Modeling Work Plan

DRAFT

5 November 2019

# Essential Ingredients and Research Questions

1. Mildly nonlinear models.
   * Can technology models be linearized or quadraticized sufficiently to adequately represent their nonlinearities?
   * Is linearization or quadraticization sufficiently beneficial in solution speed and finding global optima?
2. Flexibility regarding mathematical programming vs Monte-Carlo simulation.
   * How can a computational framework be architected to support both mathematical programming and simulation approaches?
3. Experience-curve representation of expert opinion.
   * How does the quantification of expert opinion affect the tractability of the optimization problem?
4. Multiple objectives.
   * How nonlinear are the objective functions?
   * How much does having multiple objectives complicate mathematical programming approaches and the search for solutions?
5. Two-stage decisions.
   * Does this necessarily imply nesting two optimizations?
6. Identification of non-dominated and robust solutions.
   * Can generative methods be used to map these, or are generic nonlinear search methods needed?
7. Unification of probabilistic and robust methods.
   * Can robust methods be treated as a specialization of probabilistic ones, or vice versa?
8. Coupling with user interface and graphics.
   * How decoupled can the specifics of the modeling approach be from its control and presentation?

*Subtext:* If the models can be framed as mathematical programs, then they can be solved much faster than Monte-Carlo simulations and will find global solutions. However, layering complexities of multiple stages, multiple objectives, and nonlinearities on the problem may make mathematical programming not viable. If that is the case, then the framework could either solely employ straightforward Monte-Carlo simulation or nest mathematical programs within a broader framework.

# Timeline

1. November–January:
   * Attempt to prototype a mathematical-programming-based framework addressing the eight requirements above.
   * Fall back on nesting the mathematical programs withing a larger dataflow if having a unified mathematical program is not viable.
   * Fall back on a purely Monte-Carlo (non-mathematical=programming) approach if nesting mathematical programs is inadequate.
   * Clone and rework prototype models to represent multiple technologies.
2. February–March:
   * Tune prototype and populate with data and to represent the notional decision problem.
   * Link prototype model to user interface.
   * Finalize choice of computational platform.
3. April–September:
   * Iteratively refine and expand models as part of engagement with example decision-makers.

# Comparison of Platforms

|  |  |  |  |
| --- | --- | --- | --- |
| *Criterion* | \* Python\* | *Julia* | *Analytica* |
| Cost per developer | free | free | $5000 |
| Deployment cost | free | free | free |
| Ease of installation | moderate | difficult | easy |
| Web deployment (server) | possible | possible | unsure |
| Web deployment (cloud) | possible | possible | yes |
| Stability | stable | weak | stable |
| Programming skill needed | moderate | moderate | basic |
| Speed of software development | moderate | moderate | fast |
| Developer base | large | growing | small |
| Ease of user interface building | tedious | tedious | easy |
| Polish of user interface | varies | varies | minimal |
| Optimization libraries | best | good | good |
| Interoperability with Python | excellent | sufficient | possible |