The Information-Collecting Vehicle Routing Problem: Stochastic Optimization For Emergency Storm Response

Lina Al-Kanj, PhD

Associate Research Scholar, Operations Research and Financial Engineering Department, Princeton University
Princeton, NJ
Outline

- Problem Description
- Probability Model for Grid Faults
- Sequential Stochastic Optimization Model for Utility Crew Routing
  - Multistage Lookahead Policy (Monte Carlo Tree Search)
  - Monte Carlo Tree Search with Information Relaxation Dual Bounds
- Stochastic Optimization Model for Sensor and Protective Device Placements
Problem Description – Power System
Problem Description - Emergency Storm Response
Problem Description - Emergency Storm Response
Problem Description - Emergency Storm Response
Problem Description - Emergency Storm Response
Problem Description - Emergency Storm Response
Problem Description - Emergency Storm Response
Problem Description - Emergency Storm Response

Diagram showing emergency storm response with symbols for buildings, houses, transformers, protective devices, power lines, poles, and roadways. The diagram includes a substation and various call points denoted by 'call'.
Outline

• Problem Description

• Probability Model for Grid Faults

• Sequential Stochastic Optimization Model for Utility Crew Routing
  • Multistage Lookahead Policy (Monte Carlo Tree Search)
  • Monte Carlo Tree Search with Information Relaxation Dual Bounds

• Stochastic Optimization Model for Sensor and Protective Device Placements
Problem Description – Grid Restoration

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>set of poles, i.e., $I= {i, i = 1, ..., I}$</td>
</tr>
<tr>
<td>$\mathcal{U}$</td>
<td>set of circuits, i.e., $\mathcal{U} = {u, u = 1, ..., U}$</td>
</tr>
<tr>
<td>$I^u$</td>
<td>set of power lines on circuit $u$, i.e., $I^u = {i, i = 1, ..., I^u}$</td>
</tr>
<tr>
<td>$\mathcal{N}^u$</td>
<td>Set of nodes on circuit $u$, i.e., $\mathcal{N}^u = {N^u_i, \forall i \in I^u}$</td>
</tr>
<tr>
<td>$n_i^u$</td>
<td>Number of customers attached to node $N^u_i$</td>
</tr>
<tr>
<td>$n_{tu}^{u,c}$</td>
<td>Number of customers that called on node $N^u_i$</td>
</tr>
<tr>
<td>$\Omega_t$</td>
<td>sample space at time $t$; each scenario $\omega$ represents phone calls, faults, travel &amp; repair times</td>
</tr>
<tr>
<td>$H_t$</td>
<td>random vector representing the realizations of received calls at time $t$</td>
</tr>
<tr>
<td>$L_t$</td>
<td>random vector representing the realizations of power line faults at time $t$</td>
</tr>
<tr>
<td>$T_{tij}$</td>
<td>random variable representing the travel time from node $i$ to node $j$ at time $t$</td>
</tr>
<tr>
<td>$R^u_j$</td>
<td>random variable representing the repair time of power line $j$ on circuit $u$</td>
</tr>
<tr>
<td>$x_t$</td>
<td>random vector representing the trajectory of the truck at time $t$; $x_t = (x_{tij})_{i,j \in \mathcal{V}}$</td>
</tr>
</tbody>
</table>

Illustration of distribution grid

- Bayes’ Theorem:

\[
p(L_t|H_t) = \frac{p(H_t|L_t)p(L_t)}{\sum_{L_t \in \mathcal{L}} p(H_t|L_t)p(L_t)} = \frac{p(H_t|L_t)p(L_t)}{\sum_{L_t \in \mathcal{L}} p(H_t|L_t)p(L_t)}
\]

Probability Model – Prior Probabilities
Outline

• Problem Description

• Probability Model for Grid Faults

• Sequential Stochastic Optimization Model for Utility Crew Routing
  • Multistage Lookahead Policy (Monte Carlo Tree Search)
  • Monte Carlo Tree Search with Information Relaxation Dual Bounds

• Stochastic Optimization Model for Sensor and Protective Device Placements
Grid Restoration Model

- Objective: Develop an optimal policy that routes the utility truck in order to minimize the number of customers in outage at any point in time.

\[ R_t: \text{Physical State} \]

\[ I_t: \text{Information State} \]

\[ K_t: \text{Knowledge State} \]

Information-Collection utility truck routing

Grid Restoration Model

- Objective: Develop an optimal policy that routes the utility truck in order to minimize the number of customers in outage at any point in time.

- **$R_t$: Physical State**
- **$I_t$: Information State**
- **$K_t$: Knowledge State**

Information-Collection utility truck routing

---

Sequential Stochastic Optimization Model

- Five fundamental elements of sequential stochastic optimization:

  - State $S_t$ - information capturing what we know at time $t$;
    
    $S_t = (R_t, P_t^L, H_t)$ where $R_t$ represents the physical state of the network.

  - Decision $x_t$ - captures the decision made at time $t$;
    
    Let $X_t^\pi(S_t)$ be the policy that determines $x_t \in X_t$ given $S_t$.

  - Exogenous information $W_t$ - new information that arrives between $t - 1$ and $t$;
    
    Includes new arriving calls, travel time, repair time and outages.

  - Transition function - $S_{t+1} = S^M(S_t, x_t, W_{t+1})$ represents the evolution of the states
    
    e.g., $H_{t+1} = H_t + \hat{H}_{t+1}, p(L_{t+1,j} = 1|x_{t,ij} = 1) = 0$ in addition to $W_{t+1}$.

  - Objective function - $\min_\pi \mathbb{E}\pi[\sum_{t=0}^{T} C(S_t, X^\pi(S_t))];$
    
    minimizes the number of customers in outage at any point in time.
Sequential Stochastic Optimization Model

- Cost function: \[ \sum_{t=0}^{T} C(S_t, X_t^\pi(S_t)) \]

**Figure 6**: Objective function; Customer outage-minute is represented by the shaded area under the curve.
Sequential Stochastic Optimization Model

- Optimization problem

\[
\min_\pi \mathbb{E}_\pi \left[ \sum_{t=0}^{T} C(S_t, X^\pi(S_t)) \right]
\]

where

\[ S_{t+1} = S^M(S_t, x_t, W_{t+1}) \]

- Two fundamental strategies for designing policies [5]:
  - Policy Search
  - lookahead Approximations
    - Value function approximation
    - Direct lookahead

Lookahead Approximations

- Lookahead approximations – Approximate the impact of a decision now on the future:
  - An optimal policy (based on looking ahead):
    \[
    X_t^*(S_t) = \arg \max_{x_t} \left\{ C(S_t, x_t) + \max_{\pi \in \Pi} \left\{ \sum_{t'=t+1}^T C(S_{t'}, X_{t'}^\pi(S_{t'})) \mid S_{t+1} \right\} \mid S_t, x_t \right\}
    \]
  2a) Approximating the value of being in a downstream state using machine learning ("value function approximations")
    \[
    X_t^*(S_t) = \arg \max_{x_t} \left( C(S_t, x_t) + \left\{ V_{t+1}(S_{t+1}) \mid S_t, x_t \right\} \right)
    \]
    \[
    X_t^{VFA}(S_t) = \arg \max_{x_t} \left( C(S_t, x_t) + \left\{ \tilde{V}_{t+1}(S_{t+1}) \mid S_t, x_t \right\} \right)
    \]
    \[
    = \arg \max_{x_t} \left( C(S_t, x_t) + \tilde{V}_t^x(S_t^x) \right)
    \]
  2b) Approximate lookahead models – Optimize over an approximate model of the future:
    \[
    X_t^{LA}(S_t) = \arg \max \left\{ C(S_t, x_t) + \tilde{\max}_{\tilde{\pi} \in \tilde{\Pi}} \left\{ \sum_{t'=t+1}^T C(\tilde{S}_{t'}, \tilde{X}_{t'}^{\tilde{\pi}}(\tilde{S}_{t'})) \mid \tilde{S}_{t+1} \right\} \mid S_t, x_t \right\}
    \]
Multistage Lookahead Approximations

- We can then simulate this *lookahead policy* over time:

![Diagram showing the lookahead model and the base model with time points t, t+1, t+2, t+3]
Multistage Lookahead Approximations

- We can then simulate this lookahead policy over time:
Multistage Lookahead Approximations

- We can then simulate this *lookahead policy* over time:
Multistage Lookahead Approximations

- We can then simulate this lookahead policy over time:

The lookahead model

The base model

\[ t \quad t+1 \quad t+2 \quad t+3 \]
Multistage Lookahead Policy

- The optimal policy is computationally intractable, requiring approximations:

\[ X_t^*(S_t) = \arg\min_{x_t \in X_t(S_t)} (C(S_t, x_t) + \mathbb{E}_{\tilde{W}_{t,t+1} \in \tilde{\Omega}_{t,t+1}} \left[ \min_{\tilde{x}_{t,t+1} \in \tilde{X}_{t,t+1}} (\tilde{C}(\tilde{S}_{t,t+1}, \tilde{x}_{t,t+1}) + \mathbb{E}_{\tilde{W}_{t,t+2} \in \tilde{\Omega}_{t,t+2}} \left[ ... \mathbb{E}_{\tilde{W}_{t,t+H} \in \tilde{\Omega}_{t,t+H}} (\tilde{C}(\tilde{S}_{t,t+H}, \tilde{x}_{t,t+H-1}) ... ) \right] \right])_{S_t} \]

where \( \tilde{S}_{t,t'+1} = S^M(\tilde{S}_{t'}, \tilde{x}_{t'}, \tilde{W}_{t,t'+1}) \), \( t' = t, ..., t + H - 1 \)

- Discretizing the time, states and decision
- Limiting the horizon from \((t, T)\) to \((t, t + H)\)
- Dimensionality reduction e.g., by limiting some variables (e.g., fixing the set of calls, limiting the fault types and travel times)
- Aggregating the outcome or sampling by using Monte Carlo sampling
Monte Carlo Tree Search

- MCTS is a recent research area; the first MCTS algorithm has been developed by Chang et al [2005].
- The applications of MCTS are broad and varied, but the strategy is traditionally most often applied to game-play AI like Go.
- MCTS biases the growth of the tree towards the most promising moves which decreases the search space [3].
- MCTS mainly applied for deterministic problems but Coutoux et al [2011] extended it to stochastic problems based on double progressive widening.

Monte Carlo Tree Search

MCTS Construction Steps

- Decision selection based on UCB (Upper Confidence Bounds for Trees) [7]:

\[
\hat{x}_{tt'}^* = \arg\max_{\hat{x}_{tt'} \in \hat{x}_{tt'}^e(\hat{s}_{tt'})} \left( -\left( \tilde{C}(\hat{s}_{tt'}, \hat{x}_{tt'}) + \tilde{V}_{tt'}(\hat{s}_{tt'}^x) \right) + \alpha \frac{\log N(\hat{s}_{tt'})}{\sqrt{N(\hat{s}_{tt'}, \hat{x}_{tt'})}} \right)
\]

Monte Carlo Tree Search – Convergence Theory

- For deterministic MCTS, the UCB policy samples the actions infinitely often and Kocsis and Szepesvari [2006] exploit this to show that the probability of selecting a suboptimal action converges to zero at the root of the tree.

- Auger et al. [2013] provides convergence results for stochastic MCTS with double progressive widening under an action sampling assumption. The asymptotic convergence of also MCTS relies on some form of “exploring every node infinitely often“.

- In Jiang et al. [2017], we design a version of stochastic MCTS that asymptotically does not expand the entire tree, yet is still optimal!

Propose a Primal-Dual MCTS, that takes advantage of the information relaxation bound idea that asymptotically converges [8] to the optimal solution while ignoring suboptimal parts of the tree.

Explore a set of actions:

\[
\hat{u}^n(x^n_{te}, a) = c_{\tau_e}(s, a, W^n_{\tau_e+1}) + \max_a [h_{\tau_e+1}(S_{\tau_e+1}, a, W^n_{\tau_e+1,T}) - z^{\nu}_{\tau_e+1}(S_{\tau_e+1}, a, W^n_{\tau_e+1,T})]
\]

Smooth with previous estimates:

\[
\bar{u}^n(x^n_{te}, a) = (1 - \alpha^n(x^n_{te}, a)) \bar{u}^{n-1}(x^n_{te}, a) + \alpha^n(x^n_{te}, a) \hat{u}^n(x^n_{te}, a)
\]

Expand it if the value is greater than \( \bar{V}^{n-1}(x^n_{te}) \)

Simulation Policy

- Generate sample path $\tilde{\omega}$ from $\tilde{\mathcal{N}}_{t,t'}$ that determines all random events (faults, travel time, repair time).

- Utility truck should visit each fault once to repair it.

- Objective: find the optimal route that minimizes the customer-outage minute.

- In the worst case, computational complexity $O(n!)$.
Simulation Policy – Integer programming

- The problem can be formulated as a non-linear integer program as follows:

\[
\min \sum_{t' \in t'} \left( N - \sum_{t'' \in t'} \sum_{u \in U} \sum_{j=1}^{N} \tilde{C}_{t''}^{u} \right) 
\]

subject to

\[
\tilde{C}_{t''}^{u} = \sum_{i} \left( \prod_{k \in O_{j}} \left( 1 - \tilde{L}_{t''}^{u} \right) \right) \tilde{x}_{t''}^{u} \sum_{k \in S_{j}} n_{k}^{u} + \sum_{s \in W_{j}} \left( \prod_{w=\min\{W_{s}^{u}\}}^{s} \prod_{k \in w} \left( 1 - \tilde{L}_{t''}^{u} \right) \right) \sum_{k \in s} n_{k}^{u} \tilde{x}_{t''}^{u} 
\]

\[
\forall j \in V, \forall t''
\]

\[
1 \tilde{L}_{t''}^{u} = 1 - \sum_{i} \sum_{t' = t'}^{t''-1} \tilde{x}_{t'ij}, \text{ such that } \tilde{L}_{t''}^{u}(\tilde{\omega}) = 1, \forall j \in T^{u}, \forall u \in U, \forall t''
\]

\[
\tilde{\Delta}_{t''}^{ij} \geq T_{ij}(\tilde{\omega}) \tilde{x}_{t''}^{ij} + \sum_{u} R_{j}^{u}(\tilde{\omega}) \left( \tilde{x}_{t''}^{ij} - \sum_{i} \tilde{x}_{t'ij} \right), \forall (i, j) \in A, \forall t''
\]

\[
\tilde{\xi}_{t''}^{ij} \geq \tilde{\xi}_{t''-1}^{ij} + \sum_{i} \tilde{\Delta}_{t''}^{ij}, \forall (i, j) \in A, \forall t''
\]

\[
\tilde{\xi}_{t''}^{ij} \leq \xi_{t''}^{ij} + \zeta \left( 1 - \sum_{i} \tilde{x}_{t''}^{ij} \right), \forall j \in V, \forall t''
\]

\[
\sum_{t'' = t'}^{t'+H} \tilde{x}_{t''}^{ij} \leq 1, \forall (i, j) \in A
\]

\[
\sum_{k} \tilde{x}_{(t'' + T_{jk}(\tilde{\omega}))jk} + \sum_{k} \tilde{x}_{(t'' + T_{jk}(\omega) + \sum_{u} R_{j}^{u}(\tilde{\omega}))jk} \leq \sum_{i} \tilde{x}_{t''}^{ij} \leq 1, \forall j \in V, \forall t''
\]

\[
\tilde{C}_{t''}^{u} \geq 0, \tilde{\xi}_{t''}^{ij} \geq 0, \tilde{\Delta}_{t''}^{ij} \geq 0, \forall (i, j) \in V, \forall t''
\]

Indicates whether power line \( j \) is still in outage

Computes required travel and repair time by going from node \( i \) to node \( j \)

Guarantee that the required travel and repair times are met

Guarantees that each arc is visited at most once in one direction

Indicates whether the truck can move to the next node

Domains of the variables
Simulation Policy – Optimal Solution

- Define the graph $G(\mathcal{V}, \mathcal{E})$ with connection costs $T_{ij}$.
- $S$: subset of nodes visited by the truck.
- $C(S,i)$: customer-outage minute starting from node 1 and ending at node $i$.
- $f(S)$: function returning the number of customers in outage after visiting the nodes of $S$.
- The cost of moving from node $i$ to node $j$ is
  \[ C(S,j) = C(S - \{j\}, i) + f(S - \{j\}) \cdot (T_{ij} + R_j) \]

Dynamic Program

For all $j \in \mathcal{V}$ do
  \[ C(\{1, j\}, j) = \sum_i n_i (T_{1j} + R_j) \]

For $s = 3$ to $n$
  For all Subsets $S$ of $\mathcal{V}$ of size $s$
    For all $j \in S, j \neq 1$
      \[ C(S, j) = \min_{i \in S, i \neq j} C(S - \{j\}, i) + f(S - \{j\}) \cdot (T_{ij} + R_j) \]

\[ opt = \min_j C(\mathcal{V}, j) \]

Computational Complexity
\[ O(n^2 2^n) \]
Figure 10: Average customer outage-hours vs. MCTS budget for ten networks.
Comparison to Industrial Heuristics

Graph 1: Customer Outage-Hours vs. Calling Probability
- X-axis: Calling Probability (p)
- Y-axis: Customer Outage-Hours
- Bars indicate the comparison between Escalation Algorithm and Lookahead Policy for different values of p (0.01, 0.10, 1.00).

Graph 2: No. of Unrepaired Faults vs. Calling Probability
- X-axis: Calling Probability (p)
- Y-axis: No. of Unrepaired Faults
- Bars show the comparison between Escalation Algorithm and Lookahead Policy for different values of p (0.01, 0.10, 1.00).
Outline

- Problem Description

- Probability Model for Grid Faults

- Sequential Stochastic Optimization Model for Utility Crew Routing
  - Multistage Lookahead Policy (Monte Carlo Tree Search)
  - Monte Carlo Tree Search with Information Relaxation Dual Bounds

- Stochastic Optimization Model for Sensor and Protective Device Placements
Optimal Protective Device and Sensor Placement

- Protective Device vs Sensor Placement to minimize Customer-Outage Minute:
  - A protective device prevents power flow to the downstream segments when a fault is detected.
  - A sensor feeds back information to the utility center whether power is on or off.

- Optimization problem for optimal protective device and sensor placements:

\[
\min_a \mathbb{E}^\pi \left[ \sum_{t=0}^{T} C(S_t(a), X^\pi(S_t(a))) | S_0(a) \right]
\]

where

\[
\pi = \text{MCTS}
\]

\[
S_{t+1}(a) = S^M(S_t(a), X^\pi(S_t(a)), W_{t+1}(a))
\]

\[
\|a\|_1 \leq M
\]

- \(a\): a binary vector; \(a_i = 1\) if a protective device is placed at node \(i\) and 0 otherwise.
- \(\binom{N}{M}\) combinations of protective devices which results in high computation complexity.
- Solved sequentially by placing one protective device at a time.
Optimal Protective Device and Sensor Placement
Case study A: sensor vs. protective device

Customer Outage-Minute Reduction:
Value of sensor = 0
Value of protective device = 225
Case study B: sensor vs. protective device

Customer Outage-Minute Reduction:
Value of sensor = 3360
Value of protective device = 0
Conclusions

- Proposed first vehicle routing problem with information collection for modelling a belief state that is included in deciding the next route of the truck.

- Proposed a lookahead policy based on Monte Carlo Tree Search (MCTS).

- Used the developed policy to assess the optimal locations of protective devices and sensors.

  - Optimal protective device placement depends on probability of faults, segments length, number of customers across a segment, number of isolated customers.
  
  - Optimal sensor placement depends on probability of faults, number of customers.

Thank you for your attention!