Shape Optimization for Improved Vehicle Safety and Reliability

R. Sidhu[a], J. Burgueño[a], R.C. Averill[b], E.D. Goodman[b]

[a]Red Cedar Technology
4572 S. Hagadorn Rd., Suite 1-E
East Lansing, MI 48823

[b]Department of Mechanical Engineering
Michigan State University
East Lansing, MI 48824

Abstract: Advancements in high-performance computing and in nonlinear dynamic structural simulation software such as ABAQUS have made it possible to virtually test potential designs prior to building and testing expensive prototypes. These tools alone, however, still require an engineer to develop a design based on intuition and numerous time-consuming and error-prone iterations. The next level of advancement is software to automate the process of iterating over a large number of design scenarios and intelligently seek optimal values for those parameters that strongly affect product performance and cost. While many design optimization approaches are limited to a small number of continuous design variables, the approach described here leads to a productive search over hundreds of variables at a time. This capability has been implemented in a software product called HEEDS (Hierarchical Evolutionary Engineering Design System). HEEDS uses multiple autonomous agents to hierarchically decompose a problem into subsets of highly decomposed overlapped relationships. Decomposition is effected by using different numbers of design variables, different levels of design variable dicretization, and/or other problem-specific divide-and-conquer rules. The system combines evolutionary search algorithms with local optimization techniques. Using Abaqus/Explicit as the finite element solver within the HEEDS optimization environment, this process has been applied to several automotive lower compartment rail designs, resulting in significant gains in performance along with up to 20% reductions in mass compared to baseline rails designed by experienced engineers. An example application of this method is described herein. A second example demonstrates the shape optimization capabilities of HEEDS when used in conjunction with ABAQUS/CAE.

1. Introduction

The design of modern vehicle structures is driven by many competing criteria, such as improved safety and fuel efficiency, lower cost, enhanced performance, and increased style flexibility. In addition, the introduction of new manufacturing processes and materials significantly increases the available design space, or the set of all possible designs for a problem. In order to explore this large design space more effectively while trying to reduce design cycle times, engineers can now take advantage of automated design optimization and simulation software tools. These tools can greatly decrease the time required to identify a set of feasible, or even near-optimal, designs prior to building and testing the first prototype. Moreover, these tools also overcome the limitations of human intuition and allow the designers freedom to seek creative solutions that are not obvious to even the most experienced engineer. This is true in general but particularly true with shape optimization problems, which can involve potentially hundreds of design variables.

In general, shape optimization can provide drastic improvements in the performance of a structure. However, due to the prohibitively large number of design variables involved in some shape optimization problems, most optimization techniques are unable to solve these problems. Another impediment to shape optimization is creating a mesh that will be reasonable for all possible forms within the design space, which can be a difficult task. Techniques that change the nodal coordinates without remeshing the structure incur problems in which the elements can become distorted or even fold over on themselves as shape variations get further away from the initial form. Remeshing at each evaluation is a necessity as shape variations become large. Parametric modeling overcomes this problem, allowing significant variations to the initial design without running into the problems of mesh distortion. Using shape optimization to design crashworthy structures increases the complexity of the problem even further due to the dynamic nature of the crash problems, resulting in a multi-modal, non-convex design space.

Optimizing multifunctional, energy-absorbing structures in a vehicle provides a challenge to safety engineers and to automated design techniques. For example, energy-absorbing structures should maintain their rigidity while carrying the anticipated in-service loads and while serving as primary mounting locations for numerous functional devices and attachments, such as the engine in an automobile or a passenger seat in a helicopter. Yet these same structures must collapse in a prescribed manner during a crash to maximize the amount of energy absorbed by the structure and limit the forces transmitted to passengers.

Energy-absorbing structures often take the form of thin-walled tubular metallic structures subjected to dynamic compressive loads. In this case, energy is absorbed primarily through plastic deformation of the material and friction due to surface contact. The ideal mode of failure is one of progressive short-column buckling, which maximizes plastic material deformation and folding contact. The design of energy-absorbing tubular structures must ensure that their collapse or buckling mode is not sensitive to expected variations in material properties, wall section thickness, cross-sectional shapes, or overall tube curvature. The structure should also be robust enough to absorb similar amounts of energy under a wide variety of off-axis dynamic loading scenarios.

Crashworthiness problems are characterized by a very complex design space with many peaks and valleys due to their highly dynamic nature. These classes of structural design problems have a very multi-modal, non-convex design space and do not lend themselves well to classical gradient techniques. Moreover, objectives and constraints related to crash energy management, stiffness, strength, and packaging are joined by additional requirements on manufacturability, noise and vibration, mass reduction, and robustness against process and material variation. These objectives compete strongly against one another, making this a very challenging multi-objective optimization problem. Finally, the structure should be somewhat insensitive to slight variations in design variables. For example, in problems involving stability or buckling, behavior can be very sensitive to geometrical and material imperfections, which may prevent a part from failing in the way that it was intended. Therefore, it is not sufficient to find a design that performs well under a set of narrowly defined objectives, constraints, and loading conditions.

2. Optimization strategy

Many optimization studies and even mathematical proofs have shown that no single optimization approach performs best on all classes of problems, while combining a set of global and local optimization techniques often yields improved results (Koch et al., 2002). This often creates significant confusion and can be misleading for the optimization newcomer. In addition, for problems that contain many design variables and criteria, it is sometimes necessary to decompose the overall problem into a set of smaller, more tractable problems to obtain improved results with reasonable computational resources. This capability has been implemented in a software package called HEEDS (Hierarchical Evolutionary Engineering Design System). HEEDS has been constructed to allow hierarchical decomposition of problems while automatically combining the strengths of global exploration and local optimization algorithms. It combines the strengths of genetic algorithms (Holland, 1975), simulated annealing (Ruthenbar, 1989), sequential quadratic programming (Schittkowski, 1985), design of experiments (Cochran et al., 1992), and response surface methodology (Khuri et al., 1996), which jointly allow it to efficiently solve a wide class of problems.

HEEDS employs adaptive autonomous agents that communicate but work semi-independently on a common problem. Each agent uses one or more search methods to intelligently scan a part (or all) of the design space in search of optimal solutions. Using this methodology, the overall problem can be hierarchically decomposed so as to provide most agents different, smaller representations of the problem. A set of design variables at a given level of discretization constitutes one representation of the problem. Each agent (or a group of agents) can use a different representation of the problem to increase the efficiency of the design search. Furthermore, each agent (or group of agents) can employ specialized search heuristics that seek to maximize the performance of its representation of the problem. This approach combines the evolutionary search methods with the traditional local search methods to increase the effectiveness of the search process for a broad class of problems. Using this approach, agents can search for good designs in different representations of the design space and/or using different sets of constraints and objectives, communicating with each other to evolve more globally optimal designs through structured sharing of information. The hierarchical decomposition allows a complex problem to be discretized into a set of highly decomposed, overlapping relationships, which have a reduced design space that is relatively easy to search. The agents seeking good designs for these smaller problems provide information to the agents searching the complete design space with all the objectives and constraints, resulting in an economically efficient search process. This approach significantly reduces the total number of design iterations required for finding excellent solutions, and allows efficient handling of very complex optimization problems, which would otherwise be impractical to solve.

By using a stochastic strategy during the search for an optimal design, HEEDS is able to account for slight variations in design variables, thus insuring robustness and reliability. This results in designs that are tolerant of slight changes in the design parameters without suffering drastic drops in performance of the designs HEEDS can be run on a single processor or a loosely connected network (cluster) of personal computers or workstations.

3. Shape optimization of a lower compartment rail

3.1 Problem definition

A lower compartment rail was designed for the offset load-case using HEEDS as the optimization software and Abaqus/Explicit to evaluate each potential design created by HEEDS. The section thickness of the rail and the cross-sectional shape at different points along the length of the rail were used as design variables. Figure 1 shows the basic geometry of the rail, as well as the location of the 11 control cross-sections located along the length of the rail. The arrows on the control points on the cross-section indicate the directions in which the control points (controlling b-splines) were offset to create new shapes during the optimization process. The objective of the optimization problem was to maximize the crush-zone energy in the front of the rail (the crush zone shown in Figure 1) for the offset crash scenario with inequality constraints on the peak rigidwall force and mass. A component level simulation of a vehicle front rail was developed for the offset crash scenario to serve as a template for the creation of potential designs. For this analysis, a lumped mass equal to approximately one-half the mass of the vehicle was placed at the rear of the rail and offset 60 mm in the negative Y direction from the rail axis to simulate the offset crash scenario (see Figure 1). The rail structure and mass were given an initial velocity and crushed into a rigid stationary wall as depicted in Figure 1. The crush-zone energy, peak rigidwall force, and mass were computed with Abaqus/Explicit. HEEDS was then used to automate the creation and evaluation of each potential design to perform design optimization.

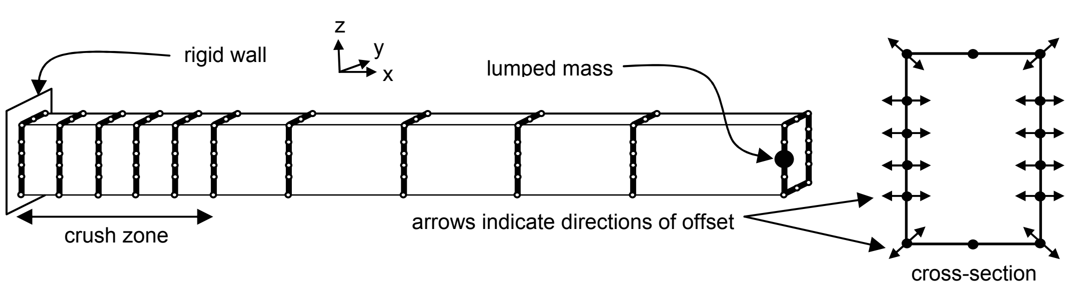

Figure 1. Geometry of baseline rail with control cross-section positions.

3.2 Problem setup

Two different runs of the same problem were performed to demonstrate the increase in the efficiency of the search process using hierarchical decomposition. The first run was performed using a single agent. The agent searched in a design space based on 67 design variables (66 control point shape variables and 1 thickness variable). Evolutionary search methods in combination with local search techniques were used to intelligently explore the design space.

The design variables for the problem are shown in Figure 2. The rail is created using a ruled surface and automatic mesh generator within HEEDS. The main design variables are the control points on the cross-sections along the length of the rail. The cross-sections are shown in Figure 1. All cross-sections have the same initial rectangular shape. The control point offsets on the cross-sections are then varied independently to create new rail shapes. Since a new mesh is created every time the rail shape is changed, the problems of mesh distortion are minimized. Master-slave conditions, which link control points together, were defined to create symmetry in the cross-sectional shape for this problem. A master-slave condition binds the offset value of the slave control points to that of the master control point. Due to the definition of the master-slave conditions, there are only 6 design variables per cross-section. This results in a total of 66 shape design variables (6 each for the 11 cross-sections along the length of the rail) for this representation of the problem. Control points 7 and 14 were not varied during these optimization runs. The master-slave conditions used for this run are shown in Figure 2.

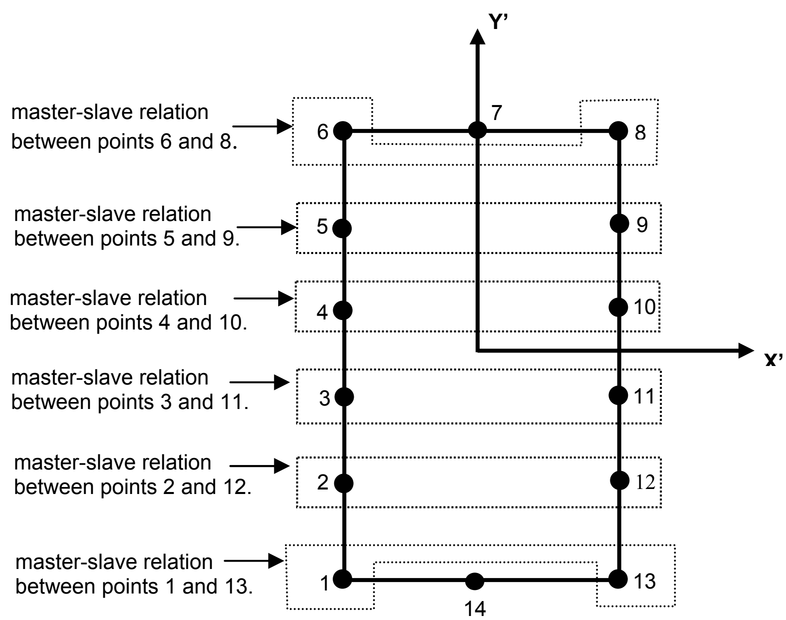

Figure 2. Cross-section definition for refined representation.

The agent seeks to maximize the internal energy absorbed in the front region of the rail during the first 14 milliseconds of the crush. Inequality constraints are imposed on the total mass of the rail and on the maximum normal force on the rigid wall.

A second run was conducted using hierarchical decomposition of the level of discretization of the design variables and the period of crush. A total of three autonomous agents were used for this second run. The topological structure used for hierarchically decomposing the problem is shown in Figure 3. Designs are shared periodically in a structured manner from agent-to-agent according to the arrows in Figure 3. Agent 0 shares designs with Agent 1, and Agent 1 shares designs with Agent 2. Each agent was executed as a separate computer process on a loosely coupled network of personal computers. They independently sought a set of good designs for each single technical objective and constraint set, using design variables at different levels of resolution. The designs from the agents working at coarser levels were fed into the agents working on a more refined search space to expedite the search process in those agents. This allows for economical emergence of solutions at a more refined level of discretization that satisfy all constraints and are driven by all technical objectives.

Agent 0 seeks designs in a representation of the design space that has 67 design variables (66 control points and 1 gage thickness) with each variable discretized in a coarse manner to reduce the size of the overall design space. In other words, the number of possible values that each design variable could be assigned was smaller in agent 0 than in the other agents. Agents 1 and 2 represent the overall problem with the same 67 design variables at a medium and refined level of discretization, respectively. A mapping of the control point design variable decomposition is depicted in Figure 3.

The agents 0 through 2 seek to maximize the amount of crush-zone energy in the front of the rail for crash scenarios with inequality constraints on the peak rigid wall force and the mass. The problem is further decomposed by considering different periods of crush time for each agent (6 ms for agent 0, 10ms for agent 1and 14 ms for agent 2). Due to this topology setup, Agent 0 quickly (due to smaller crush time as well as a coaser design space) finds good designs within its coarse design space. The designs found in agent 0 have good crush initiators near the front of the rail and are resistant to buckling in the rear of the rail. Agent 0 then shares the information from its search process with Agent 1 (which is simultaneously searching for designs in a more refined design space), pointing it to the regions in the design space with potential for good designs. This results in a significant speedup of the search process as compared to the agent searching only at the highest level of the problem representation.

Though not employed in the current example, many other techniques exist to decompose design problems within HEEDS. For example, HEEDS also allows the number of design variables to be different in each agent, often a very effective method of problem decomposition.

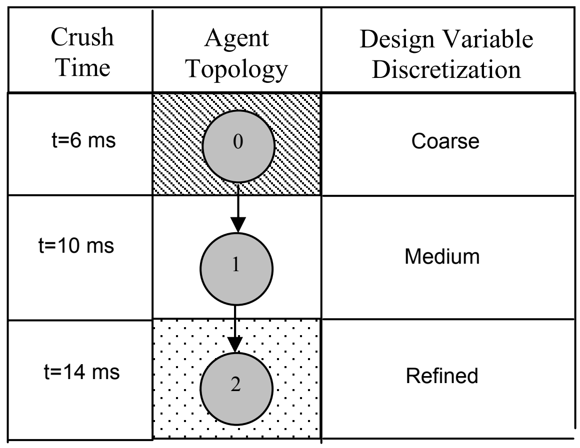

Figure 3. Topological structure used for the 3-agent run.

3.3 DISCUSSION OF RESULTS

Many high-performance designs were found during the runs, since HEEDS evolves a set of designs over a number of cycles. Figures 4 and 5 depict the undeformed geometry and the animation of crush, respectively, for the best overall design found by HEEDS for the offset load case. The design crushes progressively in an “accordion” fashion, primarily due to the structure’s shape. These progressive short-column buckling modes of crush depicted in Figure 5 are inherently robust against off-axis dynamic load cases. Energy is absorbed primarily through plastic deformation of the material during the progressive deformation. These accordion-like deformation modes help to maximize the plastic material deformation and folding contact during crash scenarios.

Figures 6 through 8 show the variation of the objective and constraints with respect to the number of design evaluations performed during the search process. The results of the search from the single agent run are compared with the results from the multi-agent run (using decomposition). The results of the three-agent run are plotted for the agent searching over the same design space as the single agent. Figure 8 illustrates the savings in terms of the number of evaluations required to find good designs when hierarchical decomposition is used in a problem. In the multi-agent run, searching at low levels of resolution identifies high-performance solutions very quickly. These solutions are then injected into agents searching at higher resolution for solution refinement. On the other hand, in the single agent run the agent has to search the entire design space autonomously, and thus requires a greater number of iterations to identify good designs. Both design runs were executed for the same number of cycles, and appear to have converged. The multi-agent run found solutions that absorbed significantly more energy than did the best solution found by the single agent run. This illustrates the ability of a multi-agent approach to search the design space more broadly, continuously identifying fruitful regions for search.

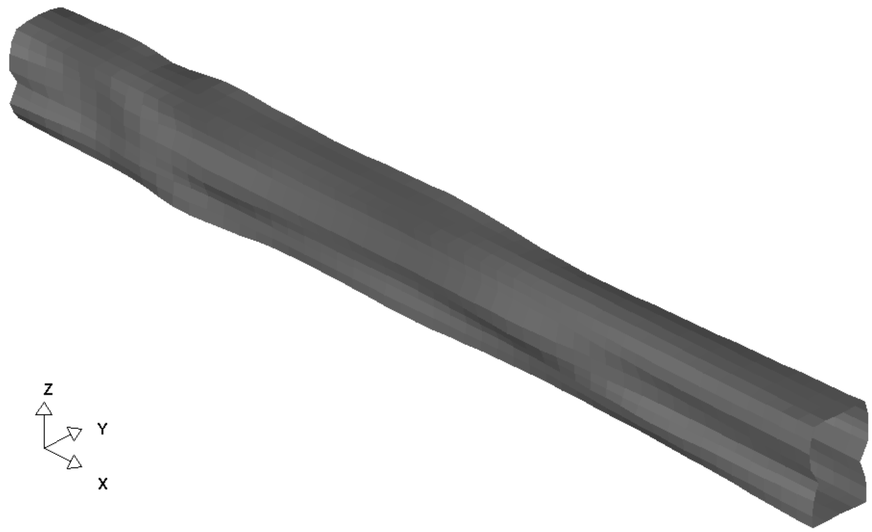

Figure 4. Geometric shape of best rail design found.

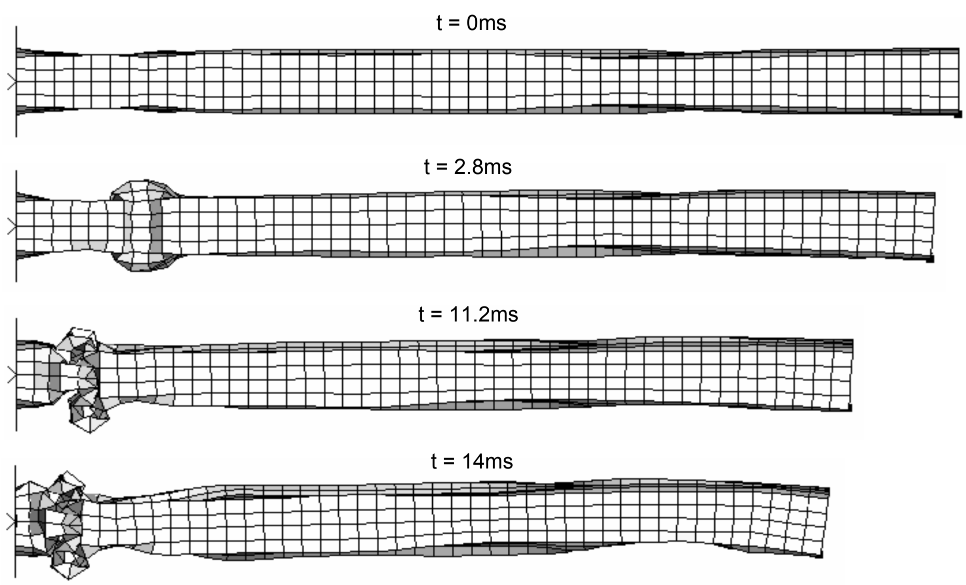

Figure 5. Progressive crush of rail under offset loading.

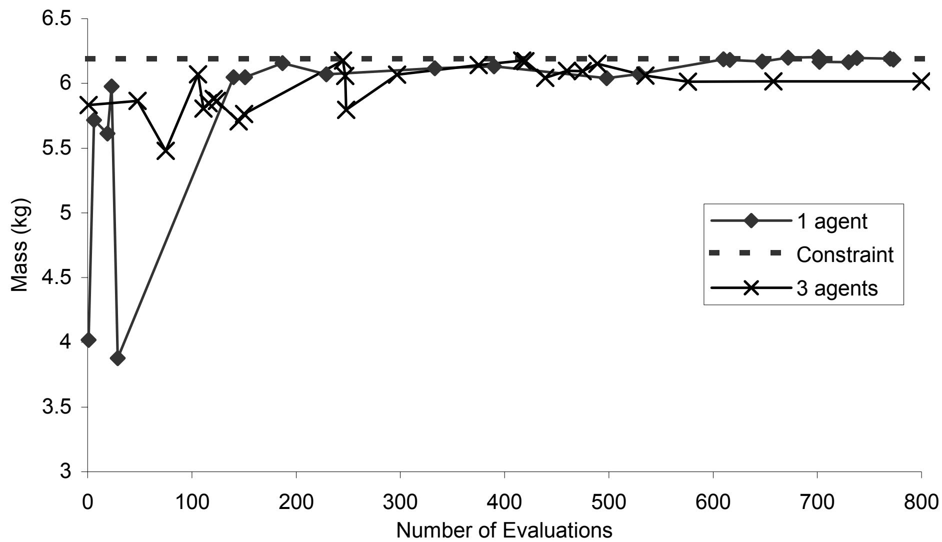

Figure 6. Trajectory of mass.

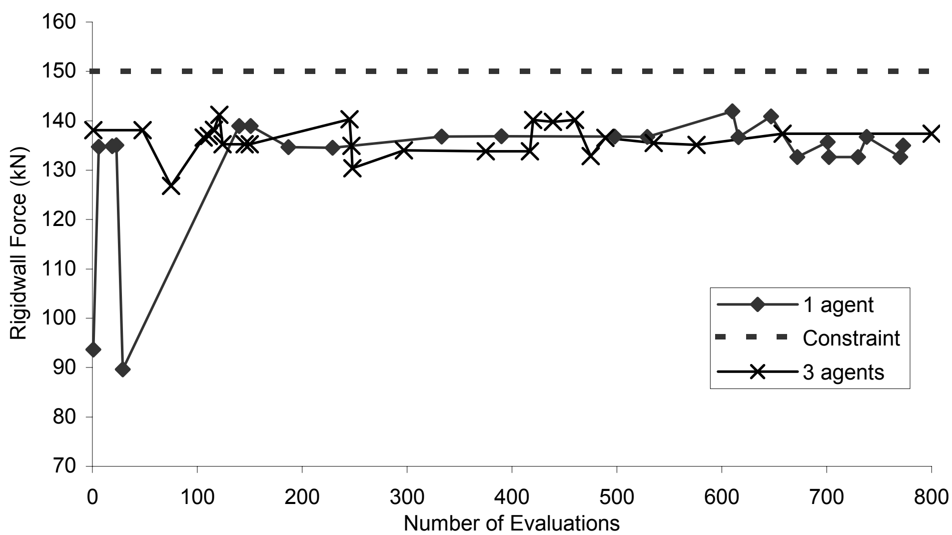

Figure 7. Trajectory of rigid wall force.

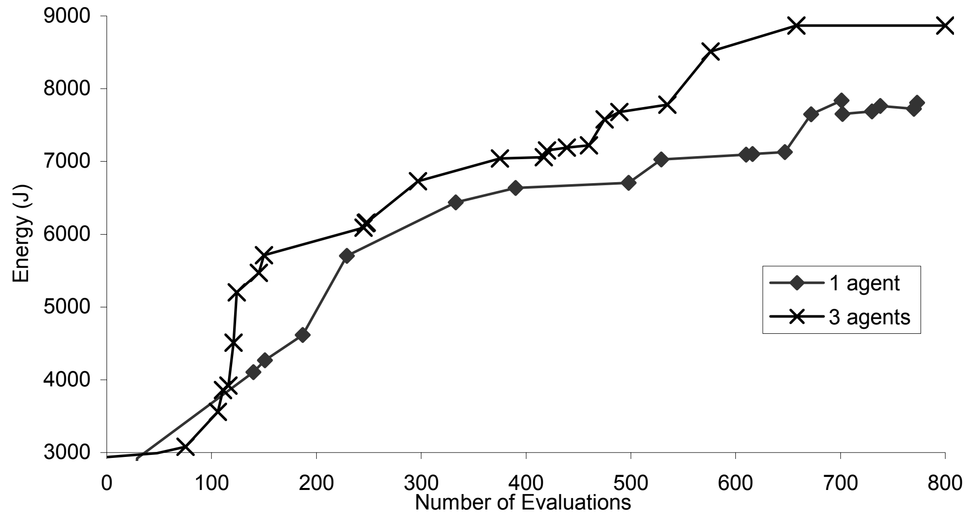

Figure 8. Trajectory of internal energy.

4. Shape Optimization of a Torque Arm

Shape optimization of a torque arm was performed. The problem statement is similar to one originally published by Botkin, Wang, Kim, and Choi (2002).

4.1 Problem definition

The original, or baseline, geometry of the torque arm is shown in Figure 9. Two concentrated loads are applied to the uppermost point of the right-end hole: 5066N in the upward direction and 2789N to the left. The left-most hole is restrained. The torque arm is made of steel and is 3 mm thick.

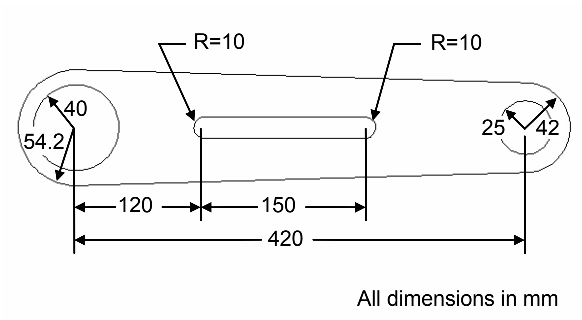

Figure 9. Geometry of torque arm.

4.2 Problem setup

The objective of the optimization problem was to minimize mass while simultaneously constraining the Von Mises stress to remain under 800 MPa. To reach these goals, the four design parameters shown in Figure 10 were allowed to vary as follows: x1 between 50 mm and 110 mm, x2 between 170 mm and 280 mm, r1 between 10 mm and 45 mm, and r2 between 10 mm and 40 mm. A single agent was used within HEEDS to solve the optimization problem.

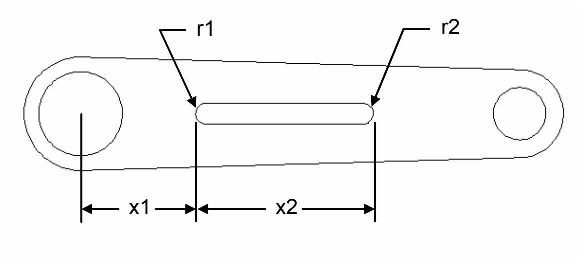

Figure 10. Design Parameters of the Torque Arm.

The potential for large element distortions in this problem dictated that a new mesh be created for each evaluation. In addition, the geometry of the torque arm had to be generated parametrically. Both of these capabilities exist within Abaqus/CAE and were used in the HEEDS optimization environment through the use of a python script file. This file instructed Abaqus/CAE to create a geometry based on the current HEEDS-generated values of the design variables, mesh the part, and write out the input file required to run Abaqus/Standard. Abaqus/Standard was then used to solve the finite element problem. This procedure was repeated for each evaluation. Linear elastic analysis was performed on each potential design.

4.3 Results and Discussion

The geometry and behavior of the best design obtained by HEEDS are shown in Figure 11 and Figure 12, respectively. Figure 13 illustrates the convergence of the mass objective, while Figure 14 shows the stress convergence. Through the integration of HEEDS with Abaqus/CAE and Abaqus/Standard, a solution 20% lighter than the original baseline design was achieved. These results were obtained using a combination of evolutionary and local gradient-based search, an automatic search strategy within HEEDS. It should be noted that attempts to solve this problem using purely local search resulted in sub-optimal solutions (local optima) being found (Botkin, et al., 2002).

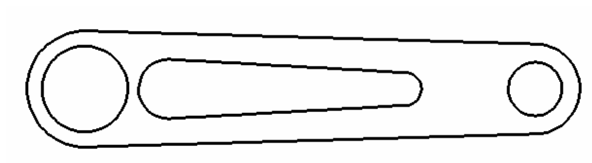

Figure 11. Geometry of final design.

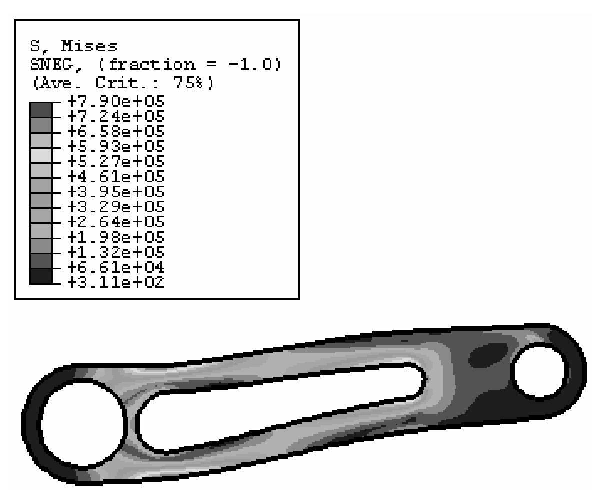

Figure 12. Stress (10^3 MPa) and deflected shape of final design.

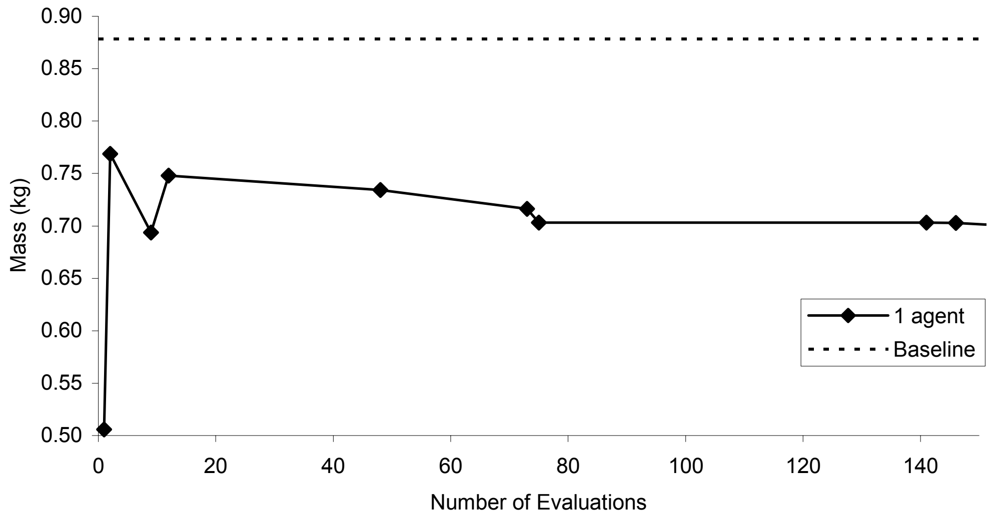

Figure 13. Trajectory of mass.

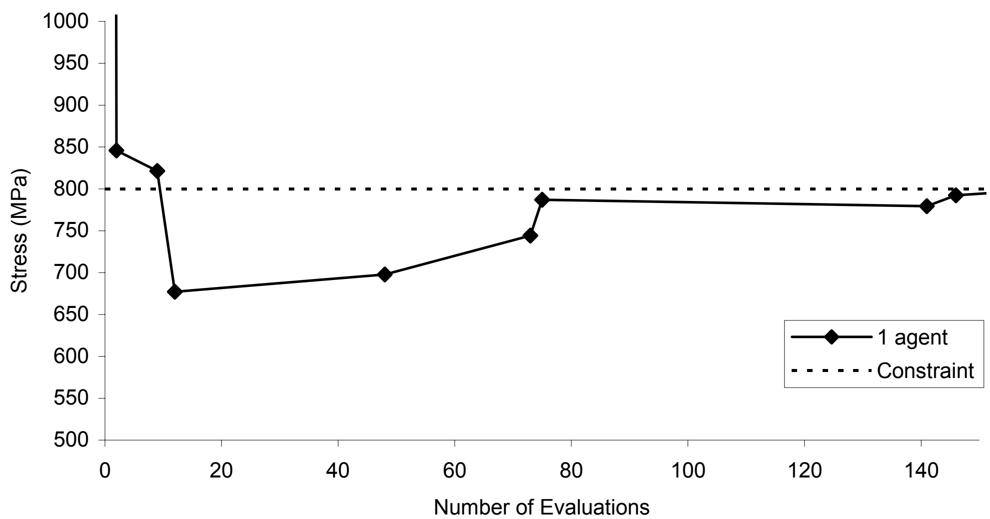

Figure 14. Trajectory of Von Mises stress.

5. Conclusions

The automated design optimization tool HEEDS was used to perform shape optimization of an automotive lower compartment rail and a torque arm bracket. A HEEDS mesh generator was used to model the rail, while Abaqus/CAE was used within the HEEDS environment to create new models for the torque arm. Abaqus solvers were used to evaluate the performance of each potential design. The rail example demonstrated the application and advantages of using hierarchical decomposition of the design space, wherein multiple agents work autonomously on subsets of the overall problem, sharing designs in a structured manner. This results not only in significant speedup of the overall search process, but also in better solutions because of the more thorough search throughout the design space. The torque arm example demonstrated the importance of combining global and local search methods to obtain a truly optimized design.

This process has been applied to many other problems, including several automotive lower compartment rail designs, resulting in significant gains in performance and reductions in mass compared to baseline rails designed by experienced engineers. These achievements were realized primarily through cross-sectional shape changes. In contrast, optimization of only material properties and section thickness in a rail with fixed cross-sectional shape will typically yield much lower performance improvements, even if mass is not reduced.

6. References

1. Botkin, M. E., H. Wang, N. H. Kim, K. K. Choi, "Shape Optimization of Two-Dimensional Automotive Components Using a Meshfree Method," 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, 2002.
2. Cochran, W. and G. Cox, "Experimental Designs," Wiley Classics Library Edition Publishing, 1992.
3. Holland, J., "Adaptation in Natural and Artificial Systems," Ann Arbor: University of Michigan Press, 1975.
4. Khuri, A. and A. Cornell, "Response Surfaces Design and Analyses," Marcel Decker Publishing, 1996.
5. Koch, P. N., J. P. Evans, and D. Powell, "Interdigitation for Effective Design Space Exploration using iSIGHT," Journal of Structural and Multidisciplinary Optimization, (In Press).
6. Ruthenbar, R., "Simulated Annealing Algorithms: An Overview," IEEE Circuits and Devices Magazine, Vol. 5, No. 1, pp. 19-26, 1989.
7. Schittkowski, K., "NLPQL: A Fortran Subroutine Solving Constrained Nonlinear Programming Problems," Annals of Operations Research, Vol. 5, pp. 485-500, 1985.