Robust Modeling of Electrical Transmission Networks

**Blake Fritz**

[BFritz@math.umd.edu](mailto:BFritz@math.umd.edu)

**Advisor: Dr. Steven A Gabriel**

[sgabriel@umd.edu](mailto:sgabriel@umd.edu)

Dept. of Mechanical Engineering and

Applied Math, and Statistics & Scientific Computation Program

**Background**

Constructing and maintaining electrical power grids is a long-term, world-wide, multi-billion-dollar topic. Previous generations of infrastructure planners had to construct a reliable electrical network taxed by an ever-increasing population and industrial utilization. Into the 21st century, not only must upgrades in the power grid take these factors into account, but also the challenge of increasing electrical production coming from renewable sources. Green energy, such as wind and solar, are unreliable compared to traditional sources, both in the long term (since it is hard to forecast future development) and from day to day as the weather changes. Non-optimal investments in regional power grids, based on poor predictions of future trends, can incur significant costs as unneeded capacity is left unutilized or bottlenecks appear that threaten to choke off further development. Since a modern nation strives for 100% reliable electrical delivery, and construction of new infrastructure can take several years, forecasting decades into the future is a necessity.

Mathematically, the process of deciding where to build new power lines is a Transmission Expansion Planning (TEP) problem. This models a central planner, such as a nation, deciding where to construct new lines based on maximizing social welfare while minimizing costs, instead of issues of profitability that may motivate private companies. However, it is expected that individual producers are in a competitive environment and will emphasize their own profits. Social welfare in this scenario is defined as meeting all electrical demand of the population consistently and as economically as possible, and costs include both line construction costs, as well as daily generation outlays on producers. While methods of solving the deterministic version of TEP have existed for decades, introducing probabilistic constraints can greatly increase the complexity and create intractable models, even on modern computer hardware. One solution is to use scenario based stochastic programming, which has the drawback of very heavy computational complexity if many scenarios are included. Another method is robust optimization, which can be less complex to solve but runs the “risk” of over-emphasizing worst-case scenarios and thus being over conservative [1]. Furthermore, other suggestions have been brought forth including chance constrained, adaptive programming and so on, each with benefits and drawbacks. Since computational feasibility is a major concern for creating a software useful for real world application, robust methods are a good place to first explore if relatively accurate solutions can be found.

Robust optimization methods rely on solving for a worst-case, or nearly worst-case, scenario based on the plausible ranges of unknown parameters. Often, values are found by analyzing past performance or by running separate prediction models. In the case of a TEP, an Adaptive Robust Optimization (ARO) approach allows the scenarios to partially develop and then reassess them on a yearly basis with revised information. For example, power lines do not need to be built all on the first year, and planners can fulfill short-term needs while waiting on annual developments to better forecast long-term trends in supply and demand before committing to building new lines.

Since 2013, at least four different groups have advanced models for solving the stochastic TEP using ARO. Jabr [2] has a model using Benders Decomposition on the dual variables, but it only considers absolute worst-case realizations. Ruiz and Conejo [3] have proposed a model that combines the bottom sublevel problems using KKT conditions and solves the top-level problem with a constraint-and-column generation method that only uses primal variables. Some research suggests avoiding the dual variables will be computationally faster. Chen *et al.* [4] use a mathematically similar model to Ruiz. Finally, Minguez and Garcia-Bertrand [5] have published a method that has been shown to solve several data sets significantly faster by using elements from previous research.

This project aims to implement several of these methods in a programming language not used before, namely in Python utilizing the Pyomo modeling language. Then, experiments will be conducted on the time and computational complexity and what tradeoffs are made in different models between accuracy and performance. This is an area of ongoing research on which several recent papers have been published, including Xu and Hobbs [6] and Zhang *et al.* [7].

**Approach**

In the most general, the TEP problem can be formatted:

Where “*x”* is a set of integer variables representing how many lines have been built between nodes, and “*y”* is a set of continuous variables representing both the hourly generation decisions by the producers as well as how much demand is unmet or excess production is unused. The parameter “*c*” is the cost of lines, often amortized annually, and “*b*” is both the cost of generation and financial penalties of excess or insufficient production. Since the variables are a combination of integers and continuous values, a mixed integer programming solution is required. The equality constraints ensure that conservation of power through the network, while the inequalities deal with limits on supply and demand, as well other issues that may vary with different models.

In a robust model, the TEP can be formatted as a three-level optimization problem, with a top-level objective function that minimizes costs. Costs include yearly annualized infrastructure investments and hourly generation costs of an independent producer. These agents are trying to make optimal decisions against realizations of worst-case scenarios, the bottom two levels. This would be represented as:

Where “*d*” are external events, such as changes in supply, demand, disruptions in generation and so on, within an uncertainty set “*D*”. is the scenario realization for a given *x* and *d*. For robust models, the probability distributions of possible scenarios are not needed, just the upper and lower bounds.

***Problem Formulation***

A more precise definition of the transmission expansion planning problem is as a Mixed-Integer Nonlinear Problem (MINLP) [1][3]. The objective function of this optimization is:

Where “” is hourly generation costs for a vector of generation level, “*Gen*”, at each node, and “” is hourly load shedding costs ,“”, at each node. Finally, is a constant, 8760, to convert hours to years to allow comparison to yearly construction expenditures.

The first constraint ensures that the lines built, “*x*”, are under some predetermined construction budget:

Next, the equality of electrical flow through the system is represented by:

This equality states that load shed at each node “*i*” is equal to the sum of electricity generated, “”, and inflow, “”, minus demand, “”, and outflow, ““.

Third, the values the stochastic parameters can take on are bounded

For a stochastic problem supply and demand will not be assumed to be a set value at each node “*i*” but instead inside a continuous range of possible values between a min and max. The minimum can be set as 0 to represent the possibility of total failure of generation at a node, or given a minimum expected capacity.

Finally, there are the inequalities used to model the electric current with a DC power flow model:

The flow across any built line “*l*” must equal the susceptance, “”, of the line times the voltage angle, “”, at the line’s sending node, “”, minus the voltage at the line’s receiving end “”. The last two constraints impose bounds on the voltage angles and set the reference voltage angle to 0 for convenience.

This model can be expanded from this base setup, for example by allowing several different production sources at each node to represent separate fossil, wind and solar production. On the other hand, if it is simplified by assuming no electrical reactance by deleting the final constraints, then it simply becomes a variation on a transshipment problem. Lastly, if the electrical flow is modeled with an AC power model, a better approximation of real-life conditions, the feasible region can become highly nonconvex. For that reason, most schemes avoid an AC power model [6]

[mention something about adaptive]

***Column-and-Constraint Method***

An MINLP three-level problem is quite unwieldy in any realistic application, and some simplification is desirable. In the robust method of [3], this is accomplished by splitting the model into two parts, a master problem and a sub problem. The subproblem takes in values of “*”* (the lines built) as a fixed parameter, and combines the two-stage inner problem of minimizing the hourly costs of the worst-case realization into a linear problem using KKT optimality conditions. This reformation into a mixed integer linear problem (MILP) allows for efficient algorithms for solving linear systems to be applied.

**Uncertainty Budget**

A unique feature of the constraint-and-column method is the inclusion of an uncertainty budge, , that allows planners to consider realizations that are less than absolute worst-cases. It is represented as:

Here, and are the upper and lower bounds of the realizations of “” within the uncertainty set “*D*” for each “k” elements of d. At the simplest, “k” could just be two elements, supply and demand, but these can be further subdivided into short-term and long-term or by type of electrical generation. Next, “r” is a group of nodes to represent a certain geographical region. Regions can have different uncertainty budgets, allowing different parts of the network to be modeled with different reliability, or to model variability in forecasting future supply and demand. Lastly, , is a reference value. When is 0, so there is no uncertainty, but with, doesn’t matter since can take any value within the range, including the worst-case. Thus, is the traditional robust optimization, and represent some moderate level of uncertainty. In this scheme, the uncertainty budget is included by adding the following two constraints to general formulation:

A special note is represents possible generation capacity, of which since a producer need not use full capacity even if available.

**Subproblem**

The subproblem attempts to find the most expensive scenario that could develop for any given arrangement of lines built, “”, assuming producers are still making optimal decisions Thus, it provides a possible upper bound of the minimization. It is formulated:

(1)

(2)

(3)

(4)

(5)

(6)

The first constraint, is the uncertainty budget constraint. In this model, this includes future growth in electrical demand, and possible fluctuations in production due to either increased future capacity or short-term shortages, such as from weather nonconductive to solar and wind production. The rest of the constraints (2) – (5) of the subproblem come from the application of the KKT conditions, with and as the dual variables for the equalities (3) and inequalities (4) respectively. The second constraint, (2), ensures the differential of the Lagrangian with respect to the lower level variables is zero. (3) maintains the balance of power being transported through the system using a DC power model of the system. A DC power flow is a simplification of the actual AC transmission that has the benefit of maintaining a convex feasible region. (4) is the primal feasibility of all the inequalities, and those used vary depending on the complexity of the model desired. Examples include penalties for either excess production or unmet demand. (5) is the complementarity condition, ensuring that any dual variable cannot be zero and (6) is the dual feasibility condition.

Unfortunately, the complementarity conditions are nonlinear, so [3] suggests linearizing the final three conditions using the Fortuny-Amat transformation utilizing a big “*M*” and a vector of binary variables “*z*”. Thus, replace (5) and (6) with the following linear constraints:

Now that the subproblem is a MILP, it can be efficiently solved with common solvers.

**Master Problem**

The master problem, on the other hand treats the stochastic variables, “”, as fixed parameters and the choice of electrical grid, “*x*”, as a variable. The objective function minimizes both the cost of building lines, , and the operating costs for the external events found in the subproblem. It is minimized against not just the last iteration of the subproblem, “*k*”, but against every iteration of the subproblem .

The constraints are the same as found in the general formulation. The master problem is also a MILP and as such can be solved efficiently with current software.

**Algorithm**

In the final algorithm, the subproblem and master problem are repeated back in forth, finding upper and lower bounds until those bound converge as follows:

* Setup) Find the cost of just the preexisting lines, if any, at the start of simulation and set this as an absolute lower bound. Set . Then, go to Step 2.
* Step 1) Input and from all previous *k* iterations of the subproblem into the master problem to solve and find a new set of lines, . Update the lower bound with the value of the objective function.
* Step 2) Input into the subproblem and solve to find a new set of and . Update the upper bound if it is lower than the previous upper bound.
* Step 3) If the upper and lower bound are within a predefined epsilon, then stop. Otherwise, increment to iteration . Return to Step 1.

Upon completion there will be an ideal electrical grid plan . [Mention about Adaptive]

***Other Methods***

Methods employed by [4] are mathematically close to constraint-and-column and mainly vary in how parameters are set, so implementing the differences should not be difficult. Time permitting, Jabr has proposed a model using a Benders decomposition scheme, by combining the inner terms into a dual variable problem. Finally, Minguez and Garcia-Bertrand introduced a new system in 2016 that, in certain situations, is orders of magnitude faster than the previous methods on larger data sets. It makes some compromises, such as not allowing a change of the uncertainty budget. However, the ability to find a fairly good solution quickly may trump a slightly better solution much slower, especially for users on the initial stages of a planning cycle where many revisions must be run and rerun. The advantages and disadvantages of each method is an ongoing source of research, and in the process of implementation of these methods in a new programming language, more insight may be gained

Into the spring there is more flexibility where to pursue research. If any further developments emerge in designing faster, but even more specialized algorithms, they can be integrated into existing code. Otherwise, since all the models presented make simplifying assumptions, it would be worthwhile attempting to identify edge cases or if the models break down in certain areas, to see how robust these robust optimizations are. More information always means more time to compute, so it can help to identify where tradeoffs in performance for accuracy are happening. Furthermore, comparisons of time complexity of these models implemented in Python would be a worthwhile resource to community. It is possible that implementing in GAMS, as most of the previous work has, will always be faster or have other advantages, but experimenting with a new language and sharing the results will still be valuable.

**Implementation**

Software implementation will be on Pyomo, an optimization modeling language written in Python [8][9]. Since Pyomo is an open source software in continued development, there is a potential user base of the proposed implementations. Several TEP models have already been designed in Pyomo, but not yet the robust models mentioned above. Most importantly, since most other work on TEPs has used algebraic modelling software that require users to purchase a license, such as the GAMS algebraic language, there is demand for development on free platforms such as Pyomo.

Both Pyomo and the software used in previous literature will often call third party solvers to do the heavy lifting of computing the mixed integer program. For example, many experiments implement the IBM developed CPLEX solver. So, by calling CPLEX through Pyomo, a very direct comparison can be made to time complexity in previous attempts. Of course, if a user desires a different solver, such as the open source GLPK, it only takes a slight change.

To test maximum portability with non-state-of-the-art hardware, code will initially be run on a laptop with an Intel Celeron N2840 2.16GHz dual core processor and 4 GB of RAM. Scaled up models have been tested on a computer having 8 quad core processors with 256 GB of RAM, so solving more complex models on advanced hardware could serve as a later project goal depending on equipment and time availability.

**Validation / Testing**

Since the current program is just the implementation of an existing model in a new language, it will be straightforward to verify that constructed models are working. The simplest toy problem to work on is a probabilistic version of the Garver 6-bus problem, which is a well-studied model of a TEP [10][11]. This data set includes 6 nodes, each with upper and lowers bounds for both electrical supply and demand. In addition, there is the hourly generation cost and an assigned financial penalty for not meeting demand. Note these costs are all per hour.

***Basic 6-Bus Nodes***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Node* | Demand | Generation | Gen Cost | Unmet Penalty |
| **1** | 64-96 | 75-225 | 15 | 112.5 |
| **2** | 192-288 | 0 | - | 115 |
| **3** | 32-48 | 175-525 | 10 | 120 |
| **4** | 128-192 | 0 | - | 110 |
| **5** | 192-288 | 0 | - | 112 |
| **6** | 0 | 300-900 | 20 | - |

Furthermore, if using an uncertainty budget set at less than 1 (absolute worst case), the supply can be bound below by zero. This can be used to simulate partial failure of generation in the system. Of course, when using a worst-case robust model (uncertainty budget = 1), an estimated minimum must be entered. Another possible change to this basic setup is assuming there is an amount of green power at each node that has no cost to generate, and there is a penalty, a shedding cost, if all capacity is not utilized.

In addition, there are a different two-way routes between the nodes at which lines can be built. Each route has a build cost (annualized), maximum transmission capacity, and electrical reactance. In this particular data set, every connection is assumed valid. Furthermore, to simulate an existing network at the start of the model, the initial can be set to non-zero values. Lastly, the original Garver 6-bus model assumes that up to three lines can be built between nodes, and those lines on the same route have the same properties, including cost.

***Basic 6-Bus Lines***

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Line | Route | Cost *x 1M* | | Capacity | Susceptance | Line | Route | Cost *x 1M* | Capacity | Susceptance |
| **1** | *(1,2)* | 7.7232 | 100 | | 500 | **9** | *(2,6)* | 5.7924 | 100 | 500 |
| **2** | *(1,3)* | 7.33704 | 100 | | 500 | **10** | *(3,4)* | 11.39172 | 82 | 500 |
| **3** | *(1,4)* | 11.5848 | 80 | | 500 | **11** | *(3,5)* | 3.8616 | 100 | 500 |
| **4** | *(1,5)* | 3.8616 | 100 | | 500 | **12** | *(3,6)* | 9.26784 | 100 | 500 |
| **5** | *(1,6)* | 13.12944 | 70 | | 500 | **13** | *(4,5)* | 12.16404 | 75 | 500 |
| **6** | *(2,3)* | 3.8616 | 100 | | 500 | **14** | *(4,6)* | 5.7924 | 100 | 500 |
| **7** | *(2,4)* | 7.7232 | 100 | | 500 | **15** | *(5,6)* | 11.77788 | 78 | 500 |
| **8** | *(2,5)* | 5.98548 | 100 | | 500 |  |  |  |  |  |

After that the IEEE 24-bus problem should be workable, with results generated in a matter of minutes on a regular laptop. If resources are available, scaling up to the IEEE 72 or even 118 bus system is also possible. However, even while using several dozen cores and 100s of GB of memory, it could take up to several hours at the worst to solve this type of problem on these large data sets.

Finally, we have timing benchmarks for several of the models when implemented on different software, mainly GAMS. Since a major goal of the project is writing software that is desirable to a user, the necessity of being competitive with these times will be a major concern. Of course, it could be hoped to surpass these benchmarks by implementing current research into the speeding up the problem, but the feasibility of this will need to be determined with further experimentation.

**Deliverables / Documentation / Distribution**

Deliverables will consist of mid-year (December) and end of year (May) reports as well as presentations. Throughout, an organized series of results are to be provided in a format to be determined. Finally, a set of fully documented code that is usable by third parties will be shared throughout the life of the project. Both the database of information and code aims to be user friendly and able to be integrated into further research with minimal difficulty. For this purpose, the use of Github can serve as a repository for current updates, with revision and improvement after serious milestones are reached. Towards the end of the year the possibility of a full user guide will be assessed, although it is probably not needed in the short term while the program is being constantly changed.

**Results**

To demonstrate the use of the constraint-and-column method introduce by [3], the Garver 6-bus problem, stated above, is solved. The demand uncertainty budget is set to 1 (worst case) and the generation supply uncertainty budget to 0.4. All possible connections are viable, three lines can be built on each route, and there is no existing infrastructure at the start. For this simple model, no loses due to electrical resistance are included, making it a pure transshipment formulation:

**Step Cost # of Lines in Route Notes**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |  | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |  |
| Sub | 8.3549 x 108 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | No Lines Yet |
| Master | 5.9702 x 108 | 0 | 0 | 0 | 2 | 0 | 1 | 0 |  | 0 | 3 | 0 | 2 | 1 | 0 | 2 | 0 | Lines Added |
| Sub | 5.7924 x 108 | 0 | 0 | 0 | 2 | 0 | 1 | 0 |  | 0 | 3 | 0 | 2 | 1 | 0 | 2 | 0 |  |
| Master | 5.5387 x 108 | 0 | 0 | 0 | 1 | 0 | 3 | 0 |  | 0 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | Change of Lines |
| Sub | 5.5387 x 108 | 0 | 0 | 0 | 1 | 0 | 3 | 0 |  | 0 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | Convergence |

Here, “cost” is the combined price of the infrastructure (annualized) and hourly operating/penalty costs. The 15 rows represent the 15 connections between nodes, , and the corresponding number in each column is the number of lines built on that route. First, the optimal of the first subproblem is calculated with no preexisting lines, so most of the outlay comes from the penalties for not meeting demand. Then, calling the first run of master problem results in lines being added, and the subproblem further optimizes the hourly costs. Then, the second call to the master problem finds a better distribution of power lines. Finally, the third execution of the subproblem optimizes to the same value. Thus, the upper and lower bounds are equal, and the program is done. Below is the final state of the network, with demand being fully met everywhere.

A picture containing text

Description generated with high confidence

Next is a demonstration with the same parameters but now a limit of only one electrical line per route. In this solution, demand is not fully met at nodes 4 and 5 due to total demand being above what can be transported even with all possible lines being built.

A close up of text on a white background

Description generated with very high confidence

Here is a table of the results for solving the above after each sub step. The final cost is more than with three lines allowed, both because of need to build less efficient connections and the penalties from not being able to meet full demand.

**Step Cost # of Lines in Route Notes**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |  |
| Sub | 8.3549 x 108 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | No Lines Yet |
| Master | 6.5592 x 108 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | Lines Added |
| Sub | 6.2527 x 108 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |  |
| Master | 5.9055 x 108 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | Change of Lines |
| Sub | 5.9055 x 108 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | Convergence |

In fact, most variations on the parameters of a 6-bus data set tested do not even require a second call of the master problem, since an optimum is found on the first pass. Furthermore, the time to execute the program with the Garver 6-bus data set over several attempts is a matter of seconds on a laptop. This is in rough agreement with experiments with GAMS code, although further tests should be done, with both programs being run on the same hardware to establish a true comparison.

**Analysis**

The constraint-and-column method does make many simplifying assumptions that could introduce errors, especially in assuming the linearity of certain parameters. One that seems unreasonable is that the cost of building multiple electrical lines on the same route is a direct relationship. To address this, each line on a connection could be modeled with a separate cost, but this could increase the number of variables greatly. On the subject of cost assumptions, power generation is also assumed to be linear, which does seem somewhat realistic. However, nothing in the model prevents a power plant from being tasked to run at a very small percentage of max capacity, and that does not seem a plausible expectation in actual economic practice.

The issue of whether the power network would be best modelled as a simple transshipment problem, a DC load flow, or some other approximation is still much debated in the literature. There has been some research to suggest that a DC based model offers poor increases in accuracy for the computational cost [6], and this question will be explored as the project progresses in the spring. Furthermore, the central assumption underlying the whole constraint-and-column method is that the inner bilevel problem, of minimizing hourly costs against a worst-case realization, can be combined into a one-level linear problem by using the KKT conditions and the Fortuny-Amat transformation. It may be useful to attempt to identify if the model stands up empirically through repeated experiments on different data sets, or find what conditions cause these assumptions to break down.

Of special note, the element that separates the work of Ruiz and Conejo [3] from similar approaches is their inclusion of the uncertainty budget. Essentially, it is a model-wide or regional adjustment of the ranges of uncertain parameters. In fact, adjusting the uncertainty budget on a node by node basis is functionally equivalent to adjusting the input parameters for each node. As such, it seems hard to justify including it except as a minor convenience. Possibly it could serve as a way of identifying if the optimal solution found is highly sensitive to small changes in the model by quickly looping through models with slightly varied budgets.

On the code itself there are three identified areas for improvement that should be addressed as the project moves forward. The first is that data is being entered in a somewhat clumsy manner using AMPL data files. To address this, Pyomo has a Pyomo Network package specifically designed for transportation problems that may turn out to be a more user-friendly alternative. The second is the code needs to be reexamined for possible inefficiencies or fragilities to further the goal of making software that is desirable to the community. The final is that all possible parameters, such as differentiating between renewable and generated power, have not been included. However, since maximum flexibility and development best practices has been built in from day one, these changes should all be relatively minor.

Into the spring, the project will implement in Pyomo the three other models put forth to address the TEP with robust optimization.Then, the time complexity for several different data sets can be compared across implementations. Resource permitting, scaling up to larger models may better demonstrate the efficiency of each model. Finally, code and documentation need to be updated to an acceptable standard for dissemination through a potential user base, and results collated for the final presentation.

**Nodes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Node* | Demand | Generation | Gen Cost | Unmet Penalty |
| **1** | 200 | 300 | 18 | 40 |
| **2** | 0 | 250 | 25 | - |
| **3** | 0 | 400 | 16 | - |
| **4** | 150 | 0 | - | 52 |
| **5** | 100 | 300 | 32 | 55 |
| **6** | 200 | 150 | 35 | 65 |

***Basic 6-Bus Lines***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Line | Route | Cost *x 1M* | Capacity | | Susceptance |
| **1** | *(1,2)* | 0 | | 150 | 500 |
| **2** | *(1,3)* | 0 | | 150 | 500 |
| **3** | *(2,3)* | 7.000 | | 150 | 500 |
| **4** | *(2,4)* | 14.000 | | 200 | 500 |
| **5** | *(3,4)* | 18.000 | | 150 | 500 |
| **6** | *(3,6)* | 16.000 | | 200 | 500 |
| **7** | *(4,5)* | 0 | | 150 | 500 |
| **8** | *(4,6)* | 8.000 | | 150 | 500 |
| **9** | *(5,6)* | 7.000 | | 150 | 500 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Cost** | **in Master** | **Lines in Route** | **Notes** |
| Master | 0 | 0 | (1,2) (1,3) (4,5) | Already Existing Network |
| Sub | 1.73448 x 108 | - | (1,2) (1,3) (4,5) |  |
| Master | 1.08558 x 108 | 1.05558 x 108 | (1,2) (1,3) (4,5) **(2,4) (3,6)** | **Lines Added** |
| Sub | 1.08558 x 108 | - | (1,2) (1,3) (2,4) (3,6) (4,5) | Convergence |

**References**

[1] Antonio J. Conejo, Luis Baringo Morales, S. Jalal Kazempour, and Afzal S. Siddiqui. 2016. Investment in Electricity Generation and Transmission: Decision Making Under Uncertainty (1st ed.). Springer Publishing Company, Incorporated.

[2] Jabr R.A. “Robust Transmission Network Expansion Planning with Uncertain Renewable Generation and Loads.” *Ieee Transactions on Power Systems*, vol. 28, no. 4, 2013, pp. 4558–4567., doi:10.1109/TPWRS.2013.2267058.

[3] Ruiz, C, and A.J Conejo. “Robust Transmission Expansion Planning.” *European Journal of Operational Research*, vol. 242, no. 2, 2015, pp. 390–401., doi:10.1016/j.ejor.2014.10.030.

[4] Bokan Chen, et al. “Robust Optimization for Transmission Expansion Planning: Minimax Cost Vs. Minimax Regret.” *Ieee Transactions on Power Systems*, vol. 29, no. 6, 2014, doi:10.1109/TPWRS.2014.2313841

[5] Mínguez R, and García-Bertrand R. “Robust Transmission Network Expansion Planning in Energy Systems: Improving Computational Performance.” *European Journal of Operational Research*, vol. 248, no. 1, 2016, pp. 21–32., doi:10.1016/j.ejor.2015.06.068.

[6] Q. Xu, B.F. Hobbs, “Value of Model Enhancements: Quantifying the Benefit of Improved Trans-mission Planning Models”, Working Paper, Dept. Environmental Health & Engineering, JHU, 2018.

[7] Zhang, Xuan, and Antonio J Conejo. “Robust Transmission Expansion Planning Representing Long- and Short-Term Uncertainty.” *IEEE Transactions on Power Systems*, vol. 1, 2017, pp. 1–1., doi:10.1109/TPWRS.2017.2717944

[8] Hart, William E., Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson, and John D. Siirola. *Pyomo – Optimization Modeling in Python*. Second Edition. Vol. 67. Springer, 2017.

[9] Hart, William E, and David L Woodruff. “Pyomo: Modeling and Solving Mathematical Programs in Python.” *Mathematical Programming Computation*, vol. 3, no. 3, 2011, pp. 219–260.

[10] L. L. Garver, "Transmission Network Estimation Using Linear Programming," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 7, pp. 1688-1697, Sept. 1970.

[11] Romero, R., A. Monticelli, Ae Garcia, and S. Haffner. "Test systems and mathematical models for transmission network expansion planning." IEE Proceedings-Generation, Transmission and Distribution 149, no. 1 (2002): 27-36.