Hierarchical Multi-Level-Optimization of crashworthy structures using automatic generated submodels

Harman Singh[1], Axel Schumacher[1], Carlos J. Falconi D. [2], Alexander F. Walser[2],
Sven Trentmann[3], Leyre Benito C. [3], Christophe Foussette[4], Peter Krause[4], Heiner Müllerschön[5]

[1]University of Wuppertal, Faculty for Mechanical Engineering and Safety Engineering, Chair for Optimization of Mechanical Structures, Gaußstraße 20, 42119 Wuppertal, Germany
[2]Automotive Simulation Center Stuttgart e. V., Nobelstraße 15, D-70569 Stuttgart
[3]Gesellschaft für numerische Simulation mbH, Am Gaußberg 2, D-38114 Braunschweig
[4]divis intelligent solutions GmbH, Joseph-von-Fraunhofer-Straße 20, D-44227 Dortmund
[5]Scale GmbH, Friedrichshofener Str. 20, 85049 Ingolstadt

1 Introduction

The optimization of large crashworthy systems like a vehicle body in a crash loaded case is a time consuming and costly process. Extensive computer simulations are required to improve the crashworthiness of such large systems using an optimization technique. The simulation time can be reduced by dividing the large system into small subsystems also known as submodels. The submodels are used in the optimization to shorten the response time. But the generation of the submodels by hand is a challenging process and it requires a huge effort to validate them. This paper deals with the structural optimization of large crashworthy systems with a hierarchical Multi-Level-Optimization method using automatically generated and validated submodels.
The process of automatic generation of submodels is developed using the so called connecting island algorithm [1]. The algorithm is developed in *Tcl* language using the software *Generator* [2] and *Animator* [3]. The two important parameters in this process are the threshold ratio and the connecting island value. These are based on an evaluation function which is a structural response of the large system with time averaging and space averaging method. The size of the submodel depends on these two parameters. Evolutionary algorithms are applied to optimize the time required to generate an appropriate submodel size with help of *ClearVu Analytics* [4] software. The process of automatic validation of submodels is based on four methods, so called local, global, mean and response validation method which are discussed in detail in [1].
The Multi-Level-Optimization process is carried out in two hierarchical levels. The level 1 is a large system and level 2 is a submodel of the large system. Each level has different design variables, constraints and structural responses. The levels are coupled together using interface boundary conditions in form of nodal displacements. These boundary conditions are extracted from the large system and mapped onto the subsystem in an automatic process. If the boundary conditions are exact, the submodel region will deform identical to the deformation of this region in the large system [1] [5]. The update of these boundary conditions and the correlation between the two levels is studied and discussed in detail.
The research is demonstrated on an academic example of a cantilever frame impacted by a rigid sphere and on an industrial application of a *Toyota Yaris front crash* [6]. The future possibilities of the Multi-Level-Optimization method and the submodeling technique are discussed in this paper.

2 Hierarchical Multi-Level-Optimization

The structural optimization is a tool to improve the mechanical properties of a structure. The mechanical properties like the stiffness, weight etc. can be improved. The structural optimization is basically classified into three main categories i.e. the topology, the form and the dimensional optimization. For details about the classification and their definition refer to [7]. This paper deals with the optimization of the topology of the structure.

2.1 Optimization in Single-Level

An optimization process requires an analysis model and an optimization algorithm. The analysis model can be in form of an analytical or a numerical model which is to be optimized. During the optimization setup an optimization target and the input variables are defined. By varying the input variables the design of the structure can be changed. The target in the optimization is referred to that mechanical property

of the structure which is to be optimized. The optimization process in a Single-Level is shown in figure 1. The start design is a reference model in the optimization. The optimization algorithm variates the input variables and the loop repeats until an optimum or the stopping criteria is reached. The optimum here refers to a design which is better than the start design. The stopping criteria can also be the maximum number of function calls. One loop in the optimization process refers to one function call.

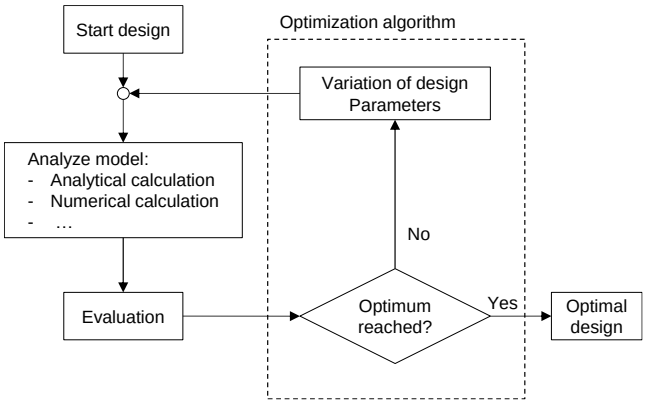


Fig.1: Optimization loop [7]

The optimization loop shown in figure 1 is the classical method of the optimization. The optimization target could be for example the increase of stiffness or reduction of weight of the structure. The problem of this classical method is that, depending upon of the optimization problem and the number of input variables, it can require a large of number of function calls. This problem could be handled by dividing the optimization into different levels. This type of optimization is referred as the Multi-Level-Optimization and is explained in detail in section 2.2.

2.2 Optimization in Multi-level

The process of running the optimization in different hierarchical levels coupled to each other is called the Multi-Level-Optimization process. The process for two hierarchical levels is shown in figure 2.

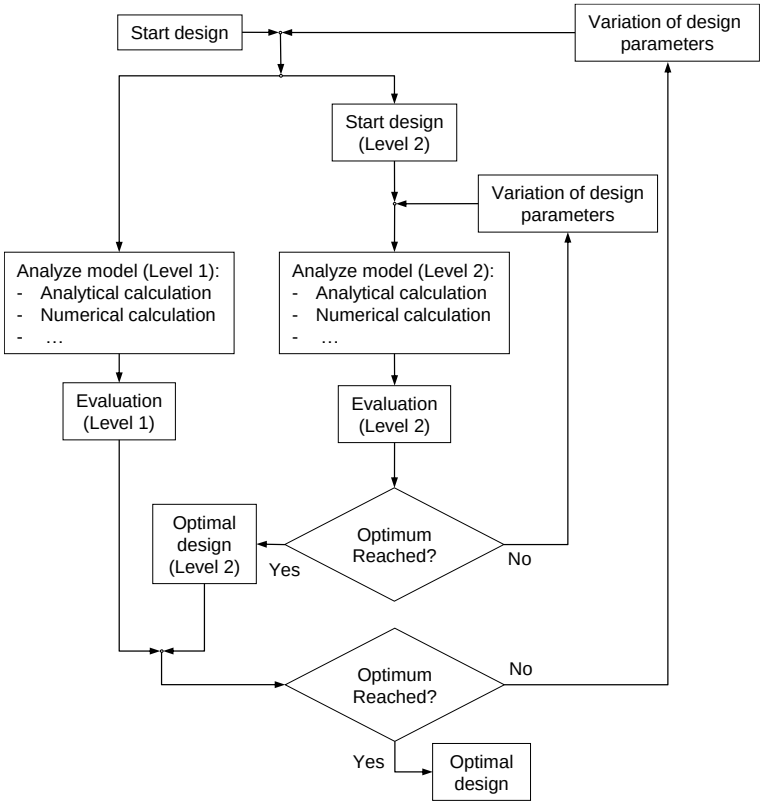


Fig.2: Nested optimization loop for the hierarchical levels 1 and 2 [7]

The level 1 is the outer loop and level 2 is the inner loop of the optimization. For each loop in level 1 an optimization in level 2 is started. Each level consists of an optimization loop with different set of input variables. The input variables of level 1 are first transferred to the level 2 which is referred to the start design in level 2. After the optimization in level 2 the optimum design of level 2 is transferred to the level 1. If the optimum is not reached the input variables are changed in level 1 and the loop is repeated until an optimal design is found in level 1. This kind of optimization process could be helpful in finding a global optimum faster and effectively. The Multi-Level-Optimization can be made more effective by introducing the submodels in level 2, which would help to fasten the optimization process.
A submodel is a region of interest cut out from the main model which is to be analyzed in detail. Due to the small size of the submodel their simulation time is relatively smaller than the main model. The main model here refers to the finite element model of a structure. The submodel is analyzed with help of boundary conditions extracted from the finite element analysis of the main model. The Multi-Level-Optimization process for two hierarchical levels with use of submodel technique is demonstrated in figure 3.

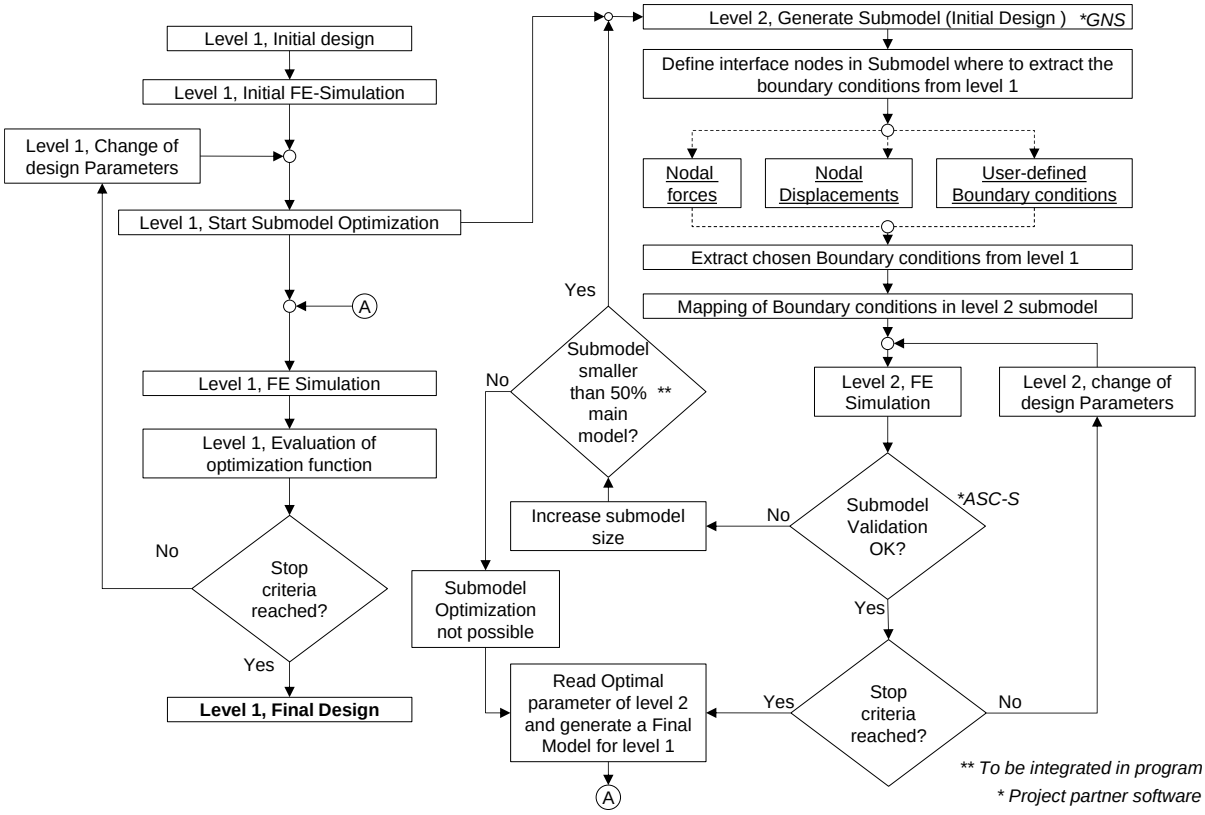


Fig.3: Nested Multi-Level-Optimization loop with submodel technique

The level 1 refers to the optimization of the main model and level 2 refers to the submodel optimization. The process starts with the start design in level 1 which is the main model. The initial finite element simulation is required to evaluate the main model in order to generate a submodel. The start design in level 2 is the submodel. To simulate a submodel it is necessary to define the interface boundary nodes. These interface nodes are at the boundary where the submodel is cut from the main model. The interface boundary conditions can be in form of displacements, nodal forces or user defined boundary conditions. This paper deals with the interface boundary conditions in form of nodal displacements. These boundary conditions are extracted from the main model using a finite element simulation of the main model. With these boundary conditions the submodel is simulated in level 2. Submodel validation process identifies the exactness of the boundary conditions. If the submodel is not validated the submodel size is increased. When the submodel size is greater than 50% of the main model the optimization in level 2 is not run. This step is however not yet integrated in the process displayed in figure 3. The idea it to use only validated submodels in the optimization in level 2. After a successful optimization in level 2, the optimal design of level 2 is transferred to the level 1. The new design in level 1 is referred as the final model. The final model in each iteration of level 1 is a result of a combination of input variables of level 1 and the optimized variables of level 2. If the stop criteria in level 1 is not reached, the design variables of level 1 are changed and the optimization loop is repeated. The stop criteria can be the maximum

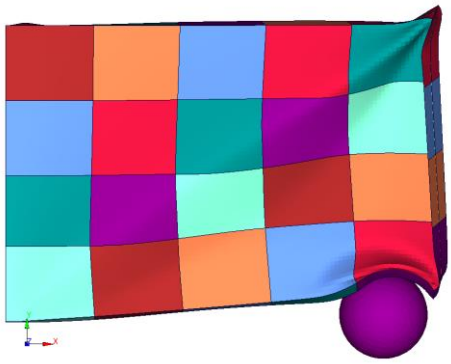
number of iterations or an optimum reached. A study is performed in chapter 3 to analyze and discuss the Multi-Level-Optimization with and without submodels.

3 Cantilever frame example

The Multi-Level-Optimization workflow is demonstrated on an academic example of a cantilever frame hit by a rigid sphere. This academic example is explained in detail in section 3.1.

3.1 Structural model and optimization problem

The structural problem is an aluminum cantilever frame of mass 45 g clamped at one side and impacted by a rigid sphere. The sphere has an artificial mass of 1 kg and initial velocity of 8000 mm/s. The cantilever is made of a sheet of size 150 x 100 mm and thickness 0.6 mm which is surrounded by a frame of width 15 mm and thickness 1 mm. The sheet consists of 20 parts and the frame consists of 14 parts. The structural analysis is performed using the finite element method with explicit time integration in software *LS-DYNA*. The result of the Finite-Element simulation of the cantilever frame is depicted in figure 4.



Reference design
Sphere Intrusion = 18.0786 mm
Sphere End velocity = -51.58 mm/s

Fig.4: Results of the finite element simulation of cantilever frame hit by a rigid sphere

The result in figure 4 is used as the reference design which has the sphere intrusion of 18.0786 mm. The optimization task is to minimize the intrusion of the rigid sphere. The restriction is the end velocity of rigid sphere should be less than or equal to 0 mm/s. To achieve this a bead is inserted in the structure to increase its stiffness. The design parameters are the position and the angle of the bead. The bead position is a discrete variable which defines the position of the bead at center of one of the 20 parts in the inner field of the cantilever. The bead angle is a continuous design variable between -45 and +45 degrees. The angle is limited to reduce the number of function calls. The bead length is 20 mm and radius 3 mm which are constants in the optimization. The design parameters bead position and bead angle (exemplary -45°) are displayed in figure 5.

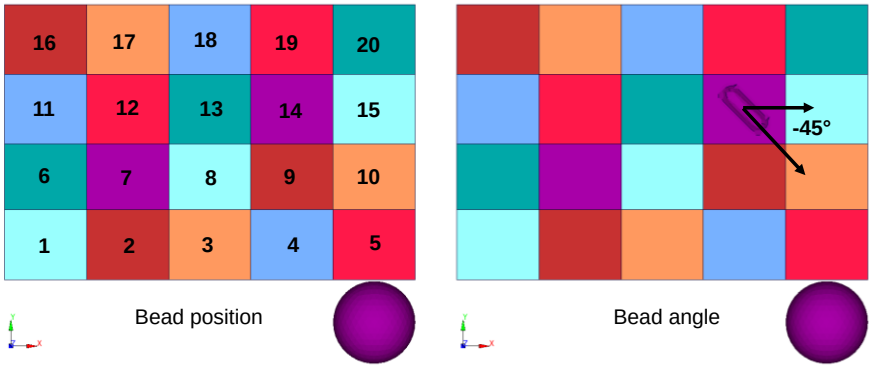


Fig.5: Design parameters: Bead position and angle in the Cantilever frame example

The start design for the optimization has bead position 19 and bead angle 9.73 degrees, with sphere intrusion of 16.2651 mm. This design is found by performing a design of experiments for the cantilever frame problem. The submodels to study the Multi-Level-Optimization are discussed in section 3.2.

3.2 Generation of submodels for the cantilever frame

The submodel for the cantilever frame is generated using two input parameters, the threshold limit (TR_{lim}) and the connecting island limit (CI_{lim}). These input parameters define the shape and size of a submodel. The detailed explanation of these parameters can be found in [1]. Figure 6 displays two submodels generated with two different combinations of TR_{lim} and CI_{lim} for the bead position 19 and the bead angle of 9.73 degrees.

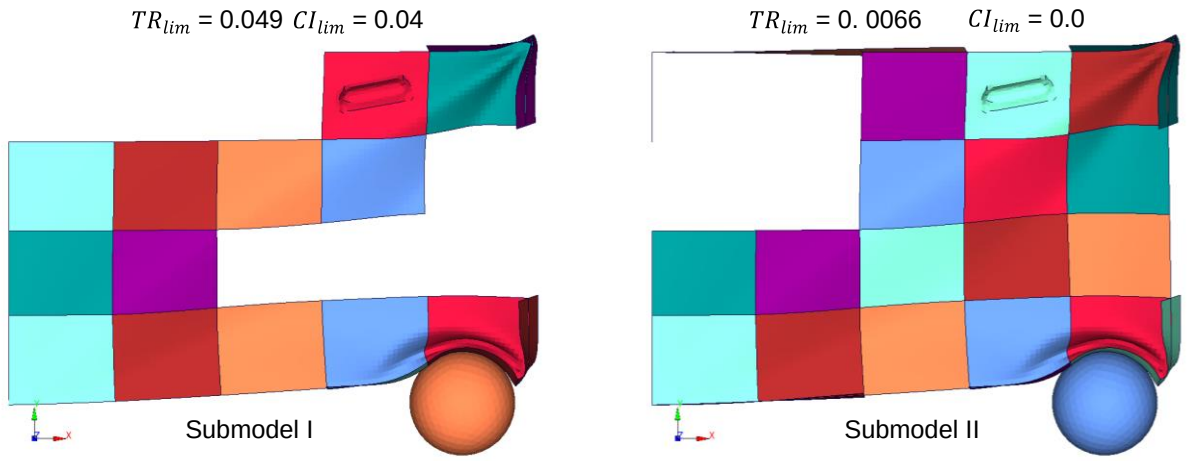


Fig.6: Submodel I and Submodel II

Submodel I has a size 53.42 % and submodel II has a size 76.12 % of the main model. The submodels are simulated using the finite element method with nodal displacements as interface boundary conditions. These boundary conditions are extracted from the main model with same design parameters as mentioned. Using these boundary conditions on the submodel, we get nearly the same sphere intrusion as in the main model. But when the design parameters are changed the structural behavior also changes. And this could mean the boundary conditions are no longer valid for the changed design of the submodel. But the question here is how far can these boundary conditions deliver good results? Is there a criteria which can show if the boundary conditions are accurate for a submodel? How to identify if the results delivered by a submodel in level 2 will also be valid in level 1 of the Multi-Level-Optimization? A criteria to identify the quality of submodels could be very helpful for the development of submodels.
In the following, one criteria is based on the distance between the bead and the interface boundaries which is explained in detail. The criteria of “Distance between Bead and Interface” could help to examine the accuracy of a submodel. This measures the smallest distance between the bead boundaries to the nearest interface node in a finite element model. This depicts how close the modification in a submodel to the interface boundaries is. The criteria is demonstrated on submodel I and II in figure X.

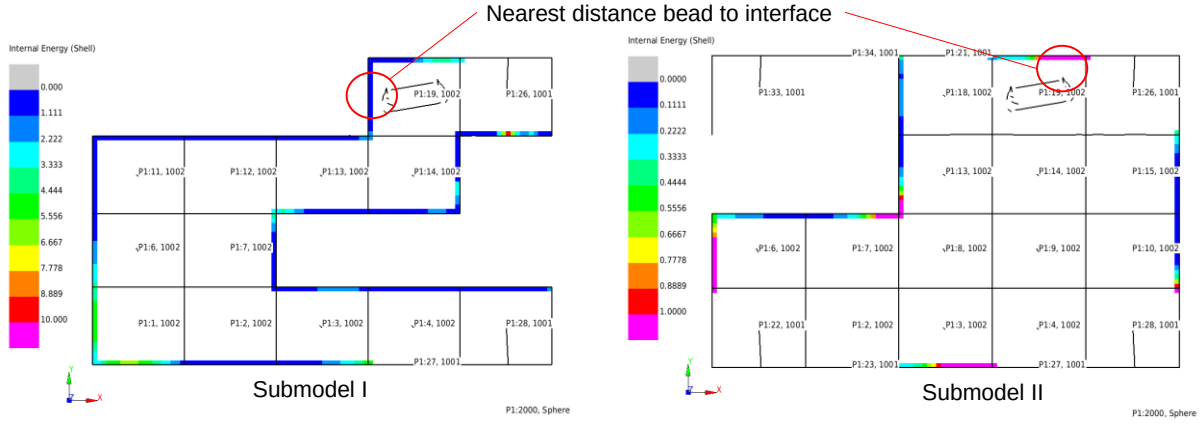


Fig.7: Smallest distance between the bead and the interface boundary shown in submodel I and II

The colored boundaries in figure 7 represent the interface boundaries in the submodels. The color range depicts the internal energy at the interface boundaries. The smallest distance b/w bead and interface in

submodel I is 3.57 mm and in submodel II is 6.97 mm. When the bead angle variates the distance will also variate. But in submodel I the distance will remain low for bead angles between -45 and 45 degrees. Because the interface boundaries in submodel I are surrounding the bead on two edges. Whereas the interface boundary in submodel II surrounds the bead only at one edge. Which means the bead is closer to the interface boundary at extreme bead angles and far for lower bead angles. It will be interesting to see influence of the bead distance to the interface boundary when the bead angle is varied in submodel. Is there a correlation between the distance of the bead to interface and the sphere intrusion in the submodel? Can this distance be used as a criteria for the validation of submodels? To verify this a design of experiments is performed where the bead angle is varied between -45 and +45 degrees with an interval of 5 degrees. This experiment is done with both submodels and the main model. The results of sphere intrusion are compared. First we examine the results of sphere intrusion in the main model.

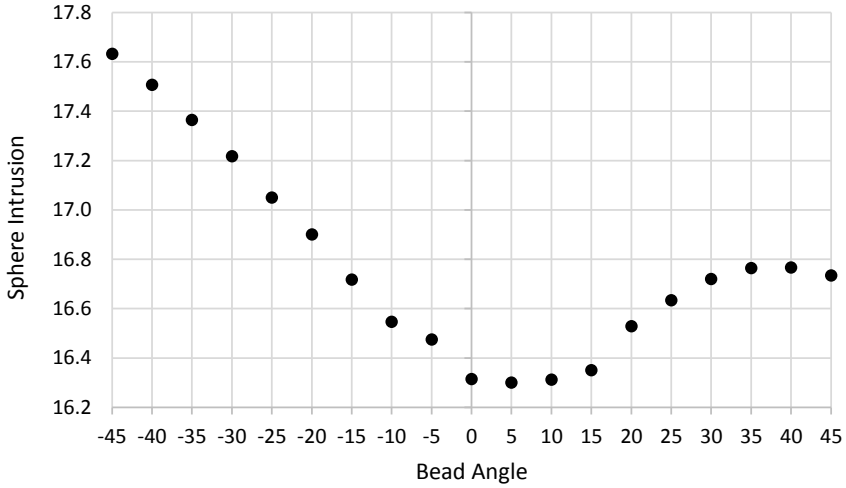


Fig.8: Sphere Intrusion at bead position 19 for different bead angles in main model

The appearance of results in figure 8 demonstrates the sphere intrusion at the various bead angles at bead position 19. The difference between the minimum and the maximum intrusion is approximately 8%. The sphere intrusion is minimum for the bead angle between 0 and 15 degrees. Whereas the sphere intrusion increase at the extreme values of the bead angle which is near to -45 and +45 degrees. These results are also expected from the submodels. Figure 9 shows the design of experiment results for submodel I.

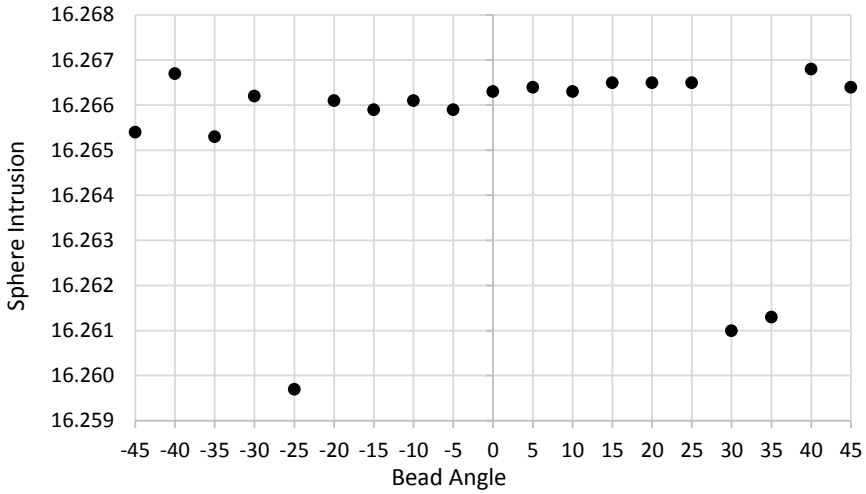


Fig.9: Sphere Intrusion at bead position 19 for different bead angles in submodel I

The results depicts the intrusion not being sensitive to the variation in bead angle in submodel I. The problem of this insensitivity lies in the interface boundaries being too close to the bead. Due to which the changes in the bead angle does not affect the mechanical behavior of the submodel. To see the influence of the distance of the bead to the interface boundaries, the same design of experiments is also

carried with submodel II. The sphere intrusion in submodel II for different bead angles is plotted in figure 10.

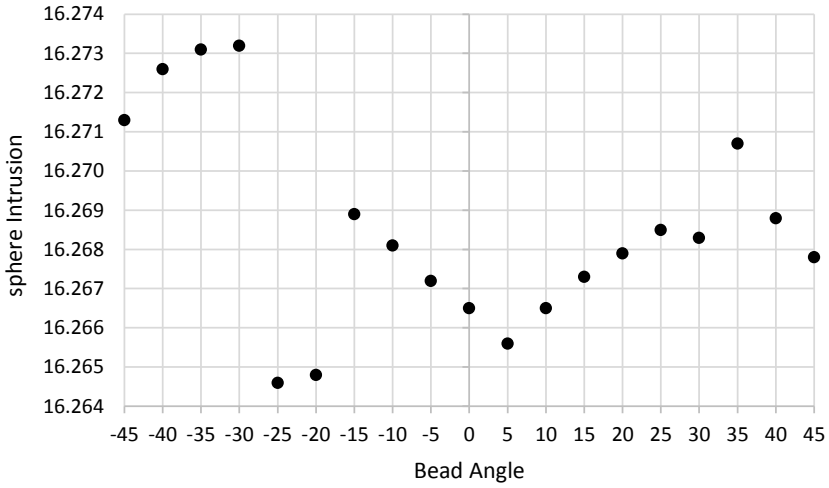


Fig.10: Sphere Intrusion at bead position 19 for different bead angles in submodel II

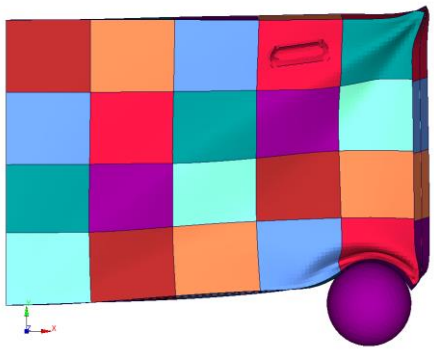
The results with submodel II follow almost the same trend as in the main model results shown in figure 8. It is also clear that the intrusion is not sensitive enough to the bead angle. Though in comparison to the submodel I the results with submodel II are much better. This is because the mechanical behavior in the submodel is able to react sensitive to the design modification in submodel. Which is only possible at bead angles between -15 and 15 degrees because the influence of the interface boundary conditions on the bead is less. But when the bead angle is below -15 and above +15 degrees, the distance between the bead and the interface boundary decrease. This leads to poor results in the sphere intrusion as seen in figure 10. It shows that the interface boundary conditions are only valid for a certain amount of change in the submodel. If the distance of bead to interface is able to predict the behavior of the submodels in the optimization, this could be used as a validation criteria for submodels. It is here necessary to perform these experiments with other examples and develop more criterion for validation. In this case the submodel II is a better choice for the use in the Multi-Level-Optimization. To verify the results the optimization studies are shown in section 3.3.

3.3 Optimization studies

The optimization is carried out using the three different workflows: Single-Level-Optimization (Section 2.1, Figure 1), Multi-Level-Optimization without submodel (Section 2.2, Figure 2) and Multi-Level-Optimization with submodel (Section 2.2, Figure 3).

3.3.1 Single-Level-Optimization

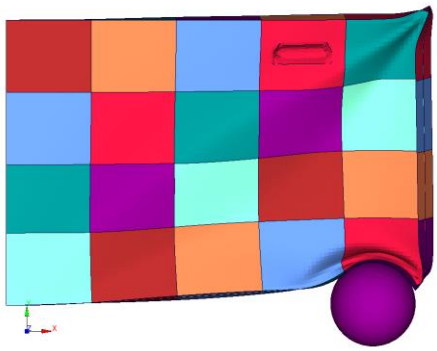
The Single-Level-Optimization of the academic model is executed using 3000 function calls. The number of design points in each iteration are 40 and the maximum number of iterations are 100. The optimal design is shown in figure 11.



Optimal design in Single-Level (1)
Sphere Intrusion = 16.2067 mm
Sphere End velocity = -54.47mm/s
Bead position = 19
Bead Angle = 6.63342°
Algorithm = Adaptive simulated annealing
Designs per iteration = 30
Iteration = 100

Fig.11: Optimal design in Single-Level-Optimization using direct simulation with 3000 function calls

The optimal design has a bead position 19 and bead angle of 6.63 degrees. The intrusion in the optimal design is 16.21 mm which is 10.35% better than the intrusion in the reference model. In order to compare the Single-Level-Optimization with Multi-Level-Optimization the maximum number of function calls are set to 400. The optimal design with 400 function calls in a Single-Level is displayed in figure 12.



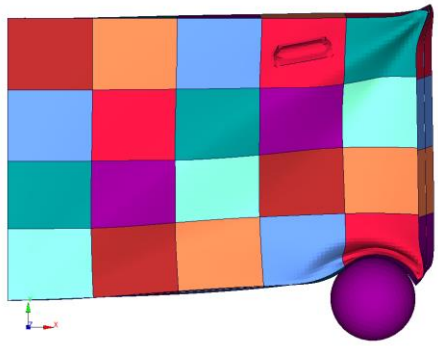
Optimal design in Single-Level (2)
Sphere Intrusion = 16.2447 mm
Sphere End velocity = -55.05 mm/s
Bead position = 19
Bead Angle = -0.261968°
Algorithm = Adaptive simulated annealing
Designs per iteration = 40
Iteration = 10

Fig.12: Optimal design in Single-Level-Optimization using Metamodel based with 400 function calls

The optimal design in this case has the bead position 19 and the bead angle of -0.26 degrees. The intrusion in this design is 16.24 mm. This design is comparable to the optimal design shown in figure 5 because the difference in intrusion is low. Therefore it is used as reference for the comparison with Multi-Level-Optimization.

3.3.2 Multi-Level-Optimization without submodel

In order to have a comparable results, the Multi-Level-Optimization also has a total number of function calls set to 400. Therefore the maximum number of function calls in level 1 and level 2 are 20 each. The bead position is varied in level 1 and bead angle in level 2. The optimization goal in both levels is to minimize the sphere intrusion. The result of Multi-Level-Optimization without submodel is demonstrated in figure 13.



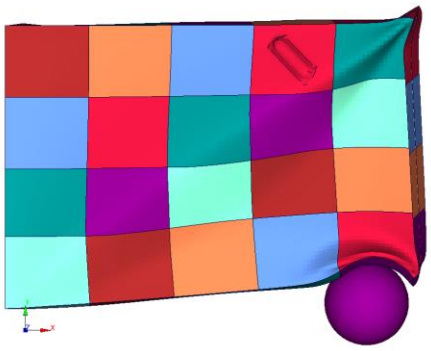
Optimal design
Sphere Intrusion = 16.2591 mm
Sphere End velocity = -55.63 mm/s
Bead position = 19 (Level 1)
Bead Angle = 9.729456° (Level 2)
Algorithm = Adaptive simulated annealing
Level 1: Total function calls = 20
Designs per iteration = 4, Iteration = 5
Level 2: Total function calls = 20
Designs per iteration = 4, Iteration = 5

Fig.13: Multi-level-Optimization without submodel, Metamodel-based, function calls = 400

The optimal design has a bead position 19 and bead angle of 9.73 degrees. The intrusion in this design is 16.26 mm. This is comparable to the results of the optimization in single level. The Multi-Level-Optimization with submodel now should also find the approximately the same bead position and bead angle. The results of Multi-Level-Optimization with submodel are discussed in subsection 3.3.3

3.3.3 Multi-Level-Optimization with submodel I

The Multi-Level-Optimization with submodel also carried out with total 400 function calls. The optimization setup is same as the Multi-Level-Optimization without submodel. But in this case the bead angle is optimized using a submodel in level 2. The results of the Multi-Level-Optimization with submodel are shown in figure in 14.

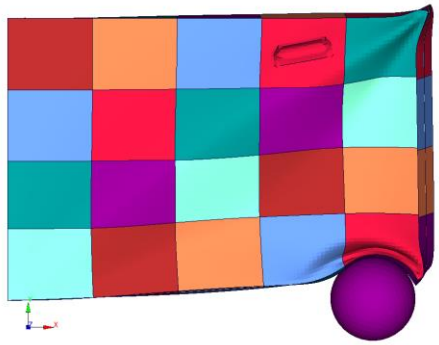


Optimal design
Sphere Intrusion = 17.6316 mm
Sphere End velocity = -51.09 mm/s
Bead position = 19 (Level 1)
Bead Angle = -45 ° (Level 2)
Algorithm = Adaptive simulated annealing
Level 1: Total function calls = 20
Designs per iteration = 4, Iteration = 5
Level 2: Total function calls = 20
Designs per iteration = 4, Iteration = 5

Fig.14: Multi-level-Optimization with submodel, Metamodel-based, function calls = 400

The optimal design shows that the Multi-Level-Optimization has found the bead position 19 and but the bead angle is -45 degrees. The intrusion is 17.63 mm. Compared to the results of the single level and multilevel without submodel this is a poor result. This is due to the inaccurate results delivered by the submodel I. The results of Multi-Level-Optimization with submodel II are explained in subsection 3.3.4.

3.3.4 Multi-Level-Optimization with submodel II



Optimal design
Sphere Intrusion = 16.2591 mm
Sphere End velocity = -55.63 mm/s
Bead position = 19 (Level 1)
Bead Angle = 9.729456° (Level 2)
Algorithm = Adaptive simulated annealing
Level 1: Total function calls = 20
Designs per iteration = 4, Iteration = 5
Level 2: Total function calls = 20
Designs per iteration = 4, Iteration = 5

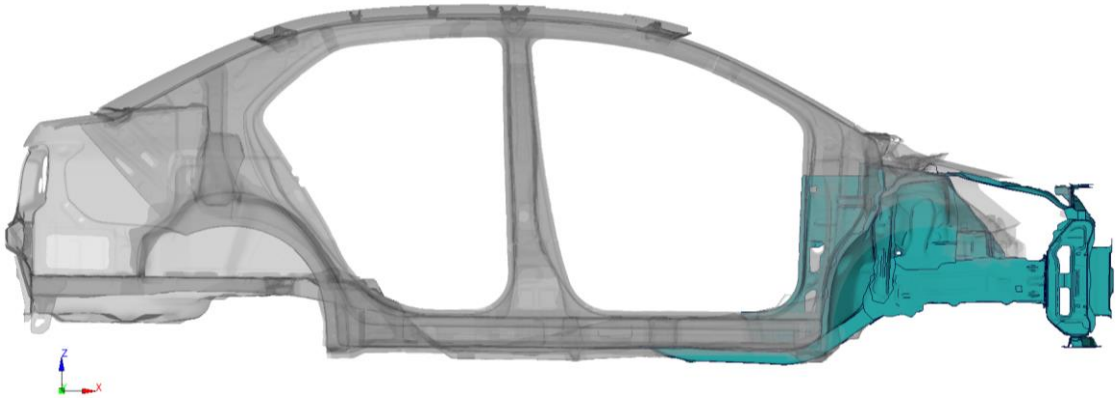
Fig.15: Multi-level-Optimization with submodel II, Metamodel-based, function calls = 400

The results show that the optimization with submodel II was successful, because it delivered better results than the submodel I. This proves that the interface boundary conditions play a very important role in the submodel optimization. The influence of these boundary conditions can be measured using criterion like the distance of bead to interface. There are other criterion which will be studied in further researches.

3.4 Industrial example: Toyota YARIS in crash

The industrial example discussed in this section is a Finite-Element crash model Toyota YARIS [6]. The submodel validation and the optimization with submodels is yet to be tested for this industrial example. The automatic submodel generation process based on the threshold and CI limit can be implied to the full car crash model. The figure 16 demonstrates the result of the automatic generation of a YARIS submodel (cyan) superimposed with main model of YARIS (grey) for a Euro NCAP front crash with 56 km/h against a rigid barrier. Figure 16a shows both the models at the simulation time of 0 ms and 16b shows the deformation in both the models at time 120ms.
The deformation in the submodel is identical to the deformation of same region in the main model. The optimization with such complex submodels delivers a great potential. Because the saving in the computational time plays an important role in the optimization of full car crash models. It is planned introduce new beads in the structure of the front pillars to optimize it for higher stiffness and weight reduction. Beads can be directly generated on the finite element mesh using the software generator [2]. This tool could be a helpful in the development process, if the user has access only to the finite element crash model. For more detail on the automatic generation of submodel refer to [1].

a)



b)

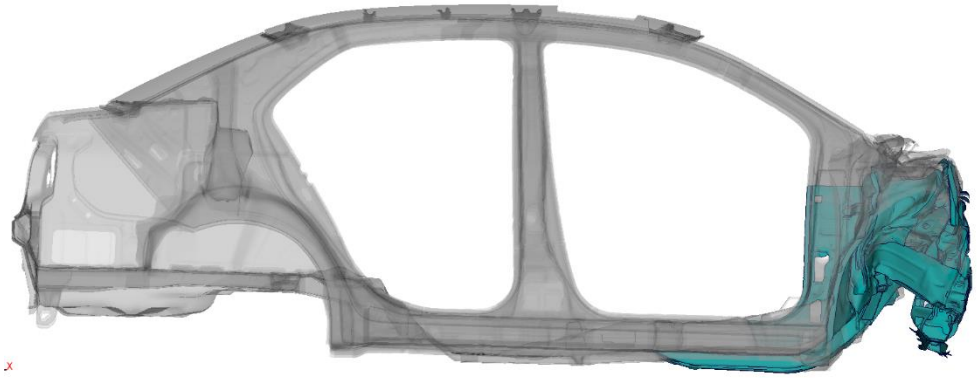


Fig.16: Toyota YARIS submodel in front crash at simulation time a) 0 ms b) 120 ms

4 Summary

The Multi-Level-Optimization process delivers a chance to fasten the optimization process. But the major challenge is the automatic generation and validation of the submodels. The automatic generation of the submodel is possible but the validation process is still under development. The advantage of using the Multi-Level-Optimization with submodels is to save computational time in the complete optimization process. In this paper work for the academic example, the time required by the optimization with submodels is 20% less in comparison to the Single-Level-Optimization. In comparison to the Multi-Level-Optimization without submodels the time saved is about 34%. The Multi-Level-Optimization process has a great potential for improvement. In future work more examples will be tested with the Multi-Level workflow.

5 Acknowledgement

This research was supported by the "Bundesministerium für Bildung und Forschung (BMBF)" within the scope of the KMU-innovativ research project "Entwicklung von Softwaremethoden zur effizienten Ersatzmodell gestützten Optimierung für die Crashauslegung im Fahrzeugentwicklungsprozess (eEgO)". Beside the University of Wuppertal, Automotive Simulation Center Stuttgart e.V., SCALE GmbH, GNS mbH, divis intelligent solutions GmbH and the Technical University of Munich are involved in the project.

6 Literature

[1] Falconi, C.J.; Walser, A.F.; Singh, H.; Schumacher, A. (2017): “Automatic generation, validation and correlation of the submodels for the use in the Optimization of crashworthy structures", to appear in the Proc of the 12th World Congress on Structural and Multidisciplinary Optimisation, 5th -9th, June 2017, Braunschweig

[2] The multipurpose Pre-Processor for FEM Analysis, GNS-mbH Gesellschaft für numerische Simulation (n.d.), retrieved from http://gns-mbh.com/generator.html

[3] The Trendsetting Post-Processor for FEM Analysis, GNS-mbH Gesellschaft für numerische Simulation. (n.d.), retrieved from http://gns-mbh.com/animator.html

[4] ClearVu Analytics, divis intelligent solutions GmbH, http://www.divis-gmbh.de, (2010-2017)

[5] Link, S; Singh, H.; Schumacher, A. (2016): “Influence of submodel size and evaluated functions on the optimization process of crashworthiness structures", LSDYNA Forum (2016)

[6] LS-DYNA-Model TOYOTA YARIS MODEL (NCAC V01): “National Crash Analysis Center (NCAC)”, http://www.ncac.gwu.edu/vml/models.html (2010)

[7] Schumacher, A. (2013): "Optimierung mechanischer Strukturen”, 2. Auflage, Springer Verlag, ISBN: 978-3-642-34699-6