A SURVEY OF RECENT ADVANCES IN TRANSMISSION NETWORK SWITCHING
Transmission Switching

- Modify network configuration using a switching heuristic to improve a desired network metric
- Drive optimization using line power loss, economic loss, outage frequency, etc. objective functions
- Maintain network health using equality/inequality constraints and tolerances
- Avoid local minima in the search space using stochastic methods and smart tempering in the search heuristic

Objective Function

[4]

Photo courtesy of Siemens AG
Exhaustive Search

- **Binary Integer Programming**
  - Binary valued switch variables
  - Simple modeling
  - Configuration restricted to the binary decision vector
  - Examples: genes in GA, switch vectors
  - Many MIP problems can be converted into BIP

- **Mixed Integer Programming**
  - Real-valued or integer valued variables
  - More common in modern transmission switching
  - Can be used to solve BIP problems
  - Unique uses for MIP: multiple lines per edge or shunts per node
Meta-Heuristic

- A combinatorial heuristic optimization that reduces search based significantly compares to exhaustive search techniques
- Most optimization strategies in the literature use some sort of meta-heuristic to drive optimization
- A compromise between problem dimensionality and solution speed
- Many of the heuristic procedures covered in this presentation date back at least 15 years, and as early as 1975 (genetic algorithm)

Hyper-Heuristic

- As problem complexity becomes less predictable hyper-heuristic procedures may allow us to optimize the algorithm to the model
- As of this date very little research has applied hyper-heuristics to TS, most likely due to a much more increased overhead of computational resources
- At this point hyper-heuristic is a purely speculative procedure for automatic transmission switching but may be worth investigation in future research
Approaches in Literature

- Genetic Algorithm (57*)
- Swarm Optimization (28)
- Simulated Annealing (27)
- Evolutionary Algorithm (19)
- Colony Optimization (13)
- Tabu Search (5)
- Memory Based Heuristic (1)
- Immune Algorithm (1)

* - # of results on IEEEXplore; searched for <method name> & “network loss reduction” & “power, energy, & industry applications”
Network Fitness Criteria

- **Objective Functions**
  - Loss Reduction
  - Operation Expenditure
  - Load Balancing
  - Violation Penalties
  - Service Restoration
  - Network Overloads
  - Voltage Profile
  - Frequency Droop
  - Reliability

- **Constraints**
  - Line Current
  - Voltage
  - Phase Angle
  - Generation Limitations
  - Dispatch Control

- **Topology**
  - Avoid islanding (in most scenarios)
  - Maintain connection of all generation and load
Radial vs. Meshed Network

**Radial Network**
- Fast computation time
- Better convergence
- Simple power flow equations
- Radial networks are generally associated with distribution models

**Meshed Network**
- More computationally intensive
- Possibility of lower convergence
- For compatibility with radial solvers, meshed models can be converted to a radial model
  - Create loop break point (LBP) dummy buses
  - An extra calculation must be performed to readjust these dummy injections
- For our purposes, we will perform OPF using conventional power flow software
- Meshed networks are more commonly seen in large transmission models
Common Assumptions:

- Line Loss is the primary metric for network health, but may also be complimented by other measures.
- In some instances meshed networks are reduced to purely radial networks, using loop edges as interchangeable switches, and ensuring radial topology.
- Models are balanced $3\varphi$, use generalized $1\varphi$ model in simulation.
- DCOPF is used in order to reduce computation time.
- Optimality is not always guaranteed, most solutions are feasible and healthier than the initial model.
- Ensure all loads and generation are connected to the network on each iteration.
Genetic Algorithm[1]

- Objective Function: loss reduction
- Candidate pool consists purely of available, initially open lines (represented by tie lines in the model)
- The genetic string “chromosome” is represented by a vector of “genes”, binary values associated with the open/close position of each line in the candidate pool

<table>
<thead>
<tr>
<th>Chromosome</th>
<th>a₁</th>
<th>a₂</th>
<th>a₃</th>
<th>a₄</th>
<th>a₅</th>
<th>...</th>
<th>aₙ</th>
</tr>
</thead>
</table>

Genes
Genetic Algorithm[1]

- **Algorithm:**
  1. Seed Population
  2. Reproduction
  3. Crossover
  4. Mutation
  5. Calculate Fitness
  6. Evaluate Convergence

- Return to 2. if convergence criteria is not met

\[ k = \text{round}(\text{rand}(1,N-1)) \quad \text{index } k \]

\[ P(\uparrow) = 0.5 \]
Genetic Algorithm\[2\]

- Revisions:
  - Crossover
    - Matched pair gene swapping is random per gene, no indexing limit or restriction is placed
    - Previous crossover method favored front end gene elements
  
  \[
  \begin{array}{cccccccc}
  a_1 & a_2 & a_3 & a_4 & a_5 & \ldots & a_N \\
  b_1 & b_2 & b_3 & b_4 & b_5 & \ldots & b_N \\
  \end{array}
  \]
  
- Mutation
  - Adaptive mutation; mutation rate decreases as minimum line loss of the population converges
Simulated Annealing \[3][4]

- Objective Function: loss reduction

- Simulate the phenomenon of annealing as applied to materials, utilize entropic behavior to escape local minima

- In addition to generic SA a perturbation mechanism is introduced to guide the search using the knowledge of system topology, loop length and distance from switch determine the next switch selection
Simulated Annealing \[3\][4]

- **SA Algorithm:**
  1. Initialize (temp, opt. config.)
  2. Set iteration limit per temperature/iteration schedule
  3. Move
     - Decrease – Accept
     - Increase – Accept/Deny depending on Temperature
  4. Detect Convergence
     - Criteria met - END
  5. Reset iterations, decrease temp, go to 2.
Objective Function: loss reduction

Classical descent method of move, compare, update

Tabu List provides a means of memorizing previous moves, moves that are “taboo” for new moves

Perturbation mechanism is used to avoid local minima

- Add/Subtract Move - random branch exchange followed by a complete sequence of branch exchange with all lines in the new loop, remove the line leading to minimum losses

- Multiplicative Move - perform branch exchange on a random number of tie lines available for swapping

- Constrained Multiplicative Move - limit the number of multiplicative moves
Tabu-Search

Algorithm:

1. Initialize
   - Tabu List length

2. Perform a move from the perturbation mechanism list
   - If move exists, perform another move
   - Else add to tabu list and save as best candidate. If tabu list is full remove oldest member of list

3. Check for convergence
   - Return to 2. if convergence is not satisfied
Objective Function: loss reduction/ violation penalty

DE is a modified Evolutionary Algorithm utilizing a unique mutation method

Differential vectors form a mutant population

A scaling factor is used to perturb mutant individuals to an even greater degree

\[ F^{t+1} = \begin{cases} 
    c_d * F^t, & \text{if } p^t_s < 1/5 \\
    c_i * F^t, & \text{if } p^t_s > 1/5 \\
    F^t, & \text{if } p^t_s = 1/5 
\end{cases} \]

Scaling factor begins at \( F^0 = 1.2 \), and scales based on the frequency of successful mutations in a generation
Algorithm

1. Initial Population
   - Create a population of configurations uniformly distributed on the entire parameter space

2. Mutant Population
   - Randomly select 2-4 unique individuals, create a difference vector, multiply by scaling factor and merge it with a seed individual to create a mutant individual

3. Population Crossover
   - Randomly pair a seed and a mutant.

4. Choose best candidate of the generation
   - Determine the best candidate of the current generation. If the candidate is more fit than the best candidate of the parent generation, retain it. If not continue a new generation with the retained candidate.

5. Perform migration if population diversity is not met in initial population of a new generation
   - Using the best candidate from the previous generation, a randomized mutation is performed to create a new population.

6. Convergence check/Update scaling factor
   - Scaling factor is updated
 Binary Particle Swarm Optimization

- Objective Function: maximize reliability
- Particles contain two vectors:
  - Position — binary vector containing switch information
    \[ x_i(t) = x_i(t - 1) + v(t) \]
    \[ x_{ij}(t) = \begin{cases} 
    1 & \text{if } \rho_{ij} < s(v_{ij}) \\
    0 & \text{for all else}
    \end{cases} \]
    \[ s(v_{ij}) = \frac{1}{1 + \exp(-v_{ij})} \]
  - Velocity — real number vector dictating the movement of the particle
    \[ v_i(t) = v_i(t - 1) + \varphi_1 \cdot r_1 \cdot (x_{i,\text{best}} - x_i(t - 1)) + \varphi_2 \cdot r_2 \cdot (x_{\text{best}} - x_i(t - 1)) \]
Binary Particle Swarm Optimization

Algorithm:
1. Initialize
   - Create a population of randomly configured particles, nil velocity
   - Calculate initial reliabilities
2. Perform feasibility check
   - Infeasible position vectors are given a heavy penalty in their fitness
3. Update position and velocity vectors
4. Particle with maximum reliability is saved and analyzed for convergence
Ant Colony Optimization

- Objection Function: loss reduction
- Ant Colony Optimization models the hunt/gather/communicate dynamic search pattern of ants
- Distance and frequency of successful moves influence the movement of each “ant”
- Good moves increase the pheromonal value of a line
Ant Colony Optimization

**Algorithm:**

- **Initiate**
  - Individuals start out on a random loop element, pheromonal value of all loop elements is initially equal
  - The authors also initialize the search with a super-ant using quick optimal path search tool

- **Move**
  - The ant moves to another loop element based on two factors: pheromonal value and distance. These two values are weighted through tuning
  - A good move will update the pheromonal value of the line element, increasing the likeliness it will be utilized by other ants
  - Eventually all the individuals create a unique radial network, the best of these individuals is selected as the heuristic spark for the next iteration
  - The network configuration is perturbed on each iteration by randomly branch swapping several loop elements, the number of swaps is usually around 2-5% of the ant population

- **New generation**

- **Stop search when convergence criteria is met for the best individuals**
\( \text{ObjFunc} = \text{(sum of load payments)} - \text{(generation gross margin)} - \text{(merchandising surplus)} \)

- Nodal price – evaluated using generation, load and network limitations
- Merchandising Surplus – a product of nodal price matrix and excess generation (gen – load)
- Switches available in the search space include only lines that are initially open
- Profitable and Un-Profitable lines are evaluated, Un-Profitable lines are selected as switch candidates
- Power transfer distribution factor (PTDF) and line outage distribution factor (LODF) are calculated to determine the economic effects of a line outage which in turn determine which line to cut in the next iteration.
- Stopping criteria is either unlimited or limited to \( I \) iterations
Other Methods

- Immune Network -
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1193637

- Artificial Neural Network -
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=252662

- Rank Removal -
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4436100

- Fuzzy Reconfiguration -
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=756119

- Tabu-Mutation Hybrid -
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=756120

- Unbalanced Phase Swapping for Distribution Networks -
  http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4104582
## Literature Methodology

<table>
<thead>
<tr>
<th></th>
<th>GA</th>
<th>SA</th>
<th>Tabu</th>
<th>DE</th>
<th>ACO</th>
<th>ETTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bus</td>
<td>8.9</td>
<td></td>
<td></td>
<td></td>
<td>8.9</td>
<td></td>
</tr>
<tr>
<td>19-bus</td>
<td>15.7</td>
<td>15.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30-bus</td>
<td></td>
<td>31.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-bus</td>
<td></td>
<td></td>
<td>31.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33-bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31.1</td>
<td></td>
</tr>
<tr>
<td>69-bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>39.7</td>
</tr>
<tr>
<td>70-bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11.1</td>
</tr>
<tr>
<td>96-bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11.7</td>
</tr>
<tr>
<td>IEEE-118</td>
<td></td>
<td></td>
<td></td>
<td>16.0</td>
<td></td>
<td>9.7</td>
</tr>
<tr>
<td>135-bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.7</td>
</tr>
<tr>
<td>148-bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19.5</td>
<td></td>
</tr>
<tr>
<td>362-bus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>79.0</td>
</tr>
<tr>
<td>1692-bus</td>
<td>19.4</td>
<td>33.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

% Reduction by Model
Approach to the Problem

- TransEx is primarily transmission expansion tool, with some parameter restrictions can it be retooled for Optimal Transmission Switching?
- TransEx uses two heuristic methods for optimization: Limited Discrepancy Search and Randomized Discrepancy Bounded Local Search
- We will be comparing LDS and RDS with other methods
  - Test Systems:
    - IEEE-14 (debug purposes)
    - RTS-96 73 bus system
    - Modified RTS-96 73 bus system
    - IEEE-118 (most common metric, for later benchmarking)

- Most methods presented performed switching optimization on radial distribution networks; we will be focusing on meshed networks
TransEx Configuration

- Existing Corridors:
  - MaxLines = number of lines in model corridor
  - MinLines = 0
- No new corridors, lines shunts, transformers, voltage upgrades, etc.
- Linear DCOPF
- LDS and RDS

- Static Line and Bus constraints
- Objective Functions:
  - Load Shedding
  - Line Overload
  - Line Loss
  - Economic Loss (future simulation)
As established by O’Neill et al.\cite{18} RTS-96 test system is frequently modified for optimal switching analysis

- Modifications:
  - Remove line (11-13)
  - Shift 480 MW from buses 14, 15, 19, 20 to 13
  - Add generation capacity to:
    - (1) – 100 MW
    - (7) – 100 MW
    - (15) – 155 MW
    - (23) – 155 MW
  - Decrease thermal capacity of line (14-16) to 350 MW
## Results

<table>
<thead>
<tr>
<th>LDS Results</th>
<th>Time (min)</th>
<th>Initial Loss (kW)</th>
<th>Final Loss (kW)</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-14</td>
<td>2.326</td>
<td>15.994</td>
<td>2.559</td>
<td>84.0</td>
</tr>
<tr>
<td>RTS_96</td>
<td>4.235</td>
<td>361.998</td>
<td>322.178</td>
<td>11.0</td>
</tr>
<tr>
<td>RTS_96m</td>
<td>5.791</td>
<td>423.915</td>
<td>377.522</td>
<td>10.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RDS Results</th>
<th>Time (min)</th>
<th>Initial Loss (kW)</th>
<th>Final Loss (kW)</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-14</td>
<td>2.312</td>
<td>15.994</td>
<td>2.559</td>
<td>84.0</td>
</tr>
<tr>
<td>RTS_96</td>
<td>7.623</td>
<td>361.998</td>
<td>350.192</td>
<td>3.26</td>
</tr>
<tr>
<td>RTS_96m</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
References


10. R. Bent, A. Berscheid, and G.L. Toole, "Transmission Network Expansion Planning with Simulation Optimization", in Proc. AAAI, 2010


