

APPLICATION OF A PRODUCT PLATFORM DESIGN PROCESS TO AUTOMOTIVE POWERTRAINS

Ryan Fellini* and Panos Papalambros†
{rfellini, pyp}@engin.umich.edu
University of Michigan
Ann Arbor, MI 48109, United States

Trudy Weber‡
trudy.weber@de.opel.com
Adam Opel AG
D-65423 Ruesselsheim, Germany

Abstract

Product platforms or product families are collections of artifacts related by shared components. Use of product platforms allows rapid deployment of product variants in the marketplace, resulting in reduced design cycle time and cost, and better-timed product launches. A method has been proposed that uses a multiobjective Pareto problem formulation to capture performance trade-offs realized when sharing components. The present study explores the use of the method for designing an automotive powertrain family. Initial results are presented along with a discussion of techniques that may lead to better solutions, such as the use of hierarchical decomposition and derivative-free global optimization methods to aid in accurate Pareto set generation. In addition, response surface models can be used to decrease numerical cost.

1. Introduction

In product platform design a family of artifacts that share common components is designed together as opposed to designing each artifact individually. A multiobjective optimization formulation was previously created to quantify the trade-offs involved in designing entire product platforms.^[1] The approach is demonstrated on a family of automotive powertrains, specifically, a product family consisting of a mid-size parallel configuration hybrid electric vehicle (HEV), a small electric vehicle (EV), and a conventional vehicle with a

continuously variable transmission (CVT). The challenge becomes one of developing methods that overcome increasingly large and computationally demanding models. Derivative-free global optimization algorithms, decomposition methods, coordination strategies, and surrogate models are explored.

1.1 Product Platform Formulation

The formulation in this study was first introduced by Nelson et al. and uses a multiobjective problem formulation to quantify the relationships between individual performance and overall efficiency in a product family.^[1] The generated Pareto sets give the designer a quantitative tool for deciding the feasibility of sharing components and for choosing designs. The general method is the following. Select a group of products that will share one or more components. Each of these products should have an optimization problem formulation associated with it. For example, two products A and B are designed individually with the models:

Product A	Product B
$\min f_A$	$\min f_B$
subject to	subject to
$g_A \leq 0$	$g_B \leq 0$
$h_A = 0$	$h_B = 0$

where f is a scalar objective and h, g are vector equality and inequality constraints, respectively. These two distinct problems can lead to a multiobjective optimization formulation that includes an equality (or commonality) constraint stating which components should be shared:

*Graduate Student, Department of Mechanical Engineering and Applied Mechanics.

†Professor, Department of Mechanical Engineering and Applied Mechanics, AIAA member.

‡Staff Research Scientist, International Technical Development Center

$$\begin{aligned}
 & \min w_A \cdot f_A + w_B \cdot f_B \\
 & \text{subject to} \\
 & \quad g_A \leq 0 \\
 & \quad g_B \leq 0 \\
 & \quad h_A = 0 \\
 & \quad h_B = 0 \\
 & \quad x_1^A = x_1^B \\
 & \quad \{ \text{commonality constraint} \}
 \end{aligned}$$

where w_A, w_B are the usual weights in the scalar substitute function, and x_1 is a subvector of the design variables that correspond to the shared components. Other scalar substitute functions can be used, but this is not the emphasis here. The next step is the computation of the null platform and utopia point, as in Figure 1.

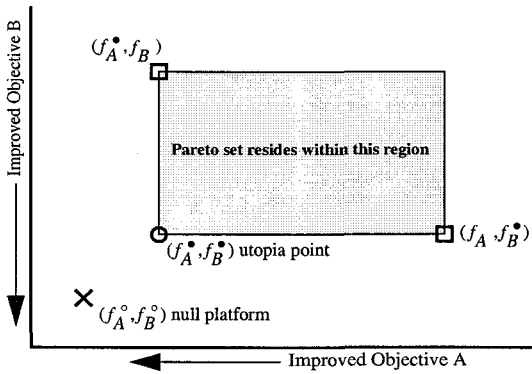


Figure 1: Utopia and Null Platform Points

The null platform point corresponds to the optimal design of each product if there is no component sharing. Likewise, the utopia point is found from optimizing the multiobjective problem formulation, and finding the bounds on the Pareto set by setting the weights to $\{0,1\}$ and $\{1,0\}$. The utopia point is a global bound on what can be achieved when sharing components. By observing the location of the null platform and utopia point, an immediate decision can be made if this amount of performance loss is acceptable. If it is, one can then generate the Pareto set to quantify the trade-off more completely, Figure 2.

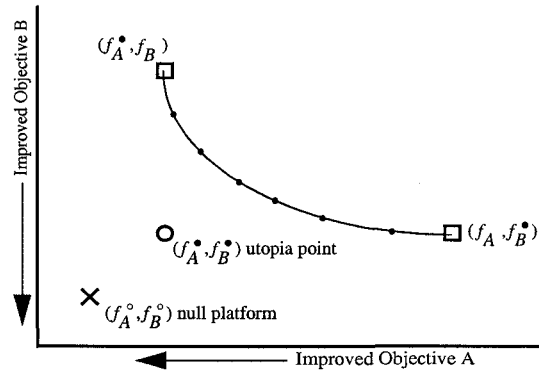


Figure 2: Pareto Set Generation

With this curve and similar ones corresponding to other parts-sharing schemes the designer has a quantitative tool to evaluate alternative design scenarios.

1.2 Derivative-Free Optimization

An important requirement when solving the product platform formulation is the generation of global solutions. Without such solutions the Pareto trade-off curve becomes less effective.

Gradient-based optimization algorithms such as Sequential Quadratic Programming (SQP) find local minima and present additional difficulties with regard to simulation-based models that contain noisy functions. There are several derivative-free optimization algorithms now available that allow for some form of global optimization. Common methods include simulated annealing (SA) and the genetic algorithm (GA). The amount of fine-tuning per optimization that is required by these stochastic algorithms does not make them desirable for this application. Methods based on the Nelder-Mead technique have supplied good results for some types of problems but are prone to cycling and further development is needed.^[1,2]

Two methods better suited to the problem at hand are Divided Rectangles (DIRECT) and Efficient Global Optimization (EGO). DIRECT works by dividing the design space and using Lipschitz theory to pick rectangles that further subdivide the space.^[4,6] EGO optimizes a function by fitting a Kriging model and using an expected improvement function to determine where to sample additional points that improve the approximation and find a better design.^[5] Here we will employ DIRECT because its constrained version is currently at a further stage of development. However, the method is effective only for problems of small size.

2. Two-Member Product Formulation

In the first part of this study a two-member family of automotive powertrains is explored. A future product mix of an automotive manufacturer includes a hybrid electric vehicle (HEV) and a conventional vehicle with a continuously variable transmission (CVT). To explore platform feasibility an attempt is made to share (1) the final drive, and (2) the IC engine. This component sharing scheme can be seen in Figure 3.

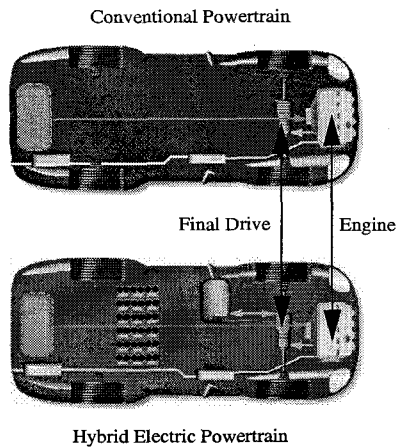


Figure 3: Two-Member Automotive Product Platform
[figures courtesy of NREL]

The Advanced Vehicle Simulator (ADVISOR) developed by the National Renewable Energy Laboratory (NREL), is employed for this study^[14]. ADVISOR is a MATLAB-based feed-backward simulation for modeling advanced powertrains, and has been used for earlier powertrain studies.^[3] The engine to be shared between the HEV and CVT powertrain is based on empirical data from a 67 kW Volkswagen 1.9L turbo-diesel (TDI) engine. The motor data used in the HEV powertrain are obtained from a Westinghouse 75 kW AC induction motor. The battery used is comprised of Ovonic NiMH power modules.

The problem formulation for the HEV uses fuel economy and performance goals put forth by the Partnership for a New Generation of Vehicle (PNGV) for a midsize passenger car:

maximize...

$$f_{HEV}(x_{HEV}) = \text{mpg (combined fuel economy)}$$

with respect to...

$$x_{HEV} = \{\text{engine size, motor size, battery size, final drive ratio}\}$$

subject to...

$$0 - 60 \text{ mph time} \leq 12 \text{ s}$$

$$40 - 60 \text{ mph (passing time)} \leq 5.3 \text{ s}$$

$$0 - 85 \text{ mph time} \leq 23.4 \text{ s}$$

$$\text{max acceleration} \geq 0.5 \text{ g}$$

$$\text{max speed} \geq 85 \text{ mph}$$

$$5 \text{ s distance} \geq 140 \text{ ft}$$

$$55 \text{ mph cruise grade} \geq 6.5 \%$$

$$\text{max grade at launch} \geq 30 \%$$

$$\text{Delta SOC for FUDS} \leq 0.5 \%$$

$$\text{Delta SOC for FHDS} \leq 0.5 \%$$

The goal is to maximize the fuel economy subject to performance constraints. The final two constraints are the maximum change on the state of charge (SOC) of the battery from start to end of the fuel economy run. They are imposed in order to properly calculate the energy usage. The fuel economy calculation is based on a combination of the U.S. Federal Urban Driving Schedule (FUDS) and Federal Highway Driving Schedule (FHDS). The design variables are engine and motor sizes, number of battery modules, and final drive ratio.

For the CVT powertrain a small car (similar in size to the EV) is assumed. The model is the following.

minimize...

$$f_{CVT}(x_{CVT}) = 0 - 60 \text{ mph time}$$

with respect to...

$$x_{CVT} = \{\text{engine size, final drive ratio, CVT lo, CVT hi ratio}\}$$

subject to...

$$\text{mpg (combined fuel economy)} \geq 40 \text{ mpg}$$

$$40 - 60 \text{ mph (passing time)} \leq 5 \text{ s}$$

$$0 - 85 \text{ mph time} \leq 24 \text{ s}$$

$$\text{max acceleration} \geq 0.5 \text{ g}$$

$$\text{max speed} \geq 100 \text{ mph}$$

$$5 \text{ s distance} \geq 140 \text{ ft}$$

$$55 \text{ mph cruise grade} \geq 6.5 \%$$

$$\text{max grade at launch} \geq 30 \%$$

The premise is that this vehicle could be marketed as a small sports hatchback in the European market. For this reason the performance constraints are made somewhat more stringent, and the objective is to minimize the 0 to

60 mph acceleration time of the car. The design variables are size of the engine, final drive ratio, and upper and lower limits on the CVT.

For each component we must define upper and lower bounds. For the final drive ratio, we allow the range to vary from 3.0 to 5.0. For the lower and upper CVT ratios, the bounds are 0.1 to 1.0 and 1.5 to 4.0, respectively. For the engine size we set a range at 50 to 84 kW. Likewise, for the motor we set the bounds at 56 to 94 kW. Finally, the batteries are allowed to vary from 10 to 40 modules.

The bounds selected for the transmission ratios and battery modules can be fairly large and are based on engineering intuition. However, to change the engine or motor size the data must be linearly scaled up or down. For such empirical data one should limit scaling to within 25%, hence the numbers chosen above. Optimization may result in optima that are simply bounded. This is not ideal, but the choice of tighter bounds was made to stay within the numerical accuracy of the empirical models.

The multiobjective model for the product platform can be formulated as follows.

$$\begin{aligned}
 & \text{minimize...} \\
 & f = (-)w_1 \cdot f_{HEV}(x_{HEV}) + w_2 \cdot f_{CVT}(x_{CVT}) \\
 & \text{subject to...} \\
 & g_{1-10-HEV} \\
 & g_{1-8-CVT} \\
 & \text{commonality constraints...} \\
 & h_1 = (\text{engine size})_{HEV} = (\text{engine size})_{CVT} \\
 & h_2 = (\text{final drive})_{HEV} = (\text{final drive})_{CVT}
 \end{aligned}$$

The objectives have been combined with minus signs added where necessary to keep a consistent minimization problem. The formulation is subject to the 18 constraints collected from all the models and includes the two commonality constraints. These define component sharing and can be included or excluded depending on what component sharing one would like to study. We can reduce the size of the problem by removing all the commonality constraints and just giving one label to the shared components: engine size and final drive.

Employing the DIRECT algorithm with an all-at-once (AAO) approach we are able to obtain the results shown in Figure 4.

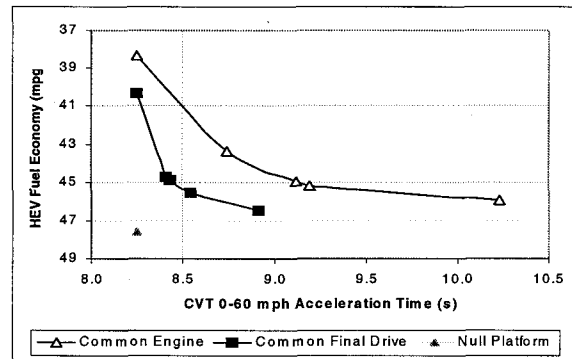


Figure 4: Trade-Off Between HEV & CVT Powertrains

The results quantify the trade-offs for a shared final drive and then a shared engine. It is established that one could share the final drive without too much sacrifice in overall performance. This is true if we select a design near the middle of the respective Pareto curve. Making the decision whether or not to share the engine would require more thought from the designer into what amount of performance could be lost from either configuration. The next level of decision would be a trade-off curve when sharing both components simultaneously.

These initial results seem useful to designers of complex systems working at the conceptual design phase of implementing a product family. Computationally, DIRECT was able to converge to a reasonable solution 75% of the time. Runs which did not arrive at a constraint bound solution were excluded in the graph of Figure 4.

3. Three-Member Product Formulation

In the second part of this study we attempt to expand the product family to include an electric vehicle (EV) meant for city use. This component sharing scheme can be seen in Figure 5. In this formulation an attempt is made to share (1) the final drive between all three vehicles, (2) the AC motor between the EV and HEV powertrains, and (3) the IC engine between the conventional and HEV powertrains.

The problem formulation for the EV vehicle is also based on the PNGV requirements. The car modeled is similar in size to the General Motors EV1. The optimization model follows below.

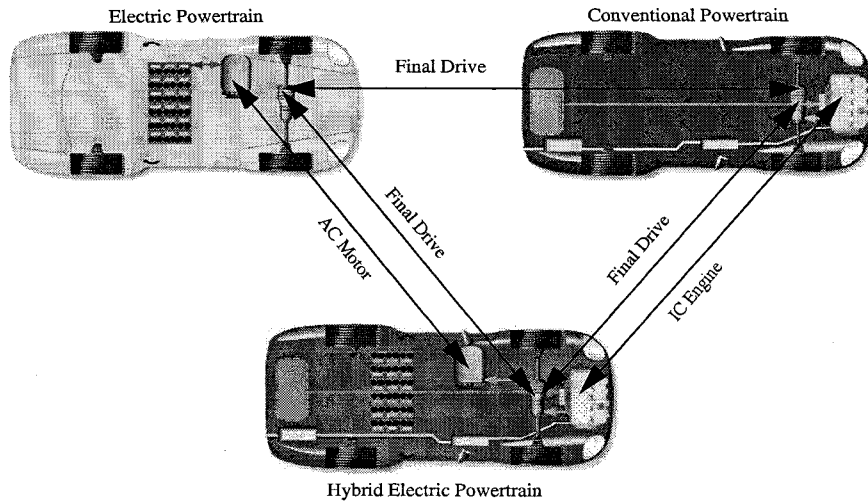


Figure 5: Three-Member Automotive Product Platform [figure courtesy of NREL]

maximize...

$$f_{EV}(x_{EV}) = \text{range (on HWFET cycle)}$$

with respect to...

$$x_{EV} = \{\text{motor size, battery size, final drive ratio}\}$$

subject to...

- $0 - 60\text{mph time} \leq 12\text{s}$
- $40-60\text{ mph (passing time)} \leq 5.3\text{s}$
- $0 - 85\text{mph time} \leq 23.4\text{s}$
- $\text{max acceleration} \geq 0.5\text{g}$
- $\text{max speed} \geq 85\text{mph}$
- $5\text{s distance} \geq 140\text{ft}$
- $55\text{mph cruise grade} \geq 6.5\%$
- $\text{max grade at launch} \geq 30\%$

We are seeking to maximize the highway range of the vehicle using the Highway Fuel Economy Test (HWFET) cycle. The design variables are motor size, number of battery modules, and final drive ratio.

The multiobjective model for the new product platform becomes the following.

minimize...

$$f = (-)w_1 \cdot f_{HEV}(x_{HEV}) + (-)w_2 \cdot f_{EV}(x_{EV}) + w_3 \cdot f_{CVT}(x_{CVT})$$

subject to...

$$g_{1-10-HEV}$$

$$g_{1-8-EV}$$

$$g_{1-8-CVT}$$

commonality constraints...

$$h_1 = (\text{engine size})_{HEV} = (\text{engine size})_{CVT}$$

$$h_2 = (\text{motor size})_{HEV} = (\text{motor size})_{EV}$$

$$h_3 = (\text{final drive})_{HEV} = (\text{final drive})_{EV} = (\text{final drive})_{CVT}$$

The new formulation is subject to the 26 constraints and includes the three commonality equalities.

The first attempt is to solve a segment of this formulation that just includes component sharing between the HEV and EV powertrains. Once again DIRECT is employed in an AAO approach. The results in Figure 6 were computed.

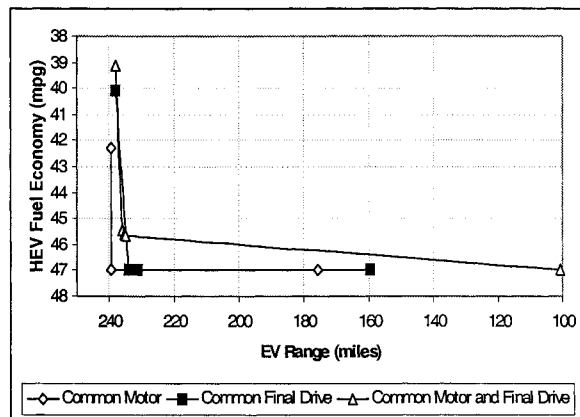


Figure 6: Trade-Off Between HEV & EV Powertrains

Some interesting observations can be made regarding these results. We obtain utopia points for all three cases (common motor, common final drive, and common motor and final drive) that are essentially the same as the null platform. Points along the Pareto set computed between the outer extremes move directly towards this utopia point (notice the concentration of points for all three plots). Thus it appears that either we have no competing objective in this formulation or the strict bounds we have imposed on the design variables are consistently active. A non-competitive trade-off seems likely because both the fuel economy of the HEV and

the range of the EV are efficiency objectives and then may not conflict. Recall that in the case of the HEV and CVT in section 2 we have a very distinct conflict between efficiency and performance (acceleration time of the CVT). The results from the second study suggest to the designer that sharing components between the HEV and EV powertrains would cause no performance loss between the products. This can not be readily concluded by simply reviewing the problem formulations.

It becomes interesting at this stage to solve the complete product platform problem and gain an understanding of how the relationships found in Figure 6 might effect the entire three-product family. DIRECT is once again implemented in an AAO approach for the full model, but now it is not possible to solve the problem along the entire Pareto surface successfully. For certain combinations of weights, DIRECT is able to find good solutions that are constraint bound. For other combinations DIRECT cannot reach a solution even when a large number of iterations is allowed. The size of the problem seems to have exceeded the capability of the method.

4. Product Platform Decomposition Strategy

This section describes a solution strategy for dealing with the product platform design problem that is currently under study.

4.1 Problem Decomposition

The simulation-based platform design model is sufficiently large to defy an AAO approach. An alternative is to break down the problem into more manageable portions. Decomposition methods can be employed to accomplish this. The techniques were thoroughly reviewed by Wagner.^[13] There are various types of decomposition such as aspect-based, object-based, and model-based. The first type of decomposition is prevalent in the field of MDO. In this study we apply a model-based technique for a hierarchical formulation as shown in Figure 7. As discussed further below, using a decomposition strategy requires a careful examination of the problem partitioning and coordination.

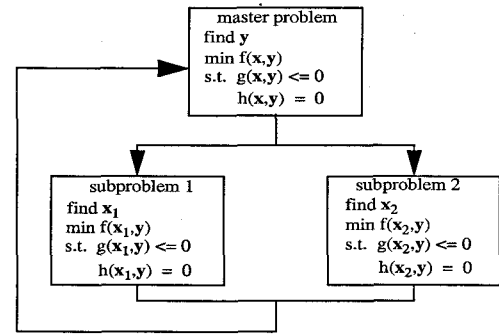


Figure 7: Hierarchical Problem Formulation

4.2 Functional Dependence Table (FDT)

The functional dependence table (FDT) is a boolean matrix that represents the dependence of the design variables on the functions of an optimization model. The FDT can be used to identify structure in the problem formulation. This structure can allow for a model-based decomposition. Previous work has been reported on applying these techniques to powertrain models based on analytical equations.^[10,13]

To create the product platform FDT we can reduce the size of the problem (as in the examples) by removing all the commonality constraints and just giving one label to the shared components: engine size, motor size, and final drive. This reduces the eleven variables to seven. Next, for each product we aggregate the constraints into a compact form in order to create the FDT. The derivation of the FDT for the case study can be seen in Figure 8.

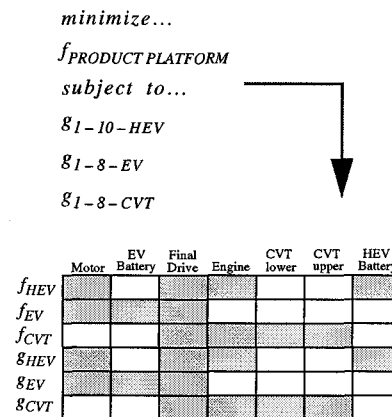


Figure 8: FDT for Product Platform

The final step is to determine a proper partition from the FDT's structure. With smaller scale problems it is often possible to determine this structure by inspection. However, more rigorous techniques are available with the use of partitioning algorithms, as discussed next.

4.3 Partitioning of the FDT

The partitioning procedures developed in the literature use a graph-based representation of functional dependence. For a specified number of subproblems, partitions reduce dependence between subproblems and balance the sizes of the subproblems.^[7,8,9,10] Figure 9 illustrates the idea.

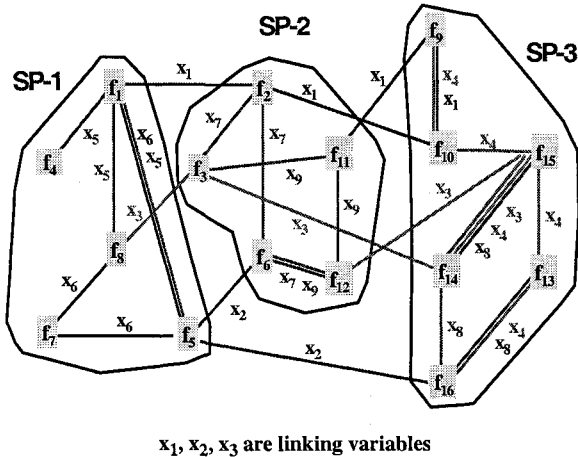


Figure 9: Partitioned Hypergraph

For this problem the integer programming-based method is used^[7,8]. Taking the original FDT, and applying the partitioning algorithm, the decomposition in Figure 10 is computed.

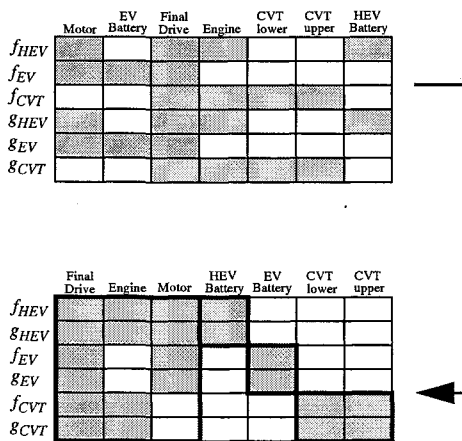


Figure 10: Partitioned Product Platform FDT

The three bordered rectangles to the right represent independent subproblems coupled by the linking variables. The linking variables and master problem are located in the large rectangle to the left. None of the individual problems have more than three variables, and the subproblems have only one or two. Figure 11 shows a visual representation of the hierarchical problem formulation derived from the partitioned FDT.

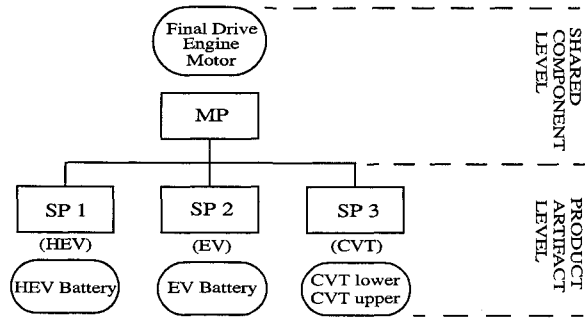


Figure 11: Decomposed Product Platform

The schematic shows all the separate problems and the variables that are optimized in each problem. Note that in the master problem we are posing the entire multiobjective problem formulation, where at the subproblem level we are posing the individual product variant formulation. In general this will be the structure of the product platform problem, and the makeup of the formulation itself makes intuitive sense.

Because the top level problem is the product platform formulation with just the shared variables, we call it the *shared component level*. Likewise, because the subproblems become representations of each of the individual artifacts, with respect to their local variables, we call this the *product artifact level*.

4.4 Global Coordination

An effective coordination strategy must be identified next. The difficulty with existing coordination strategies is that they have been developed for use with gradient-based local optimizers and are not intended for global optimization. A possible approach that retains the hierarchical formulation and allows for global optimization when using global optimizers at all problem levels, is illustrated in Figure 12.

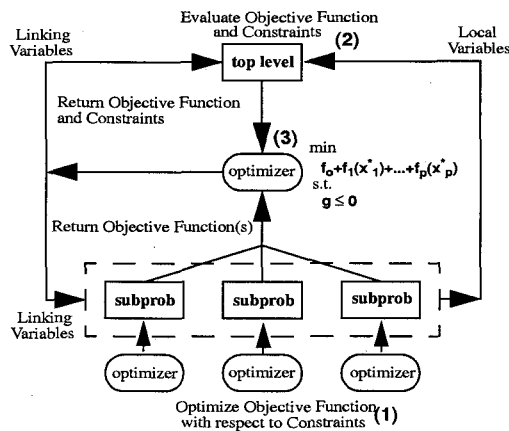


Figure 12: Non-Convex Coordination

This strategy assigns individual optimizers to each of the subproblems. Each subproblem is solved with respect to the local variables with the current values of the linking variables fixed. The next step is to pass the optimized local variables back to the top level problem, along with the linking variables, to evaluate (not optimize) the objective function and constraints (which would usually have been optimized in the master problem). The constraint and objective functions from the top level problem, along with the objective functions from the subproblems are then passed to the central optimizer. The central optimizer takes this information and uses it to pick a new set of linking variables to pass along to the top and subproblem levels. The strategy in effect is partially minimizing the objective function for every iteration of the linking variables.

The coordination strategy was implemented in MATLAB and initial tests were conducted on solving analytical global optimization problems. Results are encouraging, however, the overall expense of this coordination strategy tends to be quite high. This expense is also present in optimization runs performed in conjunction with the product platform problem. While the results obtained are better than those from the AAO approach, more development is necessary to increase the efficiency and robustness of this type of technique.

5. Conclusion

The product platform design process was applied to the conceptual design of a family of automotive powertrains. The study provided interesting results with regard to quantifying the trade-offs encountered when attempting to share components among the powertrains. Besides showing the potential of the approach, the study

exposed the computational difficulties involved, especially when using simulation-based models.

Model-based decomposition along with a hierarchical coordination strategy is suggested as a promising way to deal with these computational difficulties. Smaller-sized problems would allow algorithms like DIRECT (or EGO) to work. But coordination strategies and derivative-free methods will generally lead to increased computational expense. Since the structure for performing distributed computation is readily available within a decomposed framework, this is one possible remedy of dealing with this expense. High-fidelity surrogate models of the simulations could further decrease the computational cost greatly. Full-factorial designs along with smooth splines could effectively and efficiently approximate our three and four variable simulations, reducing them to very quick function calls. Since generation of Pareto points require extensive model reuse, surrogate models appear as a viable alternative. For a study of Kriging approximations in the context of optimization, refer to work by Sasena et al (included in these proceedings).^[12]

The global coordination idea is still in an ad hoc rudimentary form. Developing a rigorous global coordination strategy is an important challenge.

The powertrain family problem itself is a promising area for further work. If the mathematical challenges outlined above can be met, the insights that can be provided to vehicle design and managers in problems with increased complexity are very attractive.

Acknowledgements

This research was partially supported by the General Motors Satellite Research Center (GM SRL) and the Automotive Research Center (ARC), both at the University of Michigan. This support is gratefully acknowledged. The authors wish to thank D. Jones of General Motors for his advice with regard to DIRECT; and M. Cuddy, V. Johnson, T. Markel, and K. Wipke of NREL for their assistance with ADVISOR.

References

- 1 Box, M.J., "A New Method of Constrained Optimization and a Comparison with other Methods," Imperial Chemical Industries Limited, Central Instrument Research Laboratory, Bozdown House, WhitChurch Hill, Nr. Reading, Berks, 1965.
- 2 Dennis, J. and Torczon, V., "Direct Search Methods on Parallel Machines," *SIAM Journal on Optimization*, Vol. 1, pp. 448-474, 1991.
- 3 Fellini, R., Michelena, N., Papalambros, P., Sasena, M., "Optimal Design of Automotive Hybrid Power-

- train Systems,” *Proceedings of the EcoDesign '99: 1st International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, Tokyo, Japan, February, 1999.
- 4 Jones, D.R., “DIRECT,” *Encyclopedia of Optimization*, Kluwer Academic Publishers, 1999.
 - 5 Jones, D.R., Schonlau, M., Welch, W.J., “Efficient Global Optimization of Expensive Black-Box Functions,” *Journal of Global Optimization*, Vol. 13, No. 4, pp. 455-492, 1998.
 - 6 Jones, D.R., Perttunen, C.D., and Stuckman, B.E., “Lipschitzian Optimization Without the Lipschitz Constant,” *Journal of Optimization Theory and Applications*, Vol. 79, No.1, October, 1993.
 - 7 Krishnamachari, R. and Papalambros, P., “Hierarchical Decomposition Synthesis in Optimal Systems Design,” *Trans. ASME, Journal of Mechanical Design*, Vol. 199, No. 4, pp. 448-457, December, 1997.
 - 8 Krishnamachari, R. and Papalambros, P., “Optimal Hierarchical Decomposition Synthesis Using Integer Programming,” *Trans. ASME, Journal of Mechanical Design*, Vol. 199, No. 4, pp. 440-447, December, 1997.
 - 9 Michelena, N. and Papalambros, P., “A Hypergraph Framework for Optimal Model-Based Decomposition of Design Problems,” *Computational Optimization and Applications*, Vol. 8, No. 2, pp. 173-196, September, 1997.
 - 10 Michelena, N. and Papalambros, “Optimal Model-Based Decomposition of Powertrain System Design,” *Trans. ASME, Journal of Mechanical Design*, Vol. 117, No. 4, pp. 499-505, December, 1995.
 - 11 Nelson, S.A., Parkinson, M.B., and Papalambros, P.Y., “Multicriteria Optimization in Product Platform Design,” *Proceedings of 1999 ASME Design Engineering Technical Conferences*, Las Vegas, Nevada, USA, September, 1999.
 - 12 Sasena, M.J., Papalambros P.Y., and Goovaerts, P., “Metamodeling Sampling Criteria in a Global Optimization Framework,” *Proceedings of the 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-2000-4921, Long Beach, California, September, 2000.
 - 13 Wagner, T.C., *A General Decomposition Methodology for Optimal System Design*, Ph.D. Dissertation, Department of Mechanical Engineering and Applied Mechanics, University of Michigan, 1993.
 - 14 Wipke, K. and Cuddy, M., “Using an Advanced Vehicle Simulator (ADVISOR) to Guide Hybrid Vehicle Propulsion System Development,” *NREL Report for NESEA Sustainable Transportation*, S/ EV96, September 16-18, New York, NY, 1996.