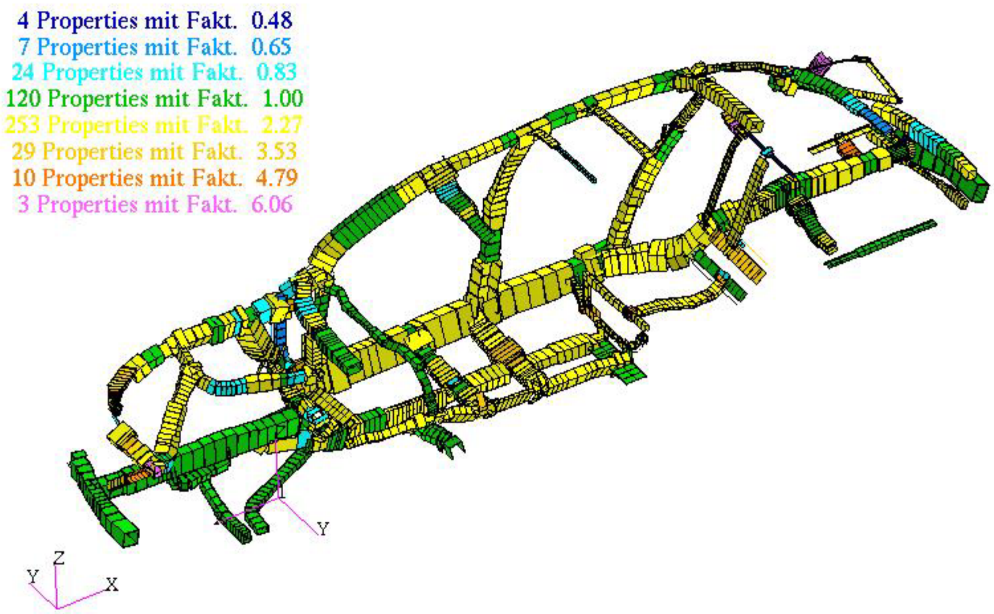

Fig. 9 Construction space difference from Design_Cluster_4 to Best_Design_Natran_old

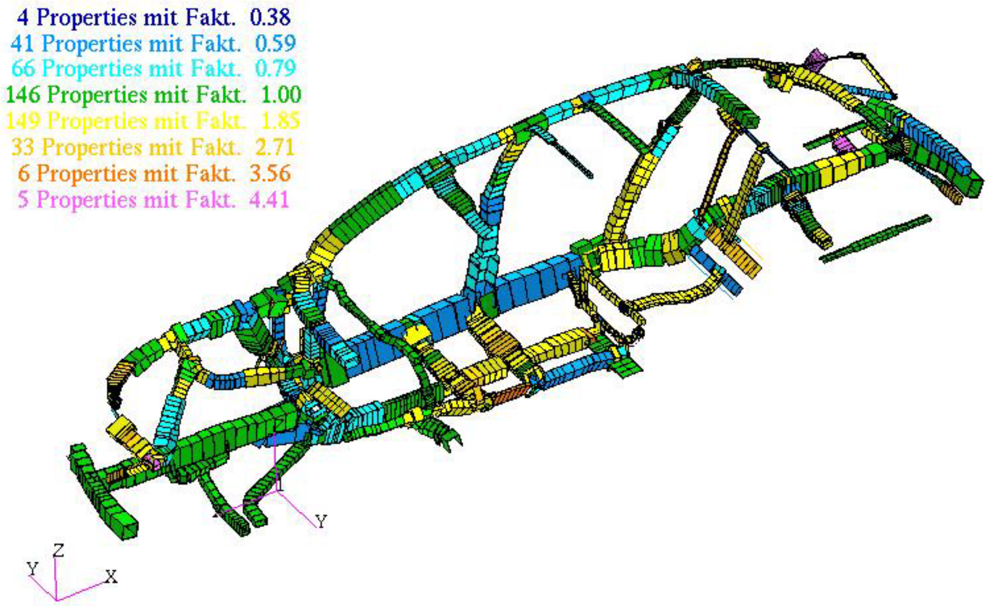

Fig. 10 Weight difference from Design_Cluster_4 to Best_Design_Natran_old

In the following, alternative whole vehicle concepts have been identified. This was done by evaluating the scaled distance of two design vectors (i.e. vectors of optimization variables). The local optimization of the clusters 1 to 10 did yield very similar designs. Therefore, it is assumed that all these clusters belong to one construction variant. That means that all clusters lie on one construction variant island. Thus, the cluster analysis is continued by increasing the cluster radius until the first ten clusters coincide and ten new stable clusters form. These clusters are called clusters 11 to 20, and for each of it a local optimization using NASTRAN Sol200 is performed. Again, for each cluster, the design most reliably fulfilling the constraints is selected as start value. Hereby, in cluster 11, an admissible design with a weight of only 1037 kg could be found. The optimization of cluster 11 is illustrated in table 2. Subsequently, a comparison of the scaled distances of the optimal designs of all 20 clusters and the previously best design of a conventional NASTRAN Sol200 optimization did yield large distances of the optimal designs in the design space. Thus, it is assumed that the designs belong to different construction variants. In fig. 13 and 14, the differences of the two construction variants are shown.

	Weight [kg]	Constraint Stiffness OptiSLang	Constraint Eigenfrequencies OptiSLang	Maximum constraint NASTRAN
OptiSLang Cluster 11	1167	0.868	0.983	
NASTRAN_Optimization from the gravity center of Cluster 11	1037			1.000

Tab. 2 Optimization results using NASTRAN Sol200 for Cluster 11

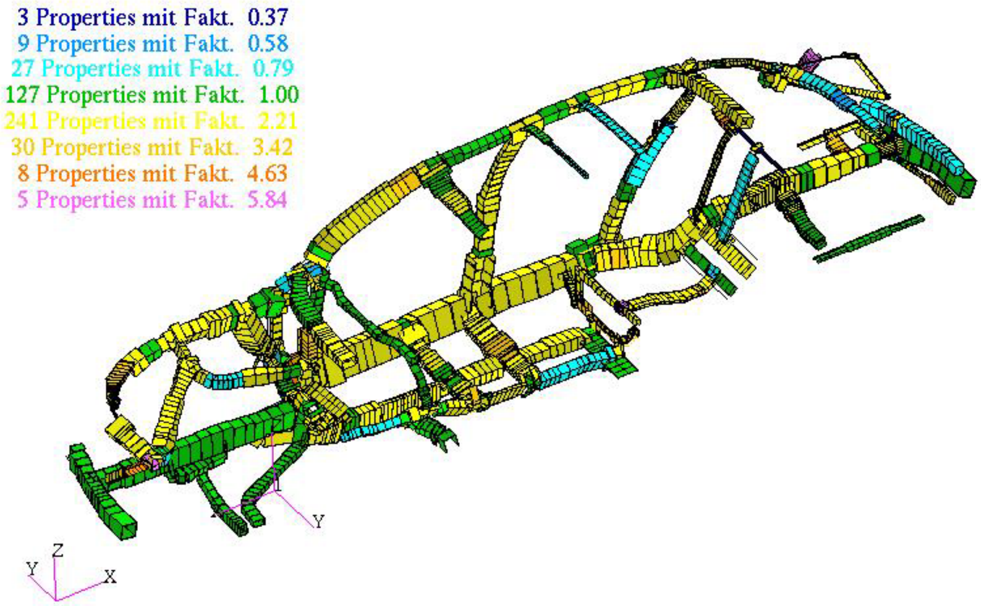

Fig. 11 Construction space difference from Design_Cluster_11 to Best_Design_Natran_old

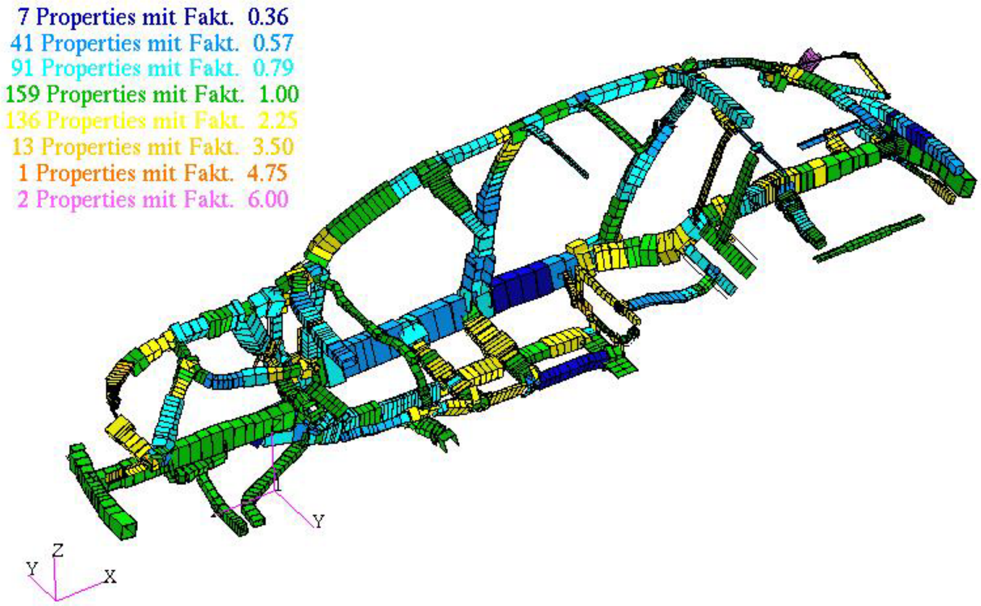

Fig. 12 Weight difference from Design_Cluster_11 to Best_Design_Natran_old

3. Summary of the hybrid optimization

Compared to the previous approach, approximately 60 kg could be economized in the preliminary draft by means of hybrid optimization. That represents about 5% of the total weight.

Comparing the weight optimal designs of the first 20 clusters, significantly different admissible designs of interesting weight could be identified. Herein, a resemblance criterion for alternative vehicle concepts has been used. Thus, admissible designs with an interesting weight could be found that belong to different design islands in the design space, i.e. different construction variants. For example, in the designs illustrated in fig. 13/14, this results in a significantly different constructed space in the rear as well as partly significantly different weights of the rear, the c-column, the roof and bottom structures.

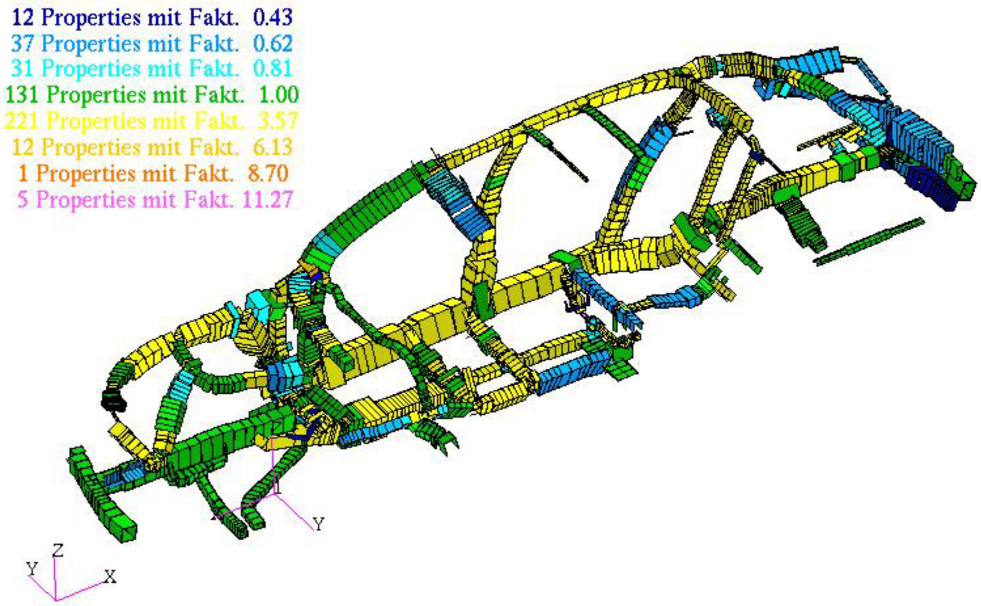

Fig. 13 Constructed space difference from Design_Cluster_4 to Design_Design_Cluster_11

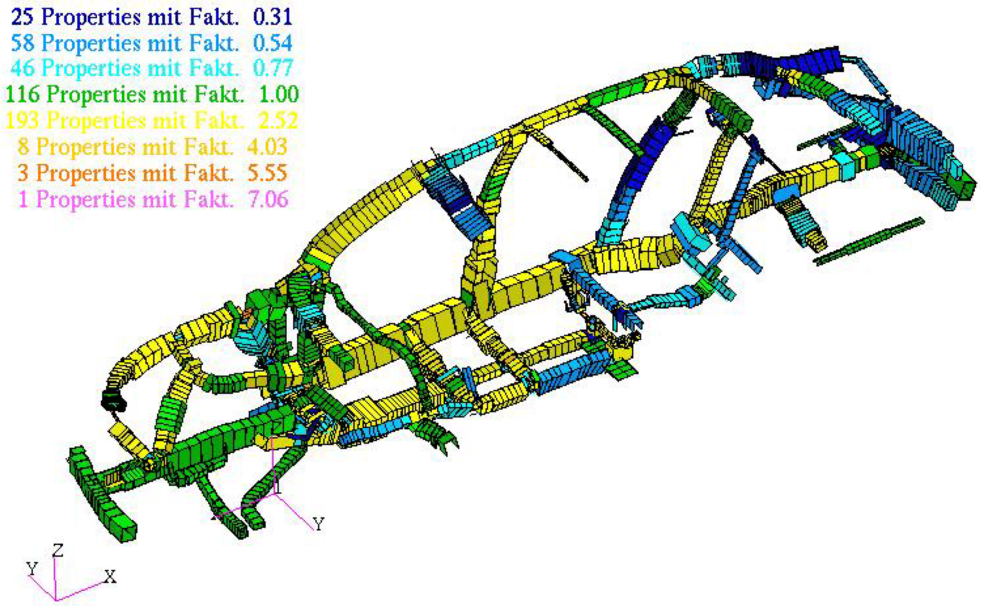

Fig. 14 Weight difference from Design_Cluster_4 to Design_Design_Cluster_11

4. Outlook

The present recipe for success can be summarized as follows:

- Use of a repair script for all designs proposed by the optimizer controlling whether the construction constraints are met and repairing the designs in case they are not. The repaired design is then reentered into the optimizer.
- Global optimization (island search) by means of genetic strategies as implemented in OptiSLang. Herein, the start generation is created around the reference design including the hierarchical knowledge, and scaled to the expected optimal weight.
- Having found a sufficient number of admissible designs, appropriate starting points for local optimization are determined by means of cluster analysis.

- Local optimization, i.e. weight optimization, is executed on the islands of admissible designs by means of a gradient optimizer.

All in all, about 13000 design evaluations with OptiSLang are necessary for the island search, resulting in 2 to 3 days of computing time. Afterwards, approximately 20 local optimizations using NASTRAN Sol 200 are performed.

A significant weight reduction could be achieved using this combination of sampling strategies, genetic optimization for global island search, cluster analysis for identification of islands of different design concepts, and gradient optimization for local weight optimization. Subsequently, a further project deals with the robustness of the whole process and further improvement of the performance of all its steps. Additionally, the whole process is integrated as a predefined workflow into a BMW specific version of the OptiSLang GUI.

References

[1] OptiSLang - the Optimizing Structural Language Version 2.0, DYNARDO, Weimar, 2004, www.dynardo.de

[2] Bucher C. et al.: SLang - the Structural Language Version 5.0, Institute of Structural Mechanics - Bauhaus-University Weimar, 2004.

[3] Will, J., Bucher, C., Riedel. J, Akgün, T. „Stochastik und Optimierung: Anwendungen genetischer und stochastischer Verfahren zur multidisziplinären Optimierung in der Fahrzeugentwicklung“; VDI-Berichte, Nr. 1701, 2002, S. 863-884