Comparison of model based and heuristic optimization algorithms applied to photoinjectors using libEnsemble

N. Neveu, J. Larson, S. Hudson, L. Spentzouris

Illinois Institute of Technology
Argonne National Laboratory
nneveu@anl.gov

Oct. 20, 2018

Argonne NATIONAL LABORATORY

ILLINOIS INSTITUTE OF TECHNOLOGY

Outline

Facility Introduction

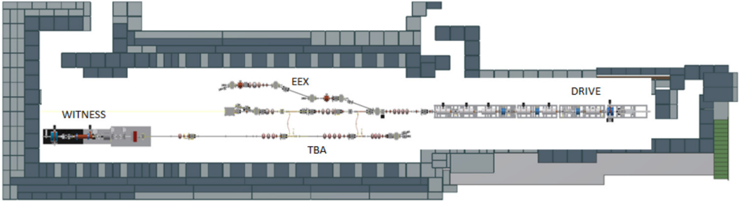
Simulation Setup

Optimization

Backup

Argonne Wakefield Accelerator Facility (AWA)

- Emittance Exchange (EEX)
- Two Beam Acceleration (TBA)
- Cathode Studies
- Plasma wakefield (very recent)



Code and Resources:

OPAL, Python, libEnsemble

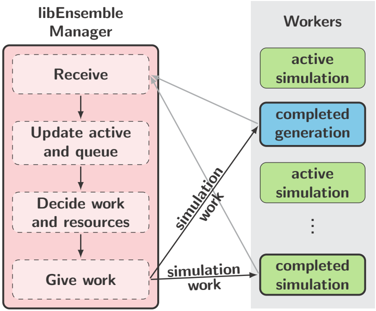
- Free
- Parallel
- Open Source
- 3D space charge (40 nC)

HPC Resources Used:

- Blues, ANL
- Bebop, ANL
- Theta, ANL

libEnsemble

- Automatically manages the asynchronous evaluation of calculations and, if desired, the optimization of outputs
- Manager/Worker paradigm
- Open Source Python code, on GitHub.
- Developers are happy to help users (myself as an example)
- Can run on laptops, clusters, and HPC systems (I have utilized over 200 KNL nodes at once using libEnsemble)

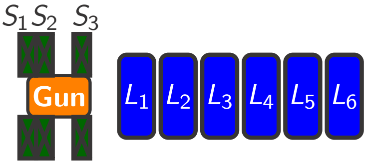


libEnsemble: Accelerator physics use case (w/ OPAL):

- Gracefully kills runs that loose particles, BEFORE end of simulation reached (which saves resources)
- Gracefully kills runs that hang (reach time limit)
- Saves specified data into a numpy array for access and storage
- Evaluates objectives based on beam criteria given (in this presentation z location)
- Each OPAL instance can be parallel
- Can also be used for random sampling of parameters

OPAL Model (Linac only)

- Photocathode, UV laser
- Charge 40 nC
  - 3D space charge
- Three solenoids (S_1-S_3)
- Six linac cavities (L_1-L_6)
- 2D RF field maps



Optimization

Model Based:

- Used BOBYQA as local optimization algorithm from Python NLopt package
- Used APOSMM to manage local / multi-start optimization runs
- APOSMM is developed at ANL, written in Python, and open source

Heuristic:

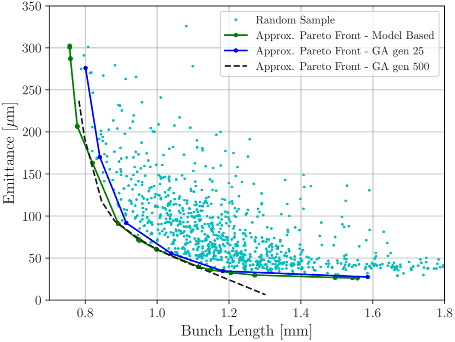
- Genetic Algorithm (GA), built into OPAL
- NSGA-II implementation
- Resource intensive (hours to days of compute time)
- Tested on several accelerator physics cases at various labs
- Trusted by the community...

Pre-libEnsemble: BOBYQA Results

- 10 design variables
- 2 objectives

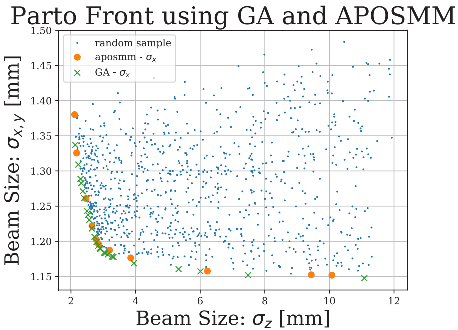
Results were encouraging but...

- Had to manually pick local opt starting points
- Cumbersome job handling
- Code was problem specific
- Not easily adapted to other optimization cases



libEnsemble: APOSMM + BOBYQA Results

- 9 design variables
- 3 objectives
- Number of local opt points chosen by APOSMM + libE
- Resources managed completely by libE
- Input file is easily transferable to new problems



Summary

- Model based methods were used to optimize a photoinjector
- LibEnsemble has the potential to simplify the optimization process and resource handling for many accelerator physics applications
- Future work will include a robust comparison of several optimization methods using libEnsemble

Thanks for your attention!

This project and AWA are funded by:

U.S. DEPARTMENT OF ENERGY

Backup: Code Used

- Used python library NLopt: http://ab-initio.mit.edu/wiki/index.php/Main_Page
- More info on libEnsemble: https://www.mcs.anl.gov/~jlarson/presentations/libEnsemble_A_Library_for_Managing_Ensembles_of_Calculations.pdf
- Questions about libEnsemble can be sent to: libensemble@lists.mcs.anl.gov
- Join the mailing list! https://lists.mcs.anl.gov/mailman/listinfo/libensemble

Pre-libEnseble: Linac Optimization Details

- 10 Design variables (table)
- Objectives were emittance and bunch length
- Used BOBYQA from NLopt python package

Variable	**Range**	**Unit**
Solenoid Strength	$0 \leq S_3 \leq 440$	amps
Phase of Gun	$-60 \leq \phi_g \leq 60$	degrees
Laser Radius	$0.1 \leq R \leq 30$	mm
Laser FWHM	$2 \leq T \leq 10$	ps
Cavity Phase	$-20 \leq \phi_L \leq 20$[1]	degrees

[1] $\phi_L = [\phi_{L_1}, \ldots, \phi_{L_6}]$

Pre-libEnsemble: Linac Optimization Details(scalarization)

- 1,000 point sample was done
- 132 simulations completed w/o error
- Scaled and shifted raw values to remove unit dependence

$$\bar{\epsilon}_x(v, z_1) = \frac{\epsilon_x(v, z_1) - \epsilon_{min}}{\epsilon_{max} - \epsilon_{min}}$$

- Used 11 weights from 0-1
- Solved 11 optimization problems $f(v, w)$ using BOBYQA

$$w \in \{0,\ 0.1,\ 0.2, \ldots, 1\}$$

$$f(v, w) = w\,\bar{\epsilon}_x(v, z_1) + (1 - w)\,\bar{\sigma}_z(v, z_1)$$