Automating and Accelerating the Additive Manufacturing Design Process with Multi-Objective Constrained Evolutionary Optimization and HPC/Cloud Computing

Mark A. Buckner[1] and Lonnie J. Love[2]
[1]RF, Communications & Intelligent Systems Group
[2]Automation, Robotics, and Manufacturing Group
Oak Ridge National Laboratory
Oak Ridge, TN, USA
bucknerma@ornl.gov; lovelj@ornl.gov

***Abstract*—The ultimate objective of additive manufacturing is the implementation of techniques that can be used throughout the full manufacturing cycle. However, since its introduction, the additive manufacturing process has been used for little more than pre-production prototyping. The goal of some current work at ORNL is to change that reality. ORNL recently completed the first step towards optimizing the final design and manufacture of a component part (a cantilever in this case) using computer-aided design (CAD) tools, finite element analysis and simulations, and internally-developed optimization software. This paper will describe the present design process, the tools used, and the progress made thus far. It will also discuss the recent porting of ORNL's Multi-Objective Constrained Evolutionary Optimization (MOCEO) algorithms to ORNL's high performance computing (HPC) resources and to other resources available for Cloud computing, and the path forward for implementing additive manufacturing designs on these resources.**

***Index Terms*—additive manufacturing; optimization; 3D printing; computer-aided design; high performance computing; cloud computing.**

I. Introduction

Additive manufacturing, also known as 3D printing, is a process whereby solid objects are made using additive techniques to lay down successive layers of a material until the object is completely formed. It differs from the traditional machining (subtractive) techniques in that material is sequentially added, instead of being removed by drilling, cutting, shaving, etc. The anticipated advantages of additive manufacturing over subtractive manufacturing are a reduction in the time it takes to form an object (no machining) and the cost of excessive materials. The ultimate objective of additive manufacturing is the implementation of techniques that can be used throughout the full manufacturing cycle, from pre-production (rapid prototyping) to full scale production (rapid manufacturing). However, since its introduction, the additive manufacturing process has been used for little more than pre-production prototyping. The goal of some current work at ORNL is to change that reality.

This paper will describe the present design process being developed, and the progress obtained thus far. In addition, the paper will discuss the recent porting of the Multi-Objective Constrained Evolutionary Optimization (MOCEO) algorithms developed at ORNL to ORNL's high performance computing (HPC) resources and to other resources available for Cloud computing, and a path forward for implementing additive manufacturing designs on these resources.

II. Additive Manufacturing at ORNL

Additive manufacturing is an innovative manufacturing technology for the fabrication of complex parts not possible by conventional manufacturing methods. It is achieved by the successive melting of layers of powder material, rather than removing material by conventional machining. This, in turn, lends additive manufacturing to the rapid energy efficient manufacturing of products with greatly reduced material waste. ORNL is presently engaged with a DOE initiative in the EERE Advanced Manufacturing Office (AMO) designed to accelerate the manufacturing capability of a multitude of additive manufacturing technologies utilizing various materials from metals to polymers to composites. ORNL is also engaged with the DARPA Digital Manufacturing Analysis, Correlation, and Estimation (DMACE) Challenge in efforts to reduce the cost and weight of robotic components made by low-cost titanium powders, allowing a paradigm shift leading to low-cost disposable robots. ORNL has already digitally manufactured over 200 complex shaped meshed spheres for the DARPA DMACE Challenge, demonstrating active predictive modeling.

III. MOCEO Overview

Multi-Objective Constrained Evolutionary Optimization (MOCEO) based on extensions of the method of detecting the presence of local minima in multivariate error surfaces and the multivariate optimization code developed by Buckner [1]. MOCEO combines a variant of differential evolution (DE) and a hybrid conjugate-gradient/line search (CGLS) technique. DE is a population-based, direct-search algorithm for global optimization. While originally designed to operate on

978-1-4673-2482-3/12/$31.00 ©2012 IEEE

continuous floating point variables, DE has been extended to optimize a mixture of integer, discrete, and continuous variables and is able to employ multiple linear and non-linear constraints. DE differs from other population-based search algorithms, like Evolutionary Strategies (ES) and Genetic Algorithms (GA), in its self-referential mutation scheme and its selection process.

DE operates on a population $\mathbf{P}$ of candidate vectors $\mathbf{x}$. The size of the population, NP, is held constant for all generations. Each population member is denoted as $\mathbf{x}_{i,G}$ where i indexes the population and G the particular generation. A population of NP, D-dimensional members is written as:

$$\mathbf{P}_G = (\mathbf{x}_{1,G}, \ldots, \mathbf{x}_{i,G}, \ldots, \mathbf{x}_{NP,G}),$$
$$\textit{where}$$
$$\mathbf{x}_{i,G} = (x_{1,1,G}, \ldots, x_{j,1,G}, \ldots x_{D,1,G}),\ i = 1,2,\ldots,NP,\ \ G = 1,2,\ldots,G_{\max}$$

There are numerous strategies for generating the initial population, including the use of domain knowledge about the problem. This provides the opportunity of seeding the population with candidate parameters obtained from any number of sources including alternative algorithms and expert knowledge. For most real-world engineering problems the input parameters will be constrained by lower and upper boundary conditions xj,i(L) and xj,i(U) where j=1,…,D of member i. In cases where hard boundary constraints can be ignored DE can search the parameter space beyond the initialization bounds. Several methods have been implemented to enforce the boundary constraints when necessary. Hard boundaries simply truncate or round the offending parameter, while soft boundaries use some type of an asymptotic rule.
Boundaries will be applied in two different ways in this work. First, boundaries will be used to initialize a population within the physical size constraints the candidate part is permitted to occupy. Second, hard boundaries will be used to ensure that only valid parameters are possible. The use of boundaries has the added benefit of preventing candidate components with bad parameters from being evaluated, thus conserving computation time and resources.

IV. Recent Work

ORNL recently completed the first step in demonstrating the feasibility of optimizing the final design and manufacture of a component part (a cantilever in this case) using computer-aided design (CAD) tools, finite element modeling and simulations, and internally-developed optimization software. A parameterized prototype cantilever was created in SolidWorks. Software was developed to allow MOCEO to interface with and drive SolidWorks.

This interface application reads a newPop.txt parameter file, modifies the specified component based on the parameters, simulates the effect of a force being applied to the end of the cantilever, extracts the safety factor (SF) and writes results to an oldPop.txt output file. The SF describes the structural capacity of the cantilever i.e., how much force it can endure before failing. This allows MOCEO to explore many generations of candidate designs performing optimization through DE driven selective breeding.

In this initial case the objective used for optimization was the SF. Future implementations will address inclusion of additional objectives specifically related to the component being designed. These might include minimizing cost based on the type and amount of material, maximizing strength and optimizing fluid flow for a hydraulically articulated prosthetic.

Depending on the number and weighting of the objectives there may be multiple candidates that meet the design goals. Several can be selected for fabrication and testing based on the preferences of the designer. Data obtained from the tests will be used to improve the parameterization, constraints and simulations.

Fig. 1 shows the initial design for the cantilever. The base of the cantilever (the wide end) was allowed to vary in width and was optimized for a safety factor of 2 as the measure of performance. The resulting optimized cantilever design is shown in Fig. 2.

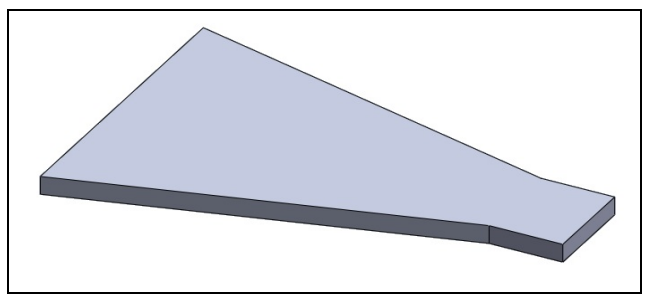

Figure 1. Initial design

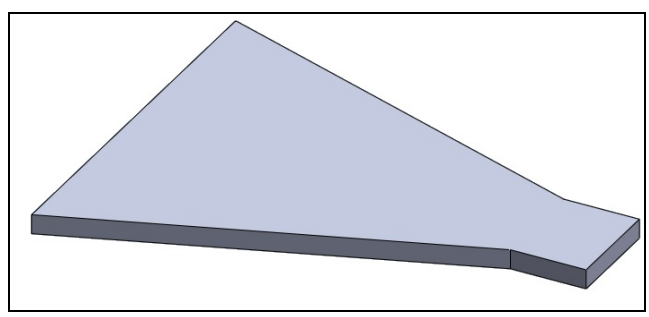

Figure 2. Optimized design with a widened base

V. Porting of MOCEO to HPC

The core of MOCEO was recently restructured, ported, parallelized, and tested on two High Performance Computing machines at ORNL, the SGI Altix ICE 8200 128 node cluster, and Jaguar, a 2.0 Petaflop Cray XT5 with ¼ million computing

cores. It was also successfully tested on Amazon's Elastic Compute Cloud (EC2).

The initial restructuring took the original MATLAB implementation and created a modular design, which allows easy reconfiguration and extensibility of the engine without modification to the engine code itself. This restructured code was re-written in C++, implementing a parallel code structure using the Message Passing Interface (MPI). The Armadillo C++ Linear Algebra Library, which provides an efficient optimized wrapper for the Basic Linear Algebra Subprograms (BLAS) and Linear Algebra PACKage (LAPACK) matrix libraries, was used to implement most of the matrix routines in the C++ code. Parallelism was achieved by dividing the test set among the available compute nodes. For every set of candidate model parameters received from the master node, each compute node calculates a set of error values and gradients for its assigned portion of the test. The master node gathers these results from the compute nodes, computes an aggregate error value for each model, and determines the next set of candidate model parameters.

Initial tests using a 400-point, 60-dimensional dataset demonstrated a 90x speedup on 256 compute cores of SGI Altix ICE 8200 machine compared to the multi-threaded MATLAB implementation on a dual quad-core workstation.

VI. Path Forward

Our vision is to radically change the way we design for additive manufacturing by combining the power of MOCEO, and the massive potential of cloud-based modeling and simulation. There are multiple paths one could take to realize this vision. This section describes one of them. It also highlights some work that we believe exemplifies approaches that can be copied, transformed and combined to create a truly innovative approach to design [2].

The power of MOCEO's selective breading lies in the ability to rapidly explore multiple generations of entire populations of candidate designs as quickly as possible. And while MOCEO can run on HPCs and the cloud, the computational bottleneck is not the breading, evaluation, and criterion checking of parameter constraints it's the manipulation of the CAD models and functional simulations such as Finite Element Analysis (FEA). It is possible to evaluate an entire population of candidate designs simultaneously using public clouds, such as Amazon's EC2. We are currently exploring this option using Bright Computing's Cloud Bursting [3]. Our onsite cluster will run MOCEO and local versions of commercial CAD modeling and simulation tools. The Bright Cluster Manager will automatically burst to the cloud for each generation running the CAD models and simulations of the entire population of candidate designs in parallel.

One of the challenges with this approach is the licensing cost associated with each instance of commercial CAD and simulation tools. To address this challenge we are investigating several alternatives including lower cost and generative CAD tools and innovative simulation approaches [4]–[20]. While the quality of these options may not be as good as there commercial counterparts they may be good enough for use in an initial design exploration phase. The top candidates from this initial phase could then be run through the local commercial tools for final assessment.

One last topic we want to introduce is the idea of motifs. Motifs are repeated themes, processes or patterns that can be used to create new innovative designs. Motifs are analogous to the concepts of Design for Additive Manufacturing (DFAM) and Manufacturable ELement (MEL) [21], and Compositional Pattern Producing Networks (CPPNs) [4], [12].

This is an exciting area of research, one which could revolutionize the way we design for additive manufacturing in the future.

Acknowledgement

The authors would like to acknowledge the contribution of Bryce L. Gamble, and thank the Volkswagen Distinguished Scholars Program for supporting his work this summer. Bryce designed the parameterized cantilever as well as the software that enabled MOCEO to interface with and drive SolidWorks. He is studying Electrical and Computer Engineering at Tennessee Technological University in Cookeville, TN, USA.

References

[1] M.A. Buckner "Learning from Data with Localized Regression and Differential Evolution," Ph.D. dissertation, Dept. Nucl. Eng., University of Tennessee, Knoxville, TN, 2003.

[2] http://www.everythingisaremix.info/everything-is-a-remix-part-3/

[3] http://www.brightcomputing.com/Linux-Cluster-Cloud-Bursting.php

[4] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," Genetic Programming and Evolvable Machines, vol. 8, no. 2, pp. 131–162, May 2007.

[5] http://endlessforms.com/

[6] Y.M. Xie, Z.H. Zuo, X. Huang, J.W. Tang, B. Zhao and P. Felicetti, "Architecture and urban design through evolutionary structural optimisation algorithms," Keynote Lecture of International Symposium on Algorithmic Design for Architecture and Urban Design , Tokyo, Japan, March 14-16, 2011, 11pp.

[7] X. Huang and Y.M. Xie, *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*. John Wiley & Sons, Chichester, England, 2010.

[8] Sundararajan, V.G. "Topology optimization for additive manufacturing of customized meso-structures using homogenization and parametric smoothing function," M.S. thesis, Univ. of Texas, Austin, TX, 2010

[9] E. Douglis, *Autogenic Structures.*Taylor & Francis, 2008.

[10] N. Oxman, S. Keating, and E. Tsai, "Functionally Graded Rapid Prototyping," 5TH International Conference on Advanced Research in Virtual and Rapid Prototyping; 2011.

[11] X. Y. Kou, S. T. Tan, and H. Lipson, "A Data-Driven Process for Estimating Nonlinear Material Models," *Applied Mechanics and Materials*, vol. 50–51, pp. 599–604, Feb. 2011.

[12] J. Clune and H. Lipson, "Evolving 3d objects with a generative encoding inspired by developmental biology," *ACM SIGEVOlution*, vol. 5, no. 4, pp. 2–12, 2011.

[13] J. Parthasarathy, B. Starly, S.Raman, (2011, Aug.). "A design for the additive manufacture of functionally graded porous structures with tailored mechanical properties for biomedical

applications," Journal of Manufacturing Processes, 13(2), pp. 160-170.

[14] [1] J. Hiller and H. Lipson, "Evolving amorphous robots," *Artificial Life XII*, pp. 717–724, 2010.

[15] http://creativemachines.cornell.edu/

[16] http://en.wikipedia.org/wiki/Adaptive_mesh_refinement

[17] Palmer, T. Nelson, E. Optimizing Next-Generation Automotive Structures Using Altair OptiStruct. Plastics in Lightweight and Electric Vehicles, Livonia, Michigan, November 7-9, 2011. [Online]. Available: http://altairenlighten.com/wp-content/uploads/2011/12/Optimizing-Next-Generation-Automotive-Structures-Using-Altair-OptiStruct.pdf

[18] http://www.paramountind.com/conformal-lattice-structures.html

[19] [1] R. Hudson, "Frameworks for practical parametric design in architecture," *Proceedings of the 26th eCAADe Conference*, pp. 2–5, 2008.

[20] http://www.altairproductdesign.com/Page.aspx?id=9

[21] D. Rosen, "Computer-aided design for additive manufacturing of cellular structures," *Computer-Aided Design & Applications*, vol. 4, no. 5, pp. 585–594, 2007.