LA-UR-10-07866

*Approved for public release; distribution is unlimited.*

*Title:* Strategic Stockpiling of Power System Supplies for Disaster Recovery

*Author(s):* Carleton Coffrin
Pascal van Hentenryck
Russell Bent

*Intended for:* Submission to the Power Engineering Society Conference

# Strategic Stockpiling of Power System Supplies for Disaster Recovery

Carleton Coffrin, Pascal Van Hentenryck, and Russell Bent

***Abstract*—This paper studies the Power System Stochastic Storage Problem (PSSSP), a novel application in power restoration which consists of deciding how to store power system components throughout a populated area to maximize the amount of power served after disaster restoration. The paper proposes an exact mixed-integer formulation for the linearized DC power flow model and a general column-generation approach. Both formulations were evaluated experimentally on real-life benchmarks. The results show that the column-generation algorithm produces near-optimal solutions quickly and produces orders of magnitude speedups over the exact formulation for large benchmarks. Moreover, both the exact and the column-generation formulations produce significant improvements over a greedy approach and hence should yield significant benefits in practice.**

## I. Background & Motivation

EVERY year, seasonal hurricanes threaten coastal areas. The severity of hurricane damage varies from year to year, but hurricanes often cause power outages that have considerable impacts on both quality of life (e.g., crippled medical services) and economic welfare. Therefore considerable human and monetary resources are always spent to prepare for, and recover from, threatening disasters. At this time, policy makers work together with power system engineers to make the critical decisions relating to how money and resources are allocated for preparation and recovery of the power system. Unfortunately, due to the complex nature of electrical power networks, these preparation and recovery plans are limited by the expertise and intuition of power engineers.

This paper is part of a larger effort to combine mathematical optimization and disaster-specific predictions in order to plan and react to disasters more effectively. The overall goal is to exploit the high-quality predictions (e.g., ensembles of possible hurricane tracks) produced by, say, the National Hurricane Center (NHC) of the National Weather Service and optimization techniques to produce, in reasonable time, robust planning and response plans that significant outperform the practice in the field.

In particular, this paper considers a strategic planning problem arising in this process: *How to store power system repair components throughout a populated area to maximize the amount of power served after disaster restoration.* This Power System Stochastic Storage Problem (PSSSP) raises some fundamental computational issues as it involves, not only discrete storage decisions in a stochastic setting, but also the modeling of the electrical power network, which is a complex physical system. In particular, simply evaluating the benefits of a specific storage configuration for a single scenario requires the solving of a complex optimization model quite similar to optimal transmission switching models (e.g., [1]).

The main contributions of this paper is to present an exact mixed-integer programming (MIP) formulation to the PSSSP (assuming a linearized DC power flow model) and a column-generation algorithm that produces near-optimal solutions under tight time constraints. The column-generation algorithm is an iterative process which alternates between generating storage configurations for each scenario independently (the subproblem) and selecting the best configurations for each scenario globally (the Master problem). The two formulations were evaluated on benchmarks produced by the Los Alamos National Laboratory, using the United States infrastructure and disaster scenarios generated by state-of-the-art hurricane simulation tools similar to those used by the National Hurricane Center. Experimental results indicates that both formulations provide significant benefits in recovering from disasters over greedy approaches (which should already improve over existing practice in the field). Moreover, the column-generation algorithm is shown to scale well to large-scale disasters, producing orders of magnitude improvements over the exact MIP approach. Note also that our column-generation algorithm is independent of any specific electrical power simulation tool. This is important since the electrical power industry has developed several tools for modeling the behavior (e.g. T2000, PSLF, Powerworld, PSS) and recognizes that there is not a single model for completely understanding the behavior of the electrical power network.

The rest of the paper is organized as follows. Section II positions this research with respect to related work in system recovery. Section III presents a specification of the PSSSP problem. Section IV presents an exact MIP formulation using a linearized DC power flow model. Section V presents the column-generation algorithm for solving PSSSPs. Section VI presents greedy algorithms for PSSSPs aimed at modeling current practice in the field. Section VII reports experimental results of the algorithms on some benchmark instances to validate the approach and Section VIII concludes the paper. Note that, although the focus is on hurricanes, the techniques are largely disaster independent and also applies to earthquakes.

## II. Previous Work

Power engineers have been studying power system restoration (PSR) for at least 30 years (see [2] for a comprehensive

C. Coffrin and P. Van Hentenryck are with the Department of Computer Science, Brown University, Providence RI 02912, USA.

R. Bent is with Los Alamos National Laboratory, Los Alamos NM 87545.

collection of work). The goal of PSR research is to find fast and reliable ways to restore a power system to its normal operational state after a blackout event. PSR research has considered not only *steady-state* behavior, in which the flow of electricity is modeled by physical laws, but also *dynamic* behavior which considers transient states occurring during the process of modifying the power system state (e.g., when energizing components). Indeed, these short, but extreme, states may cause unexpected failures which must also be considered carefully [3]. Moreover, power systems are comprised of many different components (e.g., generators, transformers, and capacitors) which have some flexibility in their operational parameters but may be constrained arbitrarily. For example, generators often have a set of discrete generation levels and transformers have a continuous but narrow range of tap ratios. Restoration algorithms must take these into account.

The PSR research community has recognized that global optimization is often impractical for such complex non-linear systems and adopted two main solutions strategies. The first strategy is to use domain-expert knowledge (i.e., the power engineer intuition) to guide an incomplete search of the solution space. These incomplete search methods include *knowledge-based and expert systems* [4], [5], [6], [7] and *local search* [8], [9]. The second strategy is to approximate the power system with a linear model and to try solving the approximate problem optimally [10], [1], [11]. Some work also hybridize both strategies by designing expert systems that solve a series of approximate problems optimally [12], [13]. It is important to emphasize, however, that most PSR work assumes that all network components are operational and need to be reactivated (e.g., [3], [6]). The PSR focus is thus to determine the best order of restoration and the best reconfiguration of the system components.

This paper considers the *stochastic stockpiling of replacement parts in disaster planning*, i.e., How to stockpile power-system components in order to restore as much power as possible after a disaster. The stochastic stockpiling problem is only concerned with the steady-state behavior of the network, but it introduces both a stochastic element and an inventory component to the restoration of a power system. To the best of our knowledge, this is the first PSR application to consider strategic stockpiling decisions for a collection of disaster predictions.

## III. Power System Stochastic Storage

The PSSSP consists of choosing which power system repair components (e.g., components for repairing lines, generators, capacitors, and transformers) to stockpile before a disaster strikes and how those components are allocated to repair the disaster damages. A disaster is specified as a set of scenarios, each of which characterized by a probability and a set of damaged components. In practice, the repository storage constraints may preclude a full restoration of the electrical system after a disaster. Therefore, the goal is to select the items to be restored in order to maximize the amount of power served in each disaster scenario prior to external resources being brought in. The primary outputs of a PSSSPs are:

**Given:**
- Power Network: $\mathcal{P}$
- Network Item: $i \in N$
  - Item Type: $t_i$
- Component Types: $t \in T$
  - Volume: $v_t$
- Storage Locations: $l \in L$
  - Capacity: $c_l$
- Scenario Data: $s \in S$
  - Scenario Probability: $p_s$
  - Damaged Items: $D_s \subseteq N$

**Maximize:**

$$\sum_{s \in S} p_s * \mathit{flow}_s$$

**Subject To:**
- Storage Capacity

**Output:**
- Which components to store at each location.
- Which items to repair in each scenario.

Fig. 1. The Power System Stochastic Storage Problem Specification.

1) The amount of components to stockpile before a disaster (first-stage variables);
2) For each scenario, which network items to repair (second-stage variables).

Figure 1 summarizes the entire problem, which we now describe in detail. The power network is represented as a graph $\mathcal{P} = \langle V, E \rangle$ where $V$ is a set of the network nodes (i.e., buses, generators, and loads) and $E$ is a set of the network edges (i.e., lines and transformers). Any of these items may be damaged as a result of a disaster and hence we group them together as a set $N = V \cup E$ of network items. Each network item $i$ has a type $t_i$ that indicates which component (e.g. parts for lines, generators, or buses) is required to repair this item. Each component type $t \in T$ in the network has a storage volume $v_t$. The components can be stored in a set $L$ of warehouses, each warehouse $l \in L$ being specified by a storage capacity $c_l$. The disaster damages are specified by a set $S$ of different disaster scenarios, each scenario $s$ having a probability $p_s$ of occurring. Each scenario $s$ also specifies its set $D_s \subseteq N$ of damaged items. The objective of the PSSSP is to maximize the expected power flow over all the damage scenarios, where $\mathit{flow}_s$ denotes the power flow in scenario $s$. In PSSSPs, the power flow is defined to be the amount of watts reaching the load points. The algorithms assume that the power flow for a network $\mathcal{P}$ and a set of damaged items $D_s$ can be calculated by some simulation or optimization model. Note also that the decisions on where to store the components can be tackled in a second step. Once storage quantities are determined the location aspect of the problem becomes deterministic and can be modeled as a multi-dimensional knapsack [14].

## IV. The Linear Power Model Approach

PSSSPs can be modeled as two-stage stochastic mixed-integer programming model provided that the power flow constraints for each scenario can be expressed as linear constraints. The linearized DC model, suitably enhanced to capture that some items may be switched on or off, is a natural choice in this setting, since it was found to be reasonably successful in optimal transmission switching [1] and network interdiction [11]. Figure 2 presents a two-stage stochastic mixed-integer programming model for solving PSSSPs optimally (when using a linearized DC power model). The first-stage decision variable $x_t$ denotes the number of stockpiled components of type $t$. Each second-stage is associated with a scenario $s$. Variable $y_{is}$ specifies whether item $i$ is working, while $z_{is}$ specifies whether item $i$ is operational. Auxiliary variable $flow_s$ denotes the power flow for scenario $s$, $P^l_{is}$ the real power flow of line $i$, $P^v_{is}$ the real power flow of node $i$, and $\theta_{is}$ the phase angle of bus $i$. The objective function (1) maximizes the expected power flow across all the disaster scenarios. Constraint (2) ensures that the stockpiled components do not exceed the storage capacity. Each scenario can only repair damaged items using the stockpiled components. Constraint (3) ensures that each scenario $s$ uses no more than the stockpiled components of type $t$. There may be more damaged items of a certain type than the number of stockpiled component of that type and the optimization model needs to choose which ones to repair, if any. This is captured, for each scenario $s$, using a linearized DC power flow model (4–13), which extends optimal transmission switching [1] to buses, generators, and loads since they may be damaged in a disaster. Moreover, since we are interested in a best-case power flow analysis, we assume that generation and load can be dispatched and shed continuously. Constraints (4–7) capture the operational state of the network and specify whether item $i$ is working and/or operational in scenario $s$. An item is operational only if all buses it is connected to are also operational. Constraint (4) specifies that all undamaged nodes and lines are working. Constraint (5–7) specify which buses (5), load and generators (6), and lines (7) are operational. Constraint (8) computes the total power flow of scenario $s$ in terms of variables $P^v_{is}$. The conservation of energy is modeled in constraint (9). Constraint (10–11) specify the bounds on the power produced/consumed/transmitted by generators, loads, and lines. Observe that, when these items are not operational, no power is consumed, produced, or transmitted. When a line is non-operational, the effects of Kirchhoff's laws must be ignored, which is captured in the traditional power equations through a big $M$ transformation (12–13). In this setting, $M$ can be chosen as $B_i * \frac{\pi}{3}$. Note also that the logical constraints can be easily linearized in terms of the 0/1 variables.

This MIP approach is appealing since it solves PSSSPs optimally for a linearized DC model of power flow. In particular, it provides a sound basis to compare other approaches. However, it does not scale smoothly with the size of the disasters and may be prohibitive computationally in real-life situations. To remedy this limitation, the rest of this paper studies a column-generation approach.

**Let:**

$D_{ts} = \{i \in D_s : t_i = t\}$
$V^b = \{i \in N : t_i = bus\}$ - the set of network busses
$V^g_i$ = the set of generators connected to bus $i$
$V^l_i$ = the set of loads connected to bus $i$
$L$ = the set of network lines
$L^-_i$ = the from bus of line $i$
$L^+_i$ = the to bus of line $i$
$LO_b$ = the set of exiting lines from bus $b$
$LI_b$ = the set of entering lines from bus $b$
$B_i$ = susceptance of line $i$
$\hat{P}^l_i$ = transmission capacity of line $i$
$\hat{P}^v_i$ = maximum capacity or load of node $i$

**Variables:**

$x_t \in \mathcal{N}$ - number of stockpiled items of type $t$
$y_{is} \in \{0,1\}$ - item $i$ is working in scenario $s$
$z_{is} \in \{0,1\}$ - item $i$ is operational in scenario $s$
$flow_s \in \mathcal{R}^+$ - served power for scenario $s$
$P^l_{is} \in (-\hat{P}^l_i, \hat{P}^l_i)$ - power flow on line $i$
$P^v_{is} \in (0, \hat{P}^v_i)$ - power flow on node $i$
$\theta_{is} \in (-\frac{\pi}{6}, \frac{\pi}{6})$ - phase angle on bus $i$

**Maximize:**

$$\sum_{s \in S} p_s * flow_s \tag{1}$$

**Subject To:**

$$\sum_{t \in T} v_t * x_t \le \sum_l c_l \tag{2}$$
$$\sum_{i \in D_{ts}} y_{is} \le x_t \quad \forall t \in T, s \in S \tag{3}$$
$$y_{is} = 1 \quad \forall i \notin D_s \quad \forall s \in S \tag{4}$$
$$z_{is} = y_{is} \quad \forall i \in V^b \quad \forall s \in S \tag{5}$$
$$z_{is} = y_{is} \wedge y_{js} \quad \forall j \in V^b, \forall i \in V^g_j \cup V^l_j, s \in S \tag{6}$$
$$z_{is} = y_{is} \wedge y_{L^+_i s} \wedge y_{L^-_i s} \quad \forall i \in L, s \in S \tag{7}$$
$$flow_s = \sum_{i \in V^b} \sum_{j \in V^l_i} P^v_{js} \quad \forall s \in S \tag{8}$$
$$\sum_{j \in V^l_i} P^v_{js} = \sum_{j \in V^g_i} P^v_{js} + \sum_{j \in LI_i} P^l_{js} - \sum_{j \in LO_i} P^l_{js} \quad \forall i \in V^b, s \tag{9}$$
$$0 \le P^v_{is} \le \hat{P}^v_i * z_{is} \quad \forall i \in V^g_j \cup V^l_j, s \in S \tag{10}$$
$$-\hat{P}^l_{is} * z_{is} \le P^l_{is} \le \hat{P}^l_{is} * z_{is} \quad \forall i \in L, \forall s \tag{11}$$
$$P^l_{is} \le B_i * (\theta_{L^+_i s} - \theta_{L^-_i s}) + M * (\neg z_{is}) \quad \forall i \in L, s \in S \tag{12}$$
$$P^l_{is} \ge B_i * (\theta_{L^+_i s} - \theta_{L^-_i s}) - M * (\neg z_{is}) \quad \forall i \in L, s \in S \tag{13}$$

Fig. 2. The MIP Model for Power System Stochastic Storage Problems.

## V. A Column-Generation Approach

When the optimal values of the first-stage variables are known, the PSSSP reduces to solving a restoration problem for each scenario $s$, i.e., to maximize the power flow for scenario $s$ under the stored resources specified by the first-stage variables.

The column-generation approach takes the dual approach: It aims at combining feasible solutions of each scenario to obtain high-quality values for the first-stage variables. In this setting, a configuration is a tuple $w = \langle w_1, \ldots, w_k \rangle$ ($T = \{1, \ldots, k\}$) where $w_t$ specifies the number of items of type $t$ being stockpiled. A configuration is feasible if it satisfies the storage constraints, i.e.,

$$\sum_{t \in T} v_t * w_t \le \sum_{l \in L} c_l.$$

For each scenario $s$, the optimal power flow of a feasible configuration $w$ is denoted by $flow_{ws}$. Once a set of feasible configurations is available, a mixed-integer program (the Master problem) selects a set of configurations, one per scenario,

**Let:**
$\mathcal{W}$ - the set of configurations
$w_t$ - the amount of components of type $t$ in $w \in W$
$flow_{ws}$ - the served power for $w$ in scenario $s$
**Variables:**
$x_t \in \mathcal{N}$ - number of stockpiled items of type $t$
$y_{ws} \in \{0,1\}$ - 1 if configuration $w$ is used in $s$
$flow_s \in \mathcal{R}^+$ - power flow for scenario $s$
**Maximize:**

$$\sum_{s \in S} p_s * flow_s \quad (1)$$

**Subject To:**

$$\sum_{t \in T} v_t * x_t \leq \sum_l c_l \quad (2)$$

$$\sum_{w \in W} w_t * y_{ws} \leq x_t \ \forall s, t \quad (3)$$

$$\sum_{w \in W} y_{ws} = 1 \ \forall s \quad (4)$$

$$\sum_{w \in W} flow_{ws} * y_{ws} = flow_s \ \forall s \quad (5)$$

Fig. 3. The The Master Problem for the PSSSP.

maximizing the expected power flow across all scenarios. Figure 3 presents the MIP model. In the model, the objective (1) specifies that the goal is to maximize the expected flow. Constraint (2) enforces the storage requirements, while constraint (3) links the number of components $x_t$ of type $t$ used by scenario $s$ with the configuration variables $y_{ws}$. Constraint (4) specifies that each scenario uses exactly one configuration. Constraint (5) computes the power flow of scenario $s$ using the configuration variables $y_{ws}$.

It is obviously impractical to generate all configurations for all scenarios. As a result, we follow a column-generation approach in which configurations are generated on demand to improve the objective of the Master problem. Recall that our goal is twofold: First, we aim at designing an approach which is largely independent of the power flow simulation or optimization model. Second, we aim at producing an optimization model which scales to large disasters. For these reasons, we generate configurations based on techniques inspired by online stochastic combinatorial optimization [15], [16].

### A. The Column-Generation Subproblem

In our column-generation approach, the configurations are generated for each scenario independently and in a systematic fashion. Figure 4 describes a generic optimization model for generating a configuration. In the model, the storage and power flow constraints are abstracted for generality. The storage constraints contain at least the storage constraints of the PSSSP but generally adds additional constraints to generate "interesting" configurations for the Master problem. The power flow constraints are abstracted to make the approach independent of the power flow model. The optimization model features a lexicographic objective (1), seeking first to maximize the power flow and then to minimize the number of repaired components. The storage constraints (2) are expressed in terms

**Let:**
$s$ - a scenario
$T_t = \{i \in N : t_i = t\}$ - the set of nodes of type $t$
**Variables:**
$r_i \in \{0,1\}$ - 1 if item $i$ is repaired
$flow \in \mathcal{R}^+$ - served power
**Maximize:**

$$(flow, -\textstyle\sum_i r_i) \quad (1)$$

**Subject To:**

$$storageConstraint(\{T_t\}_{t \in T}, \{r_i\}_{i \in N}) \quad (2)$$

$$flowConstraints(\mathcal{P}, D_s, \{r_i\}_{i \in N}, flow) \quad (3)$$

Fig. 4. The Generic Configuration-Generation Model For a Scenario.

of the decision variables $r_i$ that indicates whether item $i$ is repaired: They will be described shortly. Constraint (3) encapsulates the power flow model which computes the flow from the values of the decision variables.

The column-generation algorithm generates two fundamental types of configurations for a given scenario. Each type takes into account information from the other scenarios and instantiates the generic model in Figure 4 with specific storage constraints.

*a) Upward Configurations:* The intuition behind upward configurations is as follows. Some storage decisions coming from other scenarios are fixed by a configuration $w$ and the goal is to generate as best a configuration as possible for scenario $s$ given these decisions, i.e., a configuration for scenario $s$ that maximizes its power flow given the fact that the repairs must include at least $w_t$ components of type $t$. More precisely, *upwardConfiguration(w,s)* denotes the solution to the model in Figure 4 where the storage constraint becomes

$$\sum_{t \in T} v_t * \max(w_t, \sum_{i \in T_t} r_i) \leq \sum_l c_l.$$

*b) Downward Configurations:* The intuition behind downward configurations is as follows. Our goal is to give an opportunity for a configuration $w$ coming from another scenario to be selected in the Master problem, while ensuring that all other scenarios generate a mutually compatible configuration. As a result, we seek to maximize the power flow for scenario $s$, while not exceeding the storage requirements of $w$. More precisely, *downwardConfiguration(w,s)* denotes the solution to the model in Figure 4 where the storage constraint becomes

$$\forall t \in T : \sum_{i \in T_t} r_i \leq w_t.$$

Two special cases of upward and downward configurations are important for initializing the column-generation process: the clairvoyant and the no-repair configurations.

*c) Clairvoyant Configurations:* The clairvoyant configuration for a scenario $s$ is simply the upward configuration with no storage constraint imposed:

$$clairvoyant(s) = upwardConfiguration(\langle 0, \ldots, 0 \rangle, s).$$

Scenarios for which their clairvoyant configuration is selected in the Master problem cannot be improved.

*d) No-Repair Configurations:* The no-repair configuration for a scenario $s$ is simply the downward configuration with no storage availability, i.e.,

$$\textit{no-repair}(s) = \textit{downwardConfiguration}(\langle 0, \ldots, 0\rangle, s).$$

The no-repair configuration guarantees that each scenario can select at least one configuration in the Master problem.

### B. The Column-Generation Algorithm

Figure 5 presents the complete column-generation algorithm. Lines 1–4 describe the initialization process, while lines 7–12 specify how to generate new configurations. The overall algorithm terminates when the newly generated configurations do not improve the quality of the Master problem.

*e) Initialization:* The initialization step generates the initial set of configurations. These contain the clairvoyant and the no-repair configurations for each scenario, as well as the downward configurations obtained from the expected clairvoyant, i.e., the configuration $w^e$ whose element $t$ is defined by

$$w^e_t = \sum_{s \in S} p_s * \textit{clairvoyant}(s).$$

*f) The Colum-Generation Process:* At each iteration, the algorithm solves the Master problem. If the solution has not improved over the previous iteration, the algorithm completes and returns its configuration, i.e., the values of the decision variables $x_t$ describing how many elements of component type $t$ must be stockpiled.

Otherwise, the algorithm considers each scenario $s$ in isolation. The key idea is to solve the Master problem without scenario $s$, which we call a restricted Master problem, and to derive new configurations from its solution $w^{-s}$. In particular, for each scenario $s$, the column-generation algorithm generates

1) one upward configuration for $s$ (line 9);
2) one downward configuration for every other scenario in $S \setminus \{s\}$ (line 12).

The upward configuration for $s$ is simply the best possible configuration given the decisions in the restricted Master problem, i.e.,

$$\textit{upwardConfiguration}(w^{-s}, s)$$

The downward configurations for the other scenarios are obtained by selecting an existing configuration $w^{+s}$ for $s$ which, if added to the restricted Master solution, would violate the storage constraint: Downward configurations of the form

$$\textit{downwardConfiguration}(w^{+s}, j)$$

are then computed for each scenario $j \in S \setminus \{s\}$ in order to give the Master an opportunity to select $w^{+s}$. The configuration $w^{+s}$ aims at being desirable for $s$, while taking into account the requirement of the other scenarios. Initially, $w^{+s}$ is the clairvoyant solution. In general, $w^{+s}$ is the configuration in $\mathcal{W}$ which, when scaled to satisfy the storage constraints, maximizes the power flow for scenario $s$, i.e.,

$$\begin{array}{ll} \max_{w \in \mathcal{W}} & p * flow_w \\ \text{subject to} & \\ & \exists t \in T : w_t > w^{-s}_t \\ & \forall t \in T : p * w_t \leq w^{-s}_t \\ & 0 \leq p \leq 1 \end{array}$$

```
CONGIGURATIONGENERATION()
 1  W ← {no-repair(s) | s ∈ S}
 2  W ← W ∪ clairvoyant(s) | s ∈ S}
 3  w^e ← expectedClairvoyant(S)
 4  W ← W ∪ downwardConfiguration(w^e, s) | s ∈ S}
 5  while The master objective is increasing
 6  do w^m ← Master(S)
 7      for s ∈ S
 8      do w^{-s} ← Master(S \ {s})
 9          W ← W ∪ upwardConfiguration(w^{-s}, s)
10          w^{+s} ← selectConfiguration(w^{-s}, s)
11          for j ∈ S \ {s}
12          do W ← W ∪ downwardConfiguration(w^{+s}, j)
13  return w^m
```

Fig. 5. The Column-Generation Algorithm for the PSSSP.

*g) Simulation-Independent Optimization:* The column-generation algorithm only relies on the power flow model in the subproblem of Figure 4. This is essentially an optimal transmission switching model with limits on the component types that can be repaired. It can be solved in various ways, making the overall approach independent of the power flow model.

## VI. A Greedy Storage Model

This section presents a basic greedy storage model which emulates standard practice in storage procedures and provides a baseline for evaluating our optimization algorithms. However, to the best of our knowledge, the storage models used in practice are rather ad-hoc or based on past experience and the models presented in this section probably improve existing procedures. In fact, it is well-known that power companies often rely on spare parts from neighboring regions in post-disaster situations.

The greedy storage algorithm generates a storage configuration $w^h$ by computing first a distribution of the component types and then filling the available storage capacity with components to match this distribution. Once a scenario $s$ is revealed, the quality of greedy storage can be evaluated by computing a downward configuration

$$\textit{downwardConfiguration}(w^h, s).$$

The distribution used to produce $w^h$ is based on the number of occurrences of the component types in the undamaged network. This metric is meaningful because it models the assumption that every component type is equally likely to be damaged in a disaster. The computation for a distribution

| Benchmark | $\|N\|$ | $\|S\|$ | $\max_{s\in S}(\|D_s\|)$ |
|---|---|---|---|
| BM1 | 326 | 3 | 22 |
| BM3 | 266 | 18 | 61 |
| BM4 | 326 | 18 | 121 |
| BM5 | 1789 | 4 | 255 |

TABLE I
FEATURES OF THE PSSSP BENCHMARKS.

$Pr$ proceeds as follows. When all of the components have a uniform size $v$, the quantity of component type $i$ is

$$\lfloor(\sum_{l\in L} c_l * Pr(i))/v\rfloor.$$

When the component types have different sizes, the storage configuration is the solution of the optimization problem

$$\begin{aligned} \min_{t\in T} \quad & \left|\frac{w_t^h}{\sum_{i\in T} w_i^h} - Pr(i)\right| \\ \text{subject to} \quad & \\ & \sum_{t\in T} w_t^h * v_t \leq \sum_{l\in L} c_l \\ & 0 \leq w_i^h. \end{aligned}$$

## VII. BENCHMARKS & RESULTS

This section reports the experimental results of the proposed models. It starts by describing the benchmarks and the algorithms. It then presents the quality results and the efficiency results. It concludes by reporting a variety of statistics on the behavior of the column-generation algorithms.

### A. Benchmarks

The benchmarks were produced by Los Alamos National Laboratory and are based on the electrical infrastructure of the United States. The disaster scenarios were generated by state-of-the-art hurricane simulation tools similar to those used by the National Hurricane Center [17], [18]. Their sizes are presented in Table I which gives the number of network items, the number of scenarios, and the size of the largest damage produced by the scenarios. For instance, BM5 considers a network with 1789 items and 4 disaster scenarios. The worst disaster damages 255 items in the network. Each of these benchmarks is evaluated for a large variety of storage capacities (i.e., different values of $c_l$). This makes it possible to evaluate the behavior of the algorithms under a wide variety of circumstances, as well as to study the tradeoff between storage (or budget) and the quality of the restoration. Overall, more than 70 configurations of each algorithm were studied in these experiments.

### B. Implementation of the Algorithms

The optimization algorithms were implemented in the COMET system [19], [20], [21] and the experiments were run on Intel Xeon CPU 2.80GHz machines running 64-bit Linux Debian. The experiments use the standard linearized DC power flow equations for the subproblems. To quantify its benefits compared to the practice in the field, the column-generation algorithm is compared to the greedy storage procedures presented earlier. The global quality of the column-generation algorithm is evaluated by comparison to the optimal MIP model presented in Section IV and to the clairvoyant solution, i.e., the solution obtained when the actual disaster scenario is known in advance. Benchmarks 1 and 3 are small enough that the MIP models can be solved to optimality. However, benchmarks 4 and 5 present large damage scenarios with over 100 damaged components and optimal solutions to the MIP models may not be obtained in a reasonable amount of time. The following time limits are then imposed: 18 hours of the global MIP, 2 hours for the evaluation of each greedy and clairvoyant scenario, and 15 minutes for column-generation sub-problems.

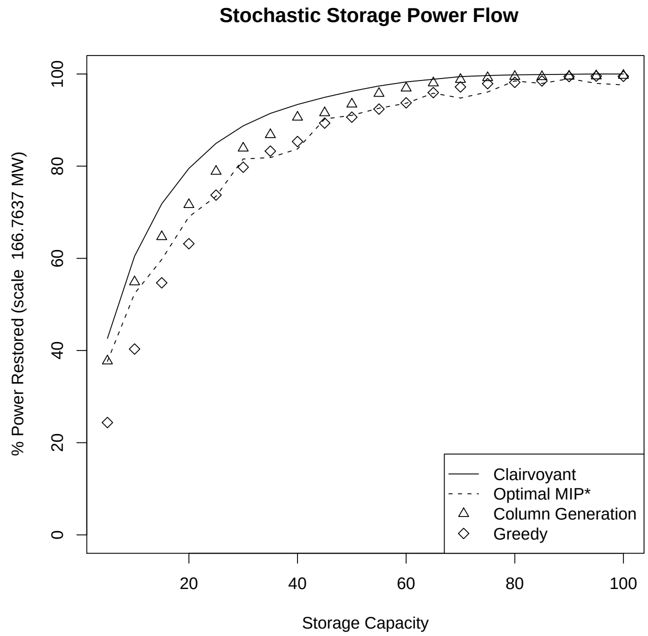


Fig. 6. Solution Quality for the PSSSP Algorithms on Benchmark 4.

### C. Quality of the Algorithms

Figure 6 depicts the quality results for Benchmark 4, i.e., the percentage of the backout restored by the various algorithms for a given storage capacity. The figure shows the results for the greedy procedure, the column-generation algorithm, the MIP model, and the expected clairvoyant solution.

The results indicate that the MIP model and the column-generation algorithm produce significant improvements over the greedy approach, reducing the blackout by almost 20% in the best case. The improvements are especially significant for low storage capacities, which seems to be the case in practice as mentioned earlier. Remarkably, the column-generation algorithm outperforms the quality of the MIP model, which always reaches its time limit. On this set of benchmarks, the MIP model and the column-generation algorithm are close to the clairvoyant solution, indicating that the stochasticity is reasonable in PSSSPs. This implies that our algorithms produces high-quality solutions for the scenarios.

Space constraints prevent us from including similar graphs for all benchmark classes. Instead, we present an aggregation of these results for each benchmark class, averaging the quality

| Benchmark | Clairvoyant | MIP Model | Column Generation | Greedy |
|---|---|---|---|---|
| BM1 | 100% | 99.1% | 98.5% | 73.4% |
| BM3 | 100% | 97.4% | 96.6% | 71.8% |
| BM4 | 100% | 91.4% | 97.5% | 88.3% |
| BM5 | 100% | 77.5% | 85.4% | 69.3% |

TABLE II
EXPERIMENTAL RESULTS ON AGGREGATED SOLUTION QUALITY.

| Benchmark | Clairvoyant | MIP Model | Column Generation | Greedy |
|---|---|---|---|---|
| BM1 | 100% | 97.9% | 96.3% | 66.0% |
| BM3 | 100% | 92.8% | 92.8% | 21.9% |
| BM4 | 100% | 62.7% | 75.0% | 34.2% |
| BM5 | 100% | 70.7% | 81.5% | 11.4% |

TABLE III
AGGREGATED SOLUTION QUALITY ON SMALL STORAGE CAPCITIES.

over 20 storage capacities for each benchmark. The results are presented in Table II, which depicts the average relative gap of each algorithm from the clairvoyant solution. The column-generation approach brings substantial benefits over the greedy approach in smaller benchmarks and over both the greedy and MIP approach on larger ones, demonstrating its scalability. Table III gives another perspective as it only aggregates results for small storage capacities which often correspond to real-life situations. The benefits of the column-generation approach over the greedy algorithm are dramatic and the gap over the MIP is even more significant on the larger benchmarks.

### D. Performance of the Algorithms

Figure 7 depicts the performance results for Benchmark 4 under 20 different storage capacities. It depicts the runtime of the MIP Model and the column-generation algorithm in seconds, using a log scale given the significant performance difference between the algorithms. The results show substantial improvements in performance for the column-generation algorithm over the MIP model. This is especially the case for low to medium storage capacities, which typically model the reality in the field. For these storage configurations, the column-generation algorithm runs from 2 to 100 times faster than the MIP model (despite its time limit) and is about 3 times faster in average. In summary, on large instances, the column-generation approach improves both the quality and performance of the MIP models. Table IV presents the average runtime results for all benchmarks. It also gives the time to evaluate the greedy and expected clairvoyant problems as a basis for comparison. These results are quite interesting in the sense that the MIP model behaves very well for reasonably small disasters, but does not scale well for larger ones. Since our goal is to tackle disaster planning and restoration at the state level, they point out to a fundamental limitation of a pure MIP approach. It is also interesting to observe that the column-generation algorithm takes about 4 times as much as time as evaluating the greedy power flow. This is quite remarkable and indicates that the column-generation is likely to scale well to even larger problem sizes. Overall, these results indicate that the column-generation algorithm produces near-optimal solutions under tight time constraints and represents an appealling approach to PSSSPs.

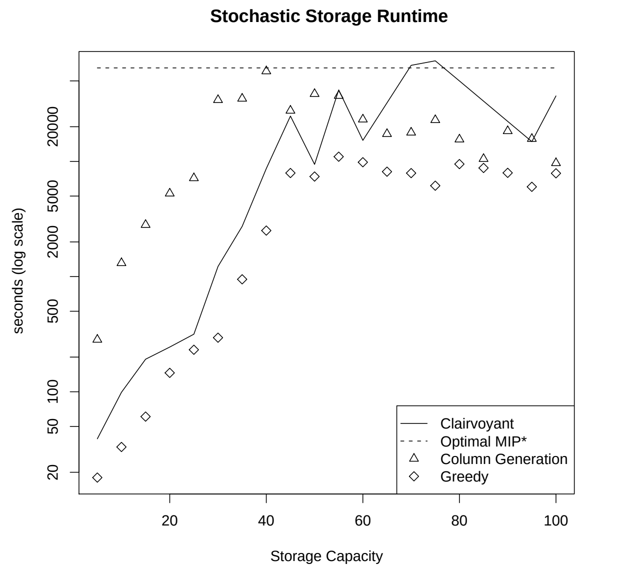


Fig. 7. Experimental Results for the PSSSP Algorithm on Benchmark 4.

| Benchmark | Clairvoyant | MIP Model | Column Generation | Greedy |
|---|---|---|---|---|
| BM1 | 2 | 6 | 8.84 | 2 |
| BM3 | 18 | 21351 | 415 | 18 |
| BM4 | 20551 | 64807 | 20038 | 5133 |
| BM5 | 20544 | 63704 | 21349 | 8308 |

TABLE IV
PSSSP BENCHMARK RUNTIME (SECONDS)

### E. Behavior of the Column-Generation Algorithm

It is useful to study the behavior of the column-generation algorithm in order to understand its performance. Table VI reports, in average, the number of configurations generated, as well as the number of downward and upward configurations in the final solution. The results indicate that the algorithm only needs a small number of configurations to produce a high-quality solution. Recall that there are 18 scenarios in BM3 and BM4, which means that the algorithm generates in average less than 7 configurations per scenario. The types of configurations are relatively well balanced, although there are clearly more upward configurations in average. This is partly caused by the variance in the scenario damages. As the storage capacity increase more of the small damage scenarios can select their clairvoyant solution. Figure 8 details these results for BM4.

## VIII. CONCLUSION

This paper studied a novel problem in power system restoration: the Power System Stochastic Storage Problem (PSSSP). The objective in PSSSPs is to decide how to stockpile components in order to recover from blackouts as best as possible after a disaster. PSSSPs are complex stochastic optimization problems, combining power flow simulators, discrete storage decisions, discrete repair decisions given the storage decisions, and a collection of scenarios describing the potential effects of the disaster. The paper proposed an exact mixed-integer formulation and a column-generation approach. The column-generation subproblem generates configurations from each scenario independently, taking into account storage decisions

| Benchmark | Clairvoyant | MIP Model | Column Generation | Greedy |
|---|---|---|---|---|
| BM1 | 2 | 4 | 10 | 2 |
| BM3 | 12 | 13060 | 179 | 12 |
| BM4 | 178 | 64807 | 3366 | 98 |
| BM5 | 14613 | 60502 | 10371 | 1123 |

TABLE V
BENCHMARK RUNTIME ON SMALL STORAGE CAPACITIES (SECONDS)

| Benchmark | Configurations | Downward Config. | Upward Config. |
|---|---|---|---|
| BM1 | 7.8 | 0.9 | 2.1 |
| BM3 | 115.5 | 8.9 | 9.1 |
| BM4 | 136.50 | 7.75 | 10.25 |
| BM5 | 30.42 | 1.89 | 2.11 |

TABLE VI
THE BEHAVIOR OF THE COLUMN GENERATION FOR PSSSPS.

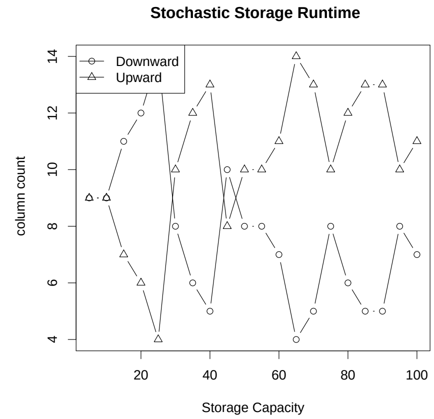


Fig. 8. The Types of Configurations in the Final PSSSP Solutions.

for other scenarios. A subset of these storage configurations are then selected in the Master problem, to produce the global storage decisions. The algorithms were evaluated on benchmarks produced by the Los Alamos National Laboratory, using the electrical power infrastructure of the United States. The disaster scenarios were generated by state-of-the-art hurricane simulation tools similar to those used by the National Hurricane Center. Experimental results show that the column-generation algorithm produces near-optimal solutions and produces orders of magnitude speedups over the exact formulation for large benchmarks. Moreover, both the exact and the column-generation formulations produce significant improvements over greedy approaches and hence should yield significant benefits in practice. The results also seem to indicate that the column-generation algorithm should nicely scale to even larger disasters, given the small number of configurations necessary to reach a near-optimal solution. As a result, the column-generation algorithm should provide a practical tool for decision makers in the strategic planning phase just before a disaster strikes.

## REFERENCES

[1] E. Fisher, R. O'Neill, and M. Ferris, "Optimal transmission switching," *Power Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 1346 –1355, aug. 2008.
[2] M. Adibi, *Power System Restoration(Methodologies & Implementation Strategies)*, 2000.
[3] M. Adibi and L. Fink, "Power system restoration planning," *Power Systems, IEEE Transactions on*, vol. 9, no. 1, pp. 22 –28, feb. 1994.
[4] T. Sakaguchi and K. Matsumoto, "Development of a knowledge based system for power system restoration," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-102, no. 2, pp. 320 –329, feb. 1983.
[5] J. Huang, F. Galiana, and G. Vuong, "Power system restoration incorporating interactive graphics and optimization," may. 1991, pp. 216 –222.
[6] M. Adibi, L. Kafka, and D. Milanicz, "Expert system requirements for power system restoration," *Power Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 1592 –1600, aug. 1994.
[7] J. Ancona, "A framework for power system restoration following a major power failure," *Power Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 1480 –1485, aug. 1995.
[8] A. Morelato and A. Monticelli, "Heuristic search approach to distribution system restoration," *Power Delivery, IEEE Transactions on*, vol. 4, no. 4, pp. 2235 –2241, oct. 1989.
[9] H. Mori and Y. Ogita, "A parallel tabu search based approach to optimal network reconfigurations for service restoration in distribution systems," vol. 2, 2002, pp. 814 – 819 vol.2.
[10] M. Yolcu, Z. Zabar, L. Birenbaum, and S. Granek, "Adaptation of the simplex algorithm to modeling of cold load pickup of a large secondary network distribution system," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-102, no. 7, pp. 2064 –2068, jul. 1983.
[11] A. Delgadillo, J. Arroyo, and N. Alguacil, "Analysis of electric grid interdiction with line switching," *Power Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 633 –641, may. 2010.
[12] T. Nagata, H. Sasaki, and R. Yokoyama, "Power system restoration by joint usage of expert system and mathematical programming approach," *Power Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 1473 –1479, aug. 1995.
[13] J. Huang, L. Audette, and S. Harrison, "A systematic method for power system restoration planning," *Power Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 869 –875, may. 1995.
[14] J. Puchinger, G. Raidl, and U. Pferschy, "The Multidimensional Knapsack Problem: Structure and Algorithms," *INFORMS Journal on Computing*, vol. 22, no. 2, pp. 250–265, 2010.
[15] P. Van Hentenryck and R. Bent, *Online Stochastic Combinatorial Optimization*. Cambridge, Mass.: The MIT Press, 2006.
[16] P. Van Hentenryck, R. Bent, and E. Upfal, "Online Stochastic Optimization under Time Constraints," *Annals of Operations Research (Special Issue on Stochastic Programming)*, vol. 177, pp. 151–183, 2009.
[17] "Fema hazus overview," http://www.fema.gov/plan/prevent/hazus/, 2010.
[18] D. A. Reed, "Electric utility distribution analysis for extreme winds," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 96, no. 1, pp. 123–140, 2008.
[19] "Comet 2.1 user manual, dynadec inc." http://dynadec.com/, 2009.
[20] L. Michel, A. See, and P. Van Hentenryck, "Transparent Parallelization of Constraint Programming," *INFORMS Journal on Computing*, vol. 21, no. 3, pp. 363–382, 2009.
[21] P. Van Hentenryck, *Constraint-Based Local Search*. Cambridge, Mass.: The MIT Press, 2005.

**Carleton Coffrin** received a B.Sc. in Computer Science and a B.F.A. in Theatrical Design from the University of Connecticut, Storrs, CT and a M.S. from Brown University, Providence, RI. He is currently a Ph.D. candidate in the department of computer science at Brown University.

**Pascal Van Hentenryck** is a Professor of Computer Science at Brown University. He is a fellow of the Association for the Advancement of Artificial Intelligence, the recipent of the 2002 ICS INFORMS award, the 2006 ACP Award, a honorary degree from the University of Louvain, and the Philip J. Bray award for teaching excellence. He is the author of five MIT Press books and numerous publications in the premier journals in artificial intelligence, numerical analysis, operations research, and programming languages. Most of this research in optimization software systems has been commercialized and is widely used in academia and industry.

**Russell Bent** received his PhD in Computer Science from Brown University in 2005 and is currently a staff scientist at Los Alamos National Laboratory in the energy and infrastructure analysis group serving as a capability research and development task leader. Russell is responsible for the development of LogiSims, a project for resource management, planning, and distribution for disasters. He is also co-principal investigator for a project optimization and control theory for Smart Grid technologies. Russell's publications include deterministic optimization, optimization under uncertainty, infrastructure modeling and simulation, constraint programming, algorithms, and simulation. In the last five years, Russell has published 1 book and over 25 articles in peer reviewed journals and conferences in artificial intelligence and operations research.