A New Method for the Structural Optimization of Product Families

Larsgunnar Nilsson
Engineering Research Nordic AB
Brigadgatan 14, SE-58758 Linköping, Sweden
Larsgunnar.nilsson@erab.se
and
Linköping University
IEI/Division of Solid Mechanics
SE-58183 Linköping, Sweden
Larsgunnar.nilsson@liu.se

Michael Öman
Scania CV AB
SE-15187 Södertälje, Sweden
Michael.oman@scania.com

Abstract

This paper discusses the problem of structural optimization of product families subjected to multiple load cases, evaluated by computationally costly finite element analysis. Product families generally have a complex composition of shared components that makes individual product optimization difficult as the relation between the shared variables is not always intuitive. More optimal is to treat the problem as a product family optimization problem. For product families subjected to multiple and computationally costly crash loads, however, the optimization problem takes too long time to solve with traditional methods. Therefore, a new optimization algorithm is presented that decomposes the family problem into sub-problems and iteratively reduces the number of sub-problems, decouple and solve them.

1 Introduction

The increasing competition in the automotive market is pushing the vehicle manufacturers to satisfy all the different customer requirements from different markets and segments. This fact results in a wide range of products of low volume that still have to be offered at a competitive price. A cost effective way to manage this is to take a product family approach, see Meyer and Lehnerd [1]. The basic idea of the product family approach is to maintain as many common parts as possible in between the different product variants and only change those essential for the individual product performance, c.f. Simpson et al. [6].

Figure 1 illustrates how the parts can be shared in a product family of general commonality consisting of three products. The three domains represent the three products and the areas A, B and C represent the parts unique for each product. The area marked with ABC represents the product platform, i.e. shared by all three products, but parts can also be shared by only two products as represented by the areas AB, AC and BC.

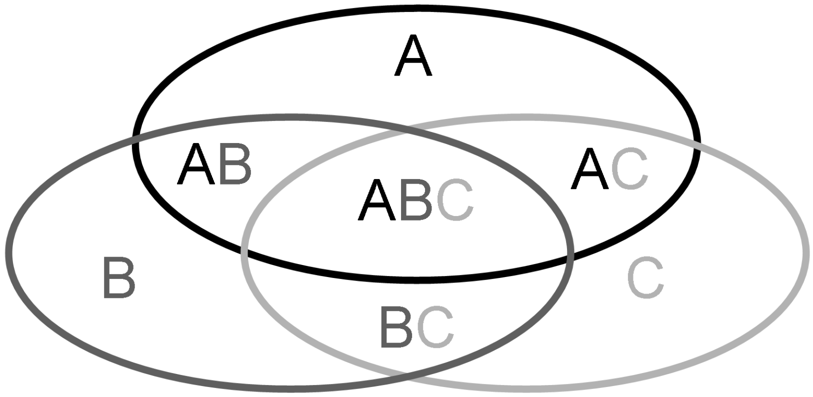

Fig 1 *Illustration of how the parts can be shared in a product family of three products with generalized commonality.*

A disadvantage with the product family approach is that an individual product may not be as optimal for its function as it would have been if only the individual product was optimized. It is therefore generally a balance between the performance lost and the cost saving in making a part shared.

A family of products is generally subjected to a number of load cases. In the design of automotive structures these are typically various crash situations and the torsion stiffness of the body structure.

What makes the structural optimization of a product family different from an optimization of only one product is the size and complexity of the problem? When considering one specific design variable, its influence on all load cases and requirements connected to the related product variants has to be considered. This fact makes optimization of product families demanding to perform. Apart from the number of design variables, the size of the optimization problem is decided by the number of product variants in the family and the number of load cases associated to each product variant. The large size of a product family optimization can be handled if the function evaluations are simple and fast to perform, but is a big concern if the evaluations are computationally costly, as is the case in crashworthiness analyses.

The objective of this work has been to reduce the size of the optimization problem by reducing the number of necessary function evaluations in order to make the problem solvable. This is done by identifying the constraints that are active in the optimal solution and only perform the function evaluations related to these critical constraints. Therefore the method has been named the Critical Constraint Method, CCM.

2 Optimization of product families

In the case of a product family the optimization problem can be formulated as

$$\min f(x) = \sum_{i=1}^{p} \alpha_i f_i(x_i) \quad i = 1,2,\ldots,p$$

$$\text{s.t.} \quad g_{ijk}(x_{ij}) \le b_{ijk} \qquad j = 1,2,\ldots,q \qquad k = 1,2,\ldots,r \tag{1}$$

$$x_t^- \le x_t \le x_t^+ \qquad t = 1,2,\ldots,n$$

where f is the objective function, x is a vector of n design variables with limit values x_t^- and x_t^+, $\boldsymbol{x}_i$ and $\boldsymbol{x}_{ij}$ are subsets of the vector $\boldsymbol{x}$ with the corresponding variables for a specific product and load case, g_{ijk} are the constraint functions subjected to the constraints b_{ijk}, p is the number of products in the family, q is the number of load cases and r is the number of constraints for a certain combination of products and load cases. Here the objective is defined as a weighted sum of the performances of each product in the family. The weight factors α_i can e.g. be based on the production volume or profit of the individual product "i".

In this work, two kinds of meta model approaches are used: Sequentially created linear polynomial meta models and successively updated radial basis function (RBF) network meta models. In the case of over-sampling a recommended number of 50% extra design points above the minimum number are added, see Roux et al. [2] and Redhe et al. [3], although previous work by Öman [4] indicates that over-sampling is expensive for product family problems. Linear Koshal and D-optimal designs have been used as design of experiment methods (DOE) for the linear meta models and Space-filling for the RBF networks. For applications of polynomial based meta models in crashworthiness problems, see Stander et al. [7] and Forsberg and Nilsson [8].

Iterative process

1. Decomposition of the family problem into sub-problems
2. System evaluation – On the product family level
3. Problem reduction – Identification of critical constraints
4. Decoupling – Decoupling by variable distribution
5. Sub optimizations – Solving the sub-problems

Fig 2 *Illustration of the iterative process of the decomposed family problem used by the CCM algorithm.*

For structural optimization of products exposed to multiple load cases analyzed by nonlinear FE-simulations, the generally applied approach is RSM based multidisciplinary feasible (MDF) methods, Stander et al. [9]. However, for large family optimization problems the number of necessary function evaluations grows rapidly and the required computational resources become huge. The presented CCM algorithm is an intention to tackle these problems by decomposing the problem into sub-problems, reducing the number of sub-problems by only considering the essential ones, and decouple the sub-problems by variable distribution, see Figure 2. The problem reduction and decoupling are done for each iteration based on a system evaluation performed on the family level.

3 Evaluated optimization methods

As the new CCM algorithm here make use of, and is also later compared to, the optimization software LS-OPT® which is based on MDF, see Stander et al. [9], the two methods are here described and the differences are identified.

3.1 Multidisciplinary feasible (MDF) using LS-OPT

The traditional MDF approach to solve the family optimization problem as one large optimization is used by LS-OPT. The different combinations of product variants and load cases are treated as disciplinary analysis returning constraint values to the optimizer. The procedure to solve the optimization problem defined by Equation 1, is illustrated in Figure 3.

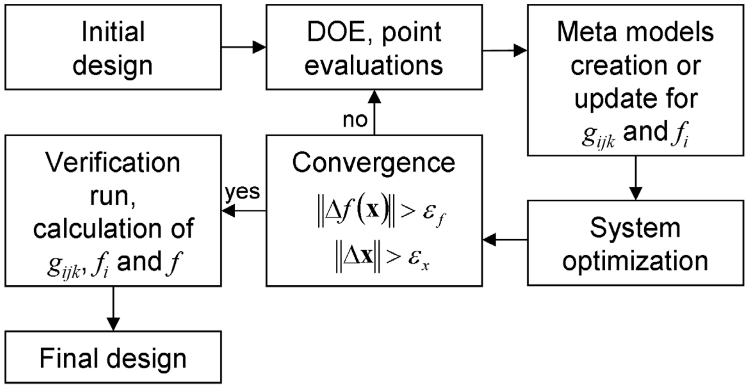

Fig 3 *Flowchart of the MDF optimization procedure for product families using LS-OPT.*

3.2 The Critical Constraint Method (CCM)

The basic idea of the new CCM algorithm is to reduce the number of costly true response evaluations and only perform the most relevant ones. This is done by decomposing the family problem into sub-problems and then iteratively reduce the number of sub-problems, decouple and solve them, see Figure 2.

The CCM algorithm is intended for families where the product variants are subjected to multiple load cases of similar nature. The combinations of product variants and load cases can then be organized and illustrated as a p x l matrix. Each row represents a product variant and each column represents a load case with the corresponding constraint functions, i.e. a load case can have more than one constraint. The family problem can then be decomposed into sub-problems, where each combination of product variant, *i*, and load case, *j*, represent a sub-problem, SP_{ij}. The sub-problems are coordinated at system level, giving a two level formulation as illustrated in Figure 4, where, *p*, is the total number of product variants and, *l*, the total number of load cases. The output from the system coordinator is the variable values, $\boldsymbol{x}_{ij}$, and the input from the sub-optimizations is the new variable values, $\boldsymbol{z}_{ij}$, the individual product weight, f_i, the constraint values, $\boldsymbol{g}_{ij}$, and the variables influence on the constraint functions, $\boldsymbol{e}_{ij}$. The link between the sub-problems is the value of the common variables that is coordinated at system level, the performance evaluation and problem reduction are based on the individual objective function values and the constraint violations, and the decoupling is based on the individual variable influence on the constraint functions.

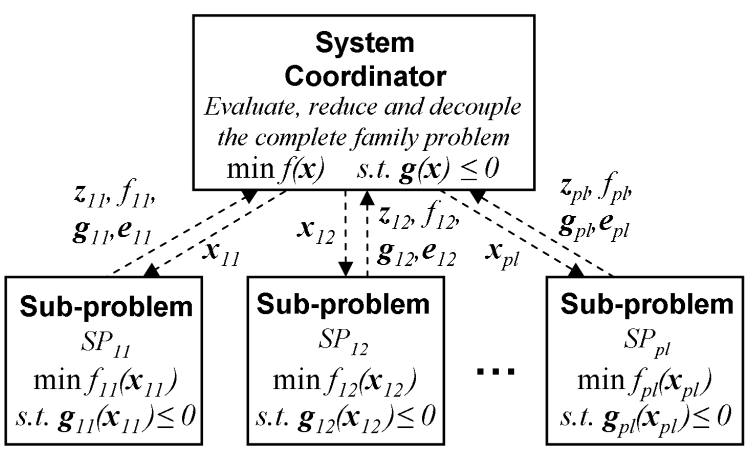

Fig 4 *Schematic illustration of the two level decomposed product family problem used by CCM.*

To reduce the size of the problem, only the sub-problems with active constraints in the optimal solution is considered. The active constraints are presumed to be within a set of critical constraints that is updated for each iteration. The critical constraints are identified as the constraints that are most violated or are closest to be violated. A set of critical sub-problems is then constructed with the combinations of product variants and load cases that correspond to the critical constraints. The true response evaluations and meta model creations are then only performed for these critical sub-problems.

To further reduce the costly true response evaluations, the sub-problems are decoupled and solved separately. This is made possible by distributing the common design variables, i.e. each design variable is only considered in one sub-problem and as a constant in the others. The assumption that the influence of a design variable on the constraint functions is dominant by one of the critical constraints is made. Therefore, the influence of the design variable on the other constraints is neglected and the design variable is only considered for the critical constraint for which it is identified as most important. This distribution of design variables is performed for each iteration, and it is here assumed that the internal energy of the component corresponding to a design variable indicates the requested influence of the variable, c.f. Forsberg and Nilsson [8] and Goel and Stander [12].

To evaluate the CCM algorithm, a steering program was written in Python. The sub-optimization problems are solved using LS-OPT, and the true response evaluations are solved by using the FE program LS-DYNA, see Hallquist [10]. The procedure to solve the family optimization problem by the CCM, is illustrated in Figure 5.

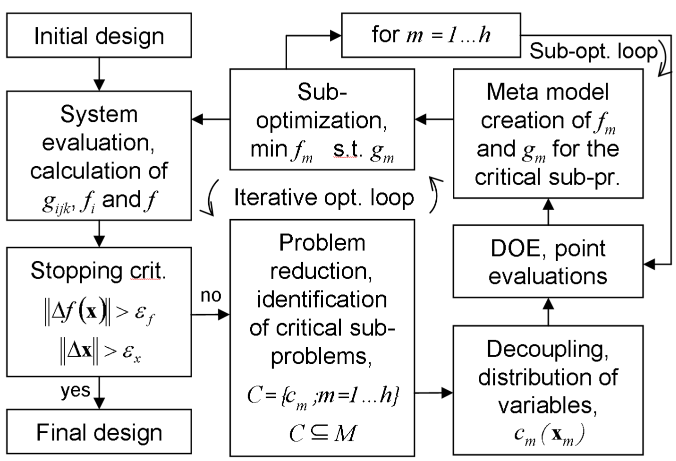

Fig 5 *Flowchart of the CCM optimization procedure for product families.*

4 Examples

In this work the total number of true response evaluations after convergence is used to evaluate the efficiency of the optimization method. The quality of the final solution is also of great interest. Is the presented method capable of finding the same solution as conventional methods although the problem has been simplified? To evaluate the CCM algorithm, a product family optimization problem was created and solved using various meta model methods and by the CCM algorithm. For reasonable computing times, the individual evaluation time for the load cases is reduced to a minimum. The crash models are therefore highly simplified compared to models in industrial practice, c.f. Lönn et al. [11], and the problems are solved using LS-DYNA within 10 to 20 seconds on one single processor.

4.2 Family of truck cabins

The simplified product family optimization problem consists of four truck cabins that are exposed to four different load cases. The truck cabins are composed in such a way that four different sizes are created from ten different design parts. The composition of the family of truck cabins is illustrated in Figure 6. The black areas represent the openings in the structures intended for the windows.

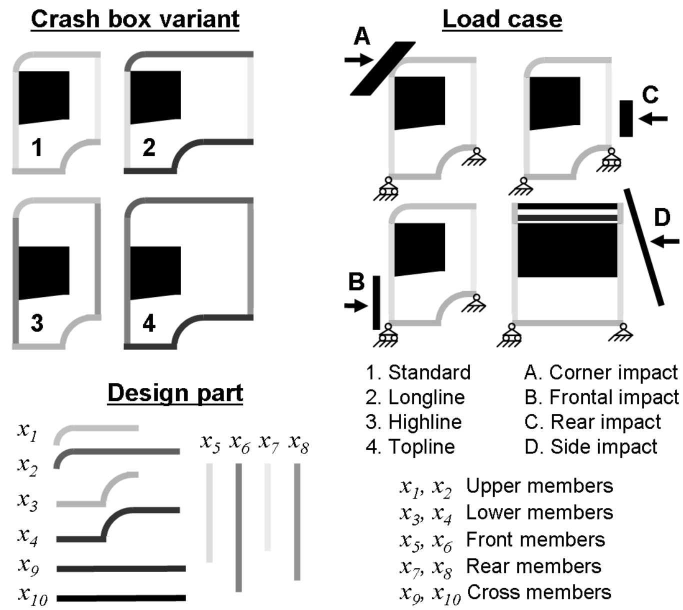

Fig 6 *Composition of the family of truck cabins subjected to four load cases.*

The four cabins are composed in the same manner but the longitudinal and vertical members exist in two lengths that can be combined to form different cabin sizes (length, hight): one small (1.4 m, 2.0 m), one high (1.4 m, 2.4 m), one long (1.9 m, 2.0 m) and one large (1.9 m, 2.4 m). The width of the cabins, 2.0 m, is identical for all cabin variants and is determined by the cross members. All members have a quadratic cross section of 50x50 mm^2 and are given the same material model. The cabins are also constructed of a steel plate floor, a roof and walls of 0.5 mm thickness, which all use the same material model. Only the thicknesses of the members are considered as design variables in this optimization problem.

The load cases are set up to resemble the actual legal requirements for trucks. The upper corner impact consists of a large rigid plate impacting horizontally with a kinetic energy of 30 kJ. The frontal impact plate hits the full width of the lower front of the cabin with a kinetic energy of 30 kJ. In the case of the rear impact the plate is 400 mm high and 1000 mm wide and it hits the center of the rear wall with a kinetic energy of 30 kJ. Also the side impact consists of a rigid plate, which is at 20 degrees angle and hits the cabin from the side with a kinetic energy of 17 kJ. The intrusion is constrained to a maximum level in all load cases, see Equation 2. The intrusion is measured as the total horizontal displacement of the plate. The boundary conditions used for each load case are indicated in Figure 6.

The objective is to fulfill the load case requirements at the lowest possible total product family weight. The weight is calculated as the sum of each individual cabin weight multiplied with a weight factor α. The weight factor can be set to reflect the number of produced cabins of each type etc. Here it is equal to one for all cabin variants. The optimization iterations are stopped when the change in the optimal value ε_f and design variables ε_x both are below 0.01.

4.2.1 Problem size and complexity

The product family optimization problem of truck cabins can be illustrated in a two dimensional matrix, see Figure 7, where the design variables are the thicknesses of the 10 design parts.

Product variant \ Load case	A	B	C	D	Corresponding variables
1	$\mathbf{x}_1$	$\mathbf{x}_1$	$\mathbf{x}_1$	$\mathbf{x}_1$	$\mathbf{x}_1 = (x_1, x_2, x_5, x_6, x_9, x_{10})$
2	$\mathbf{x}_2$	$\mathbf{x}_2$	$\mathbf{x}_2$	$\mathbf{x}_2$	$\mathbf{x}_2 = (x_1, x_2, x_7, x_8, x_9, x_{10})$
3	$\mathbf{x}_3$	$\mathbf{x}_3$	$\mathbf{x}_3$	$\mathbf{x}_3$	$\mathbf{x}_3 = (x_3, x_4, x_5, x_6, x_9, x_{10})$
4	$\mathbf{x}_4$	$\mathbf{x}_4$	$\mathbf{x}_4$	$\mathbf{x}_4$	$\mathbf{x}_4 = (x_3, x_4, x_7, x_8, x_9, x_{10})$

Fig 7 *Two dimensional illustration of the product family optimization problem of truck cabins.*

4.2.2 Problem formulation

The weight optimization problem of the family of truck cabins can be formulated

$$\min f(x) = W_{tot}(x) = \sum_{i=1}^{4} \alpha_i f_i(x_i) \qquad i = 1,2,3,4$$

$$\text{s.t.} \quad g_{ij}(x_i) \le b_j \qquad j = A,B,C,D \qquad (2)$$

$$x_t^- \le x_t \le x_t^+ \qquad t = 1,2,\ldots,10$$

$$\text{where} \quad b_A = 250mm \;\; b_{\mathbf{B}} = 75mm \;\; b_C = 75mm \;\; b_D = 30mm$$

$$x_t^- = 0.5mm \;\; x_t^+ = 3.0mm \;\; \alpha_1,\ldots,\alpha_4 = 1$$

W_{tot} is the total weight of the product family and is calculated as the sum of the weights for the individual products, W_i, multiplied with the weight factor α_i=1.

4.2.3 Product family efficiency

To evaluate the performance loss using the product family design technique, the products were also optimized as isolated individuals. The weight difference between the individual optimized products and the ones optimized as a family is presented in Table 1.

	Individual optimized products [kg]	Product family optimization [kg]	Product efficiency loss [kg]/[%]
1. Standard	251	256	5 /2
2. Highline	302	307	5 /1
3. Longline	287	293	6 / 2
4. Topline	328	349	21 / 6
Total family weight	1168	1204	36 / 3

Table 1 *Individual product weights and efficiency loss using the product family approach.*

The most affected product is the Topline cabin, with a weight increase of 6%, when it has to share parts with the rest of the product family. For the total weight of the four cabins the weight is increased by 3% using the product family approach. This loss in performance always needs to be compared to the cost savings in production as a final evaluation of the suitability of the product family approach for each individual application. Since the optimized structural parts only represent about half of the total cabin weight, the loss in performance is here only a few percent.

4.2.4 Optimization method efficiency

The product family optimization problem was solved using four meta model methods and by the CCM algorithm. The results from the various optimizations are presented in Table 2. The lowest optimum value of 1204 kg is found by the two methods using linear meta models. The one with 50% over-sampling finds the same optimal point for both starting points while the one without over-sampling finds two different solutions but with the same optimal value. The method using an RBF meta model with 50% over-sampling finds an additional local optimum with the similar optimal value of 1205 kg for both starting points. Without over-sampling the RBF method strides towards the same optimum as found by the linear approximations but it is trapped by local optima of value 1213 kg and 1206 kg, respectively, for the two starting points.

Optimization method	LS-OPT								CCM			
RSM	Linear polynomial				RBF network				Linear polynomial			
DOE	Koshal		D-opt 50%		Space-filling		Space-f. 50%		Koshal		D-opt 50%	
Starting point	1	2	1	2	1	2	1	2	1	2	1	2
Result												
No. of iterations	12	12	13	13	20	20	20	20	12	11	15	13
No. of evaluations	1384	1384	2304	2304	2256	2256	3536	3536	515	501	925	745
Optimal total weight	1204	1204	1204	1204	1213	1206	1205	1205	1206	1203	1205	1205
Variable thickness												
1. A-pillar standard	0.78	0.74	0.73	0.74	0.71	0.78	0.75	0.72	0.64	0.54	0.62	0.62
2. C-pillar standard	1.19	1.19	1.19	1.18	1.17	1.07	1.19	1.24	1.29	0.76	1.36	0.97
3. A-pillar high	1.16	1.26	1.13	1.11	0.98	1.05	1.93	1.93	1.03	0.50	0.83	0.62
4. C-pillar high	1.58	1.31	1.54	1.57	1.80	1.55	1.93	1.93	1.61	1.12	1.87	1.85
5. Upper standard	1.75	1.12	1.92	1.92	1.95	2.01	0.50	0.53	1.89	2.54	2.10	1.91
6. Lower standard	1.84	1.57	1.94	1.94	1.93	2.04	1.24	1.19	2.03	3.00	1.74	1.98
7. Upper long	1.19	0.50	1.25	1.25	1.23	1.30	1.12	1.14	1.22	1.38	1.24	1.18
8. Lower long	1.31	1.18	1.32	1.31	1.32	1.31	1.57	1.55	1.32	1.36	1.32	1.33
9. Crossmemb. frontal	0.55	1.99	0.54	0.52	0.78	0.50	1.25	1.25	0.63	0.50	0.63	0.97
10. Crossmember rear	1.37	1.91	1.22	1.23	1.08	1.21	1.31	1.31	1.17	1.21	1.13	1.25
Constraint violation												
Corner impact [max%]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Frontal impact [max%]	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.1	0.0
Rear impact [max %]	0.0	0.1	0.0	0.0	0.3	0.4	0.0	0.0	0.1	12	2.8	6.1
Side impact [max %]	0.5	0.4	0.0	0.1	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.1

Table 2 *Optimization results for the family of truck cabins optimization problem using standard meta model based methods in LS-OPT and the CCM algorithm.*

The true response surface clearly contains many local optima with similar values. Although, the four meta model methods find different optimal design points the total family weight is almost the same for each of them. As can be seen in Table 2, some smaller constraint violations are also present for the final solutions.

If the number of costly evaluations at convergence is considered, the linear meta models without over-sampling stands out as the fastest method. A total number of 1384 evaluations is needed. With 50% over-sampling the number increases to 2304 evaluations and for the generally recommended RBF method with 50% over-sampling a total number of 3536 evaluations is needed. The optimization histories for the four optimization methods with different starting points are illustrated in Figures 8 and 9.

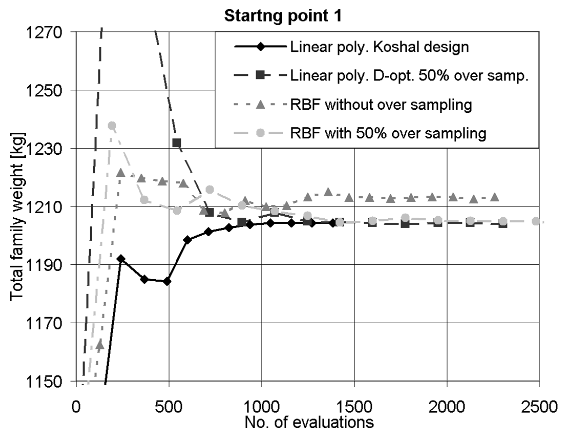

Fig 8 *Optimization history diagram for the standard meta model based methods using LS-OPT*

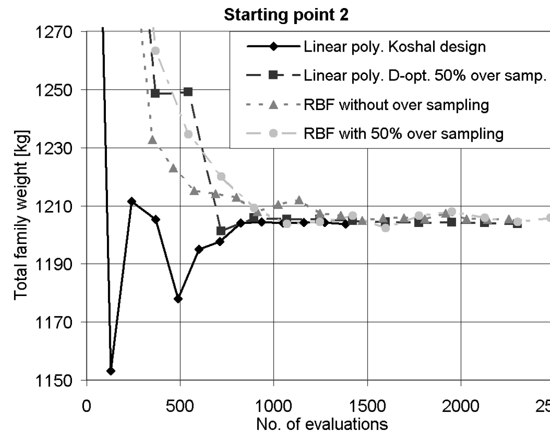

Fig 9 *Optimization history diagram for the standard meta model based methods using LS-OPT*

The product family optimization problem is also solved by the CCM algorithm. For all performed optimization runs, different local optima are found giving a total family weight of 1203-1206 kg. Hence, the total family weight is about the same as found from the standard meta model methods. However, for the two optimizations starting from point 2, constraint violations is observed. The reason is that the violated constraints are not identified as critical until late in the optimization process. This problem can occur when the initial design is far from the optimal design, resulting in a different initial global behavior.

Considering the number of evaluations needed to fulfill the convergence criteria, the number is significantly reduced for the CCM algorithm compared to the standard meta model methods. The fastest method is the one without over-sampling, which stops already after about 500 evaluations for the two different starting points. With 50% over-sampling, this number increases to about 900. If the fastest methods for the two optimization strategies are compared, the number of required evaluations is reduced by 62% for starting point 1 and 63% for starting point 2, using the CCM algorithm. Compared to the recommended RBF network with 50% over-sampling, the reduction is 85%. The optimization history for the CCM algorithm is illustrated in Figure 10.

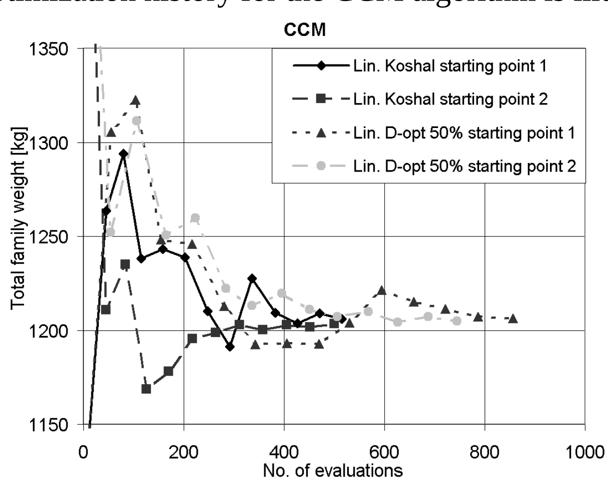

Fig 10 *Optimization history diagram for the CCM algorithm.*

Conclusions

Most product family optimization problems in industrial practice are expensive to solve and the computing resources are generally not available to solve many large problems at a reasonable time with existing optimization programs. Therefore, the reduction of the required number of evaluations was the first priority in the development of the presented CCM algorithm.
The new CCM algorithm proves to be a very efficient optimization algorithm for product family structural optimizations. The large reduction of the required number of evaluations can be the difference that makes problems practically solvable. The constraint violations might be a concern in many applications and have to be considered in further development of the CCM algorithm. Possible remedies might be to let the sub-optimizations run for more than one iteration to produce local solutions of less constraint violations or to implement a more efficient algorithm for altering the region of interest. For further details on the CCM algorithm, see Öman and Nilsson [5].
When developing the CCM algorithm, large and complex product family problems with many load cases were in mind. The algorithm has been shown to be efficient on a small demonstrating problem but we believe it should be even more efficient on a real automotive product family structural optimization problem.

References

[1] Meyer, M. and A. Lehnerd (1997). The power of product platforms: Building value and cost leadership. New York: The free press.
[2] Roux, W., N. Stander, and R. Haftka (1998). Response surface approximations for structural optimization. International Journal for Numerical Methods in Engineering 42, 517–534.
[3] Redhe, M., J. Forsberg, T. Jansson, P. Marklund, and L. Nilsson (2002). Using the response surface methodology and the d-optimality criterion in crashworthiness related problems. Structural and Multidisciplinary Optimization 24, 185–194.
[4] Öman, M. (2007). Structural optimization of product families exposed to crash loading. 6th European LS-DYNA Users Conference, 1:57–1:65, Gothenburg.
[5] Öman, M., Nilsson, L. (2010). Structural optimization of product families subjected to multiple load cases. Structural and Multidisciplinary Optimization, 41:797-815.
[6] Simpson, T., Z. Siddique, and J. Jiao (2006). Product platform and product family design: Methods and applications. New York: Springer.
[7] Stander, N., W. Roux, M. Giger, M. Redhe, N. Fedorova, and J. Haarhoff (2004). A comparison of metamodeling techniques for crashworthiness optimization. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA 2008–4489, New York.
[8] Forsberg, J. and L. Nilsson (2007). Topology optimization in crashworthiness design. Structural and Multidisciplinary Optimization 33, 1–12.
[9] Stander, N., W. Roux, T. Eggleston, and K. Craig (2008). LS-OPT User’s Manual, Version 3.3. Livermore Software Technology Corporation, Livermore.
[10] Hallquist, J. (2006). LS-DYNA Theory Manual. Livermore: Livermore Software Technology Corporation.
[11] Lönn, D., M. Öman, L. Nilsson, and K. Simonsson (2009). Finite element based robustness study of a truck cab subjected to impact loading. International Journal of Crashworthiness 14(2), 111–124.
[12] Goel, T. and N. Stander (2008). Influence of selection criterion on rbf topology selection for crashworthiness optimization. 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA 2008–5997, British Columbia.