

## Lessons Learned from Parker's Internship

1. Blackbox optimization is probably not a good fit for generic HERON models.

The various blackbox methods tried tend to be quite sensitive to problem formulation and do not robustly find a minimum. The exact reason for this is not known for certain, but it is most likely due to insufficiently accurate gradient information. While these method can and do work for some cases, experience has shown them to be insufficiently robust to handle arbitrary user problems with differing initial guess values, transfer function formulations and cost function formulations.

2. Pyomo does support blackbox, but only through a surrogate trust constraint method.

The Pyomo blackbox support method (trust-constr) replaces blackbox elements of models with an AML surrogate, allowing Pyomo to use the AML interface to IPOPT or other optimizers. While this could be ideal for greybox modeling or very simple blackbox problems it does not seem to work well for larger blackbox problems like a HERON dispatcher would need.

3. There are multiple ways to get at IPOPT as an optimizer: AML and direct.

IPOPT provides two main access methods. The first of these is a direct method where the optimizer is directly passed an objective function, jacobian function and suitable constraint and constraint gradient functions. This is the method used previously through PyOptSparse and is the method used in cyIPOPT.

The other interface is an algebraic modeling language (AML) interface in the form of AMPL files. When this interface is used, IPOPT reads the AML files and internally generates corresponding objective, jacobian, constraint and constraint gradient methods. This is the interface used by Pyomo. Pyomo does not support access to IPOPT through the direct interface.

## Current State

There are multiple resources available for anyone interested in continuing work in this direction.

- Chickadee: This is an independent python package installable through PyPi as `chickadee-opt`. The source code is available on GitHub: <https://github.com/dhill2522/chickadee>
- Working Chickadee blackbox HERON fork: This method wraps the Python Chickadee package allowing it to be used as a HERON dispatcher. The

fork is also available on the chickadee branch of the GitHub repo: <https://github.com/dhill2522/HERON/tree/chickadee>

- Incomplete cyIPOPT HERON fork: This method seeks to use the cyIPOPT package as a HERON dispatcher. The current source code is available in another fork of HERON: <https://github.com/parkerwstott/HERON>
- The installation of pyOptSparse on a system can be quite involved (compiling IPOPT) as previously noted. A script has been developed for facilitating this installation for certain systems and is available upon request.

## Recommended Future Actions

- The trust constraint method could have interesting application in HERON for greybox modeling in HERON
- If there is still motivation for a blackbox dispatcher in HERON, I would recommend debugging the difference between the PyOptSparse and cyIPOPT implementations. While this has been tried, the difference between the working chickadee and broken cypopt implementations must lie in the exact functions and parameters being passed to IPOPT. There may be some additional magic that pyOptSparse is doing that results in one problem working while another fails entirely.