

Singh, Bismark; Knueven, Bernard

**Article — Published Version**

## Lagrangian relaxation based heuristics for a chance-constrained optimization model of a hybrid solar-battery storage system

Journal of Global Optimization

**Provided in Cooperation with:**

Springer Nature

*Suggested Citation:* Singh, Bismark; Knueven, Bernard (2021) : Lagrangian relaxation based heuristics for a chance-constrained optimization model of a hybrid solar-battery storage system, Journal of Global Optimization, ISSN 1573-2916, Springer US, New York, NY, Vol. 80, Iss. 4, pp. 965-989,  
<https://doi.org/10.1007/s10898-021-01041-y>

This Version is available at:

<https://hdl.handle.net/10419/287241>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>



# Lagrangian relaxation based heuristics for a chance-constrained optimization model of a hybrid solar-battery storage system

Bismark Singh<sup>1,2</sup>  · Bernard Knueven<sup>3</sup>

Received: 14 May 2019 / Accepted: 18 May 2021 / Published online: 1 June 2021  
© The Author(s) 2021

## Abstract

We develop a stochastic optimization model for scheduling a hybrid solar-battery storage system. Solar power in excess of the promise can be used to charge the battery, while power short of the promise is met by discharging the battery. We ensure reliable operations by using a joint chance constraint. Models with a few hundred scenarios are relatively tractable; for larger models, we demonstrate how a Lagrangian relaxation scheme provides improved results. To further accelerate the Lagrangian scheme, we embed the progressive hedging algorithm within the subgradient iterations of the Lagrangian relaxation. We investigate several enhancements of the progressive hedging algorithm, and find bundling of scenarios results in the best bounds. Finally, we provide a generalization for how our analysis extends to a microgrid with multiple batteries and photovoltaic generators.

**Keywords** Chance constraints · Stochastic optimization · Lagrangian decomposition · Progressive hedging · Solar power · Photovoltaic power station · Battery storage · Virtual power plant · Out of sample validation · Microgrid

## 1 Introduction

### 1.1 Motivation

A photovoltaic power station (PPS) is a system consisting of solar panels and inverters to convert light into electricity. Solar power is uncertain, and, of course, not available for all hours. Hence, coupling a sufficiently large storage device with a PPS can significantly increase its economic value by shifting energy to times when it is in higher demand, creating a hybrid

---

✉ Bismark Singh  
bismark.singh@fau.de

<sup>1</sup> Department of Mathematics, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen 91058, Germany

<sup>2</sup> Department of Data Science, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen 91058, Germany

<sup>3</sup> Computational Science Center, National Renewable Energy Laboratory, Golden, CO 80401, USA

energy system; see, e.g., [18] for different combinations of sources forming hybrid energy systems. Such energy storage systems find frequent applications in modern microgrids [5], especially as standalone systems in remote islands [31,38,39,69].

Standalone hybrid PPS systems have also been shown to be economically feasible and viable. Bhuiyan et al. [8] show the economical feasibility of a PPS system in a remote area of Bangladesh. Similarly, a large scale system of 100MW in the Gobi desert is shown to be economically feasible [26]. Ma et al. [39] study a hybrid wind-solar system with storage from a pumped-hydro system on a remote island in Hong Kong. Zhao et al. [69] study a hybrid system with wind, solar, diesel and battery storage on an island in China. Buildings can also be powered efficiently using micro-grids [20,67]. For an introduction to PPS systems and their characteristics, see, e.g., [43,44].

Due to the uncertainty in the availability of renewables, additional complications arise when there is a need to ensure highly reliable operations. To mitigate the effect of the uncertainty in renewables, often two or more sources of energy are coupled, in addition to the storage system. Solar-wind hybrid systems have been extensively studied—optimal sizing [65,70], feasibility of operations [7,39], and ensuring reliable operations [9,70]. In this work, we consider a problem where a standalone PPS, connected to a battery storage, commits (e.g., in forward energy markets) to deliver a fixed supply of energy *reliably* at every hour. Although, we consider a single battery storage with a single (unified) PPS, the approach we propose extends to multiple power sources if they interact externally as a single group. Thus, our system supplies power as a so-called virtual power plant; see, e.g., [33,40]. The renewable source of power we consider is solar power alone, and we supply power into a self scheduling day-ahead market. Our aim is to ensure highly reliable operations, despite the fact that supply decisions are made day-ahead (and, thus, without observing the availability of the uncertain solar power). Ensuring reliability is the primary purpose, while maximizing revenues from supplying power is secondary.

The model we propose outputs a day-ahead promise of hourly power, using a forecast of the hourly day-ahead electricity prices and the hourly day-ahead solar power. We assume the electricity prices are completely known, and we have a sufficiently large number of solar power scenarios. A single solar power scenario consists of a vector of 24 values for the hourly day-ahead solar output; i.e., we assume that once a solar power scenario is realized, the solar power output for the entire day is known. Although having a dynamic hour-by-hour forecast of solar power would result in a higher fidelity model, this assumption significantly simplifies modeling and computations as we describe in Sect. 1.2. Next, at every hour, we dynamically decide whether to charge the battery (if solar power is excess of our promise) or discharge the battery (if solar power is short of our promise). We assume the battery cannot be simultaneously charged and discharged. Similar self-scheduling models have been proposed, although for different energy systems [14,56].

## 1.2 Joint chance constraints

To ensure highly reliable operations in the self-scheduling market, we make use of joint chance constraints (JCCs). The following is an example of a linear joint chance constraint:

$$\mathbb{P}(x_i \geq y_i(\xi) + a_i(\xi), \forall i \in I) \geq 1 - \varepsilon.$$

Here,  $\xi$  is a random variable,  $x$  is a vector of decisions of size  $I$ ;  $a_i(\xi)$  is a vector of data, for each  $i$ , that is known only after the realization of  $\xi$ ; and,  $y_i(\xi)$  is a vector of decisions, for each  $i$ , made after observing the realization of  $\xi$ . We require the joint probability of

satisfying the above inequality to be no less than  $1 - \varepsilon$ , where  $\varepsilon$  is a predetermined threshold. Thus, if we “fail” in even a single  $i$ , we fail to satisfy the JCC. In this sense, the JCC represents a very stringent satisfaction criterion. Computationally, optimization models with JCCs are hard to solve [37], and several tailored decomposition approaches [34,36], as well as heuristics [6,60], are available for their solution. Theoretically, such optimization models are known to be  $\mathcal{NP}$ -hard [37]; further, the feasible region is generally non-convex [47], with a few exceptions such as [23].

Nonetheless, an optimization model with JCCs is an attractive modeling choice for ensuring reliability of a set of operations. Chance-constrained models find frequent applications, e.g., power systems [28,62], traffic flow [13,59], portfolio selection [1,32], etc. There exists a modeling choice with regards to the timing of observing the uncertainty. In the first case, the entire set of future scenarios is known after making the first-stage decision,  $x$ . In the second case, the uncertainty is revealed periodically and subsequent decisions need to be made in multiple stages. Such models are known as two-stage and multi-stage stochastic programs, respectively; see, e.g., [47]. Multi-stage models are computationally harder than two-stage models. Since optimizing over JCCs already poses a significant challenge, in this tradeoff between fidelity and computational effort, we use a two-stage model instead of a multi-stage model. In our model, we make a first-stage decision of 24 hourly day-ahead power promises. Then, we observe the previously uncertain solar power output and make all the second-stage decisions of charging/discharging the battery. We use a highly reliable regime of operations ( $\varepsilon \leq 0.05$ ). Our motivation comes from the fact that failure to satisfy the promise could lead to curtailment of future contracts with the supplier, and thus ensuring reliability is valued more than increasing profit. We describe our model in greater detail in Sect. 2.

## 2 Mathematical modeling

### 2.1 Notation

*Indices and Sets:*

$t \in T$	Set of hours; $\{1, 2, \dots,  T \}$
$\omega \in \Omega$	Set of scenarios; $\{\omega_1, \omega_2, \dots, \omega_{ \Omega }\}$

*Parameters: First Stage*

$\eta$	Efficiency of battery, $0 < \eta < 1$
$\bar{X}$	Maximum energy that can be stored in the battery [kWh]
$\underline{X}$	Minimum energy that must remain in the battery [kWh]
$R_t$	Marginal revenue earned at hour $t$ [\$/kWh]
$C_c$	Operational cost of charging the battery [\$/kWh]
$C_d$	Operational cost of discharging the battery [\$/kWh]
$\bar{P}$	Maximum charging rate of battery [kW]
$\bar{Q}$	Maximum discharging rate of battery [kW]
$\Delta$	Time step [h]

*Parameters: Second Stage*

$s_t^\omega$	Solar power available at hour $t$ under scenario $\omega$ [kW]
--------------	--

*Decision Variables: First Stage*

$y_t$  Power promised to deliver at hour  $t$  [kW]

*Decision Variables: Second Stage*

$p_t^\omega$  Power charged to battery during hour  $t$  in scenario  $\omega$  [kW]  
 $q_t^\omega$  Power discharged from battery during hour  $t$  in scenario  $\omega$  [kW]  
 $x_t^\omega$  Energy stored in battery at hour  $t$  in scenario  $\omega$  [kWh]  
 $w_t^\omega$  Binary variable; 1 if battery is charging at hour  $t$ , or 0 if discharging

**2.2 Optimization model**

$$\max \sum_{t \in T} R_t \Delta y_t - \mathbb{E}[C_c \Delta p_t^\omega + C_d \Delta q_t^\omega] \tag{1a}$$

$$\text{s.t. } \mathbb{P}(y_t \leq s_t^\omega + q_t^\omega - p_t^\omega, \forall t \in T) \geq 1 - \varepsilon \tag{1b}$$

$$x_{t+1}^\omega = x_t^\omega + \eta \Delta p_t^\omega - \frac{1}{\eta} \Delta q_t^\omega, \quad \forall t = 1, 2, \dots, |T| - 1, \forall \omega \in \Omega \tag{1c}$$

$$p_t^\omega \leq \bar{P} w_t^\omega, \quad \forall t \in T, \forall \omega \in \Omega \tag{1d}$$

$$q_t^\omega \leq \bar{Q}(1 - w_t^\omega), \quad \forall t \in T, \forall \omega \in \Omega \tag{1e}$$

$$\underline{X} \leq x_t^\omega \leq \bar{X}, \quad \forall t \in T, \forall \omega \in \Omega \tag{1f}$$

$$w_t^\omega \in \{0, 1\}, \quad \forall t \in T, \forall \omega \in \Omega \tag{1g}$$

$$y_t, p_t^\omega, q_t^\omega \geq 0 \quad \forall t \in T, \forall \omega \in \Omega. \tag{1h}$$

*Boundary condition:*  $x_1^\omega = x_0, \forall \omega \in \Omega$ .

The objective function in (1a) aims to maximize profit; i.e., revenue from the promise minus the expected cost of operating (charging or discharging) the battery. A more sophisticated model could include piecewise linear or quadratic costs; see, e.g., [12,63]. The JCC in Eq. (1b) requires that the joint probability of meeting the promised energy to sell, by the available solar power (via  $s^\omega$ ) and discharging the battery (via  $q^\omega$ ), for the entire time horizon meets a threshold; excess solar energy can be used to charge the battery (via  $p^\omega$ ). If the solar power exceeds the need, we might need to curtail it. However, in this article we do not penalize this curtailment, and hence Eq. (1b) is expressed as an inequality. Here,  $\varepsilon$  is a positive quantity just greater than zero, such as 0.05. Constraint (1c) relates the energy stored by the battery in a subsequent hour with the previous hour; here, the quantity  $\eta \Delta p_t^\omega - \frac{1}{\eta} \Delta q_t^\omega$  is the energy charged or discharged by the battery during hour  $t$ . Constraints (1d), (1e) and (1g) ensure that both  $p_t^\omega$  and  $q_t^\omega$  cannot be simultaneously positive. Further, since  $\eta < 1$ , a fraction of energy is lost while both charging and discharging; i.e., we consume more than 100% while charging and supply less than 100% while discharging. Constraint (1f) ensures the minimum and maximum amount of energy stored in the battery, and thus Eq. (1d)–(1h) restrict the amount the battery can be charged or discharged. Lower bounds on  $x$  are typically included for electric batteries to maintain long and healthy lifetimes; see, e.g., [40]. Constraints (1g)–(1h) ensure the binary and non-negativity restrictions on the relevant decision variables.

One potential criticism of optimization models with JCCs, such as model (1), is that violations can be large for the  $100 \cdot \varepsilon$  percent of failed scenarios in Eq. (1b). Although, Eq. (1b) restricts the number of violations to be a few, there is nothing in model (1) that restricts the magnitude of these violations. This well-known drawback is an artifact of chance-constrained

programming in general, and is also reflected in our model. Several methods exist to bound the violations. We direct interested readers to [50] and [64] for two remedies using risk measures and envelope constraints, respectively. However, models that only penalize energy violations (or, do not employ chance constraints) can yield solutions that violate the promise with large frequency when it is economically advantageous. Our aim is to decide whether it is feasible to couple a solar plant with an appropriately sized battery unit, and whether this coupling would enable the hybrid unit to participate in the day-ahead market.

Now, consider the following optimization model in which we drop the binary variable  $w_t^\omega$ :

$$\max \sum_{t \in T} R_t \Delta y_t - \mathbb{E}[C_c \Delta p_t^\omega + C_d \Delta q_t^\omega] \tag{2a}$$

$$\text{s.t. } \mathbb{P}(y_t \leq s_t^\omega + q_t^\omega - p_t^\omega, \forall t \in T) \geq 1 - \varepsilon \tag{2b}$$

$$x_{t+1}^\omega = x_t^\omega + \eta \Delta p_t^\omega - \frac{1}{\eta} \Delta q_t^\omega, \quad \forall t = 1, 2, \dots, |T| - 1, \forall \omega \in \Omega \tag{2c}$$

$$p_t^\omega \leq \bar{P}, \quad \forall t \in T, \forall \omega \in \Omega \tag{2d}$$

$$q_t^\omega \leq \bar{Q}, \quad \forall t \in T, \forall \omega \in \Omega \tag{2e}$$

$$\underline{X} \leq x_t^\omega \leq \bar{X}, \quad \forall t \in T, \forall \omega \in \Omega \tag{2f}$$

$$y_t, p_t^\omega, q_t^\omega \geq 0 \quad \forall t \in T, \forall \omega \in \Omega. \tag{2g}$$

**Proposition 1** *In an optimal solution to model (2), both  $p_t^\omega$  and  $q_t^\omega$  cannot be positive together for any  $t \in T, \omega \in \Omega$ .*

**Proof** Assume an optimal solution of model (2) is given by  $y^* = [y_t]_{t=1,2,\dots,|T|}, p^* = [p_t]_{t=1,2,\dots,|T|}, q^* = [q_t]_{t=1,2,\dots,|T|}, x^* = [x_t]_{t=1,2,\dots,|T|}, \forall \omega \in \Omega$ . Assume that  $p_t^\omega$  and  $q_t^\omega$  are positive, for some  $t \in T, \omega \in \Omega$ . We show that this solution cannot be optimal by proving a better solution (one that gives a larger objective function value) always exists.

First, assume that  $\eta^2 p_t^\omega > q_t^\omega$ , for this  $t$  and  $\omega$  pair. Then, consider a new solution  $p_t'^\omega \leftarrow p_t^\omega - \frac{1}{\eta^2} q_t^\omega, q_t'^\omega \leftarrow 0, x_t'^\omega \leftarrow x_t^\omega, y_t' \leftarrow y_t$ . We observe that this solution is feasible to model (2); further, it results in a strictly larger objective function than that by the optimal solution. This is a contradiction.

Next, assume that  $\eta^2 p_t^\omega \leq q_t^\omega$ , for this  $t$  and  $\omega$  pair. Again, consider a new solution  $p_t'^\omega \leftarrow 0, q_t'^\omega \leftarrow q_t^\omega - \eta^2 p_t^\omega, x_t'^\omega \leftarrow x_t^\omega, y_t' \leftarrow y_t$ . This solution is feasible to model (2), and results in a strictly larger objective function than that by the optimal solution. This is a contradiction.

Thus, in any optimal solution to model (2), for any  $t$  and  $\omega$  at most one of  $p_t^\omega$  or  $q_t^\omega$  is positive.  $\square$

Proposition 1 offers the advantage that there are no binary  $w = [w_t]_{t=1,2,\dots,|T|}$  variables in the second-stage. Although, a reformulation of constraint (1b) would still entail second-stage binaries (see, Eq. (3)), the reformulation in model (2) could save computational effort by reducing  $|T||\Omega|$  binary variables. There is some evidence, however, that having additional binary variables can actually speed up the computations by offering mixed-integer program (MIP) solvers additional variables to branch and cut on; see, e.g. [41,56]. However, we choose the reformulation without the binary  $w$  variables as it lends itself better to the heuristics we describe in the later sections of this article. We reformulate the probabilistic constraint in Eq. (2b) using a big- $M$  approach as follows:

$$y_t - q_t^\omega + p_t^\omega \leq s_t^\omega + M_t^\omega z^\omega, \forall t \in T, \omega \in \Omega \tag{3a}$$

$$\sum_{\omega \in \Omega} z^\omega \leq \lfloor N\varepsilon \rfloor \tag{3b}$$

$$z^\omega \in \{0, 1\}, \quad \forall \omega \in \Omega. \tag{3c}$$

Here,  $\lfloor \cdot \rfloor$  rounds its argument down to the nearest integer. Henceforth, we use model (2) with Eq. (2b) reformulated as Eq. (3) as our proposed model. Below, we provide a sufficiently large value for the big- $M$  of Eq. (3).

**Proposition 2** *The following inequality is a valid inequality for model (2):  $y_t \leq \min\{\bar{Q}, \eta(\bar{X} - \underline{X})\} + s_t^{\omega(\lfloor N\varepsilon \rfloor + 1, t)}$ ,  $\forall t \in T$ , where  $s_t^{\omega(l, t)}$  denotes the  $l^{\text{th}}$  largest solar power value at time  $t$ .*

**Proof** We first seek an upper bound on the quantity  $q_t^\omega - p_t^\omega$ . If  $q_t^\omega = 0$ , a valid upper bound on this quantity is 0. If  $p_t^\omega = 0$ , a valid upper bound on this quantity is  $\min\{\bar{Q}, \eta(\bar{X} - \underline{X})\}$ ; this is obtained from Eqs. (2c), (2e) and (2f). Thus, from Proposition 1,  $q_t^\omega - p_t^\omega \leq \min\{\bar{Q}, \eta(\bar{X} - \underline{X})\}$ ,  $\forall t \in T, \omega \in \Omega$ . Now, the proof replicates that of Proposition 5.2.1 of [55]. By constraint (3b) effectively at most  $\lfloor N\varepsilon \rfloor$  of the  $z^\omega$  variables can take a value of 1; i.e., for every  $t$  at most  $\lfloor N\varepsilon \rfloor$  of the scenarios can be removed. Thus, the proposed inequality is valid.

**Corollary 1** *From Proposition 2, we have  $M_t^\omega \geq \min\{\bar{Q}, \eta(\bar{X} - \underline{X})\} + s_t^{\omega(\lfloor N\varepsilon \rfloor + 1, t)} - s_t^\omega$ ,  $\forall t \in T, \omega \in \Omega$ .*

We denote the upper bound, obtained from Proposition 2, on the  $y$  variables as  $y_t^{\max} = \min\{\bar{Q}, \eta(\bar{X} - \underline{X})\} + s_t^{\omega(\lfloor N\varepsilon \rfloor + 1, t)}$ ,  $\forall t \in T$ . And, in our computational experiments of Sect. 6 we use the tightest big- $M$  given from Corollary 1, namely  $M_t^\omega = y_t^{\max} - s_t^\omega$ ,  $\forall t \in T, \omega \in \Omega$ . Despite this, as we show later in this article, model (2) is computationally challenging for relatively larger values of  $\varepsilon$  (such as 0.05) or, for a large number of scenarios. Model (2) is a classic two-stage joint chance-constrained stochastic program with recourse. The second stage has integer restrictions (the variable  $z$  is binary) as well, which makes it even more challenging [2]. To this end, we investigate a Lagrangian relaxation scheme as well as a heuristic, in addition to the traditional approach with the big- $M$  constraints.

### 3 A Lagrangian dual procedure

First, we note that constraint (3b) links the scenarios together. This motivates a straightforward Lagrangian relaxation of model (2). Lagrangian relaxation schemes have been studied before in various settings; see, e.g., individual chance constraints for unit commitment [42], relaxing non-anticipativity constraints [3], and coupling scenario decomposition with Lagrangian decomposition [60].

The Lagrangian dual for a given value of  $\lambda \geq 0$  for model (2), obtained by dualizing the constraint (3b), is the following:

$$L(\lambda) = \max \sum_{t \in T} \left( R_t \Delta y_t - \mathbb{E}[C_c \Delta p_t^\omega + C_d \Delta q_t^\omega] \right) + \lambda \left( \lfloor N\varepsilon \rfloor - \sum_{\omega \in \Omega} z^\omega \right) \tag{4}$$

s.t. (2c) – (2g), (3a), (3c).

To solve model (4), we use the well-known subgradient method [45,54], with the step-size update rule from [21]. Algorithm 1 summarizes this scheme; we need a lower bound

**Algorithm 1** Lagrangian Relaxation of model (2). We use  $M = 10$ ,  $time = 2400$  seconds,  $\psi = 0.001$ .

**Input:**  $M, LB, time, \theta, \psi$ .

- 1: Solve LP relaxation of model (2), set  $UB$  to optimal objective value of LP, set  $\lambda$  to the optimal dual of constraint (3b).
- 2: **while** iter  $\leq M$  **do**
- 3: Solve model (4), obtain optimal  $z^\omega, UB \leftarrow \min\{UB, L(\lambda)\}$ .
- 4:  $\gamma \leftarrow \lfloor N\varepsilon \rfloor - \sum_{\omega \in \Omega} z^\omega$  and  $\kappa \leftarrow \theta \frac{(UB-LB)}{\gamma^2}$ .
- 5: if  $\gamma > 0$ , then  $\lambda \leftarrow \lambda - \min\{\kappa, \frac{\lambda}{\gamma}\}\gamma$ .
- 6: if  $\gamma < 0$ , then  $\lambda \leftarrow \lambda - \kappa\gamma$ .
- 7: if no improvement in two iterations,  $\theta \leftarrow \frac{\theta}{2}$ .
- 8: if  $\frac{UB-LB}{UB} \leq \psi$  or  $time \geq time$ , *STOP*.
- 9: iter  $\leftarrow$  iter+1, update time to the cumulative wall-clock time.
- 10: **end while**

for model (2) for the step-size update rule in Step 4. In principle, this lower bound can be computed using any feasible solution; setting  $\lfloor N\varepsilon \rfloor$  of the  $z$  variables to 1, and the rest to 0, provides a feasible solution. However, since the termination criteria of the Lagrangian relaxation procedure depends on this lower bound, good feasible solution values (those giving a larger value of the objective function) assist in convergence.

*Lower Bounding Heuristic:* To this end, we obtain a feasible solution using the following simple procedure described in [3]. We first solve model (2) separately for each  $\omega$  with the corresponding  $z^\omega \leftarrow 0$ ; i.e., a total of  $|\Omega|$  problems. We rank the corresponding objective function values from the lowest to the highest and choose the first  $\lfloor N\varepsilon \rfloor$  (i.e., the worst performing scenarios) of these to set  $z^\omega \leftarrow 1$ . Then, with  $\lfloor N\varepsilon \rfloor$  of the  $z$  values fixed to 1, we solve model (2), to obtain a valid lower bound (LB) to the problem.

For an initial upper bound to the Lagrangian relaxation procedure, we use the linear programming (LP) relaxation of model (2). In the following proposition, we show that model (4) is never infeasible.

**Proposition 3** *The feasible region of model (4) is non-empty.*

**Proof** Set  $p_t^\omega \leftarrow 0, q_t^\omega \leftarrow 0, x_t^\omega \leftarrow x_0, z^\omega \leftarrow 1, y_t \leftarrow 0, \forall t \in T, \forall \omega \in \Omega$ . Clearly, this choice of  $p_t^\omega, q_t^\omega, x_t^\omega, z^\omega, y_t$  is feasible to model (4) for all  $t \in T, \omega \in \Omega$ .

## 4 A progressive hedging heuristic for the Lagrangian dual

The progressive hedging (PH) algorithm can be used to solve stochastic programs that have easy-to-solve individual scenario problems [48]. While PH cannot be directly applied to model (2) due to the interlinking of the scenarios via constraint (3b), the relaxed model (4) lends itself perfectly to the decomposition structure required for PH. In this section, we leverage this idea to develop computationally cheap heuristics for the Lagrangian dual. First, we convert our maximization problems to a minimization by reversing the signs on the objective coefficients ( $R, C_c, C_d$ ), since most of the existing literature on PH considers minimization [16,60,61]. Then, model (5) is the PH sub-problem; we summarize the PH heuristic in Algorithm 2.



**Algorithm 2** Progressive Hedging Algorithm for model (4). We use  $\kappa = 0.001$ ,  $\rho_t = 1$ .

**Input:**  $\rho_t, \kappa, iter, \lambda$ .

1: Solve model (5),  $\forall \omega \in \Omega$ , with  $u \leftarrow 0, \rho \leftarrow 0$ , obtain optimal  $y_t^\omega$ . Set  $PH_{UB} \leftarrow \sum_{\omega \in \Omega} PH^\omega$ .

2:  $\bar{y}_t \leftarrow \sum_{\omega \in \Omega} y_t^\omega / |\Omega|$ , and  $u_t^\omega \leftarrow \rho_t (y_t^\omega - \bar{y}_t)$ .

3: **while**  $iter \leq \overline{iter}$  **do**

4: Solve model (5),  $\forall \omega \in \Omega$ , obtain optimal  $y_t^\omega$ .

5:  $\bar{y}_t \leftarrow \sum_{\omega \in \Omega} y_t^\omega / |\Omega|$ , and  $u_t^\omega \leftarrow u_t^\omega + \rho_t (y_t^\omega - \bar{y}_t)$ .

6: Solve model (5) without the quadratic term in objective,  $\forall \omega \in \Omega$ , set  $PH_{UB} \leftarrow \min\{PH_{UB}, \sum_{\omega \in \Omega} PH^\omega\}$ .

7: if  $\sum_{\omega \in \Omega} \sqrt{\sum_{t \in T} (\bar{y}_t - y_t^\omega)^2} \leq \kappa$ , *STOP*; else  $iter \leftarrow iter+1$ .

8: **end while**

9: Solve model (4) with  $y_t \leftarrow \bar{y}_t$ , obtain optimal  $z^\omega$ .

**Output:**  $z^\omega, UB \leftarrow PH_{UB}$ .

$$\begin{aligned}
 PH^\omega = & - \min_{t \in T} \left( -R_t \Delta y_t - [-C_c \Delta p_t^\omega - C_d \Delta q_t^\omega] \right) - \lambda \left( \frac{|N\varepsilon|}{|\Omega|} - z^\omega \right) \\
 & + \sum_{t \in T} u_t^\omega y_t + 0.5 \sum_{t \in T} \rho_t (y_t - \bar{y}_t)^2 \\
 \text{s.t. } & (2c) - (2g), (3a), (3c).
 \end{aligned} \tag{5a}$$

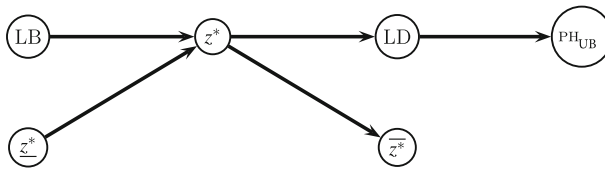
Here,  $u_t^\omega$  is the scenario-dependent weight,  $\bar{y}_t$  is the scenario-averaged value for  $y_t$ , and  $PH^\omega$  is the optimal value of each sub-problem. Solving the LR dual approximately via the PH algorithm offers several advantages and disadvantages compared to the naive solution method. We discuss some of these in Sect. 6.5.

In the presence of discrete decision variables (as is the case with our model), the PH algorithm can only be used as a heuristic as its convergence is not guaranteed [35,61]. Since we are interested in solving the Lagrangian dual, to obtain an upper bound to the true problem given by model (2), any upper bound to the Lagrangian dual is also an upper bound to the true problem. Consider Fig. 1; here, LB is the lower bound obtained via the heuristic procedure described in Sect. 3,  $z^*$  is the true optimal value of model (2), and LD is the optimal value of model (4). Since convergence of the PH algorithm is not guaranteed, if we solve the Lagrangian dual (LD) via the PH algorithm, the PH optimal objective function value could be larger or smaller than LD.

If the PH algorithm results in a bound larger than LD at any subgradient iteration of Algorithm 1, then the bound is still a valid upper bound to  $z^*$ . However, if the PH algorithm results in a bound smaller than LD at any subgradient iteration of Algorithm 1, then the bound could also be smaller than  $z^*$  (or, even smaller than LB). To prevent this case, we seek valid (i.e., guaranteed to be larger than LD) upper bounds to LD via the PH algorithm (which may or may not be optimal to PH). Such bounds can be obtained by solving model (5) without the quadratic term in the objective using the procedure described in [16]. This bound is represented in Fig. 1 by  $PH_{UB}$ , and is computed in Algorithm 2 in Steps 1 and 6.

### 5 Data sources

For the computational experiments in this article, we use  $\eta = 0.9$  [29], and values of  $R_t$  from the Electric Reliability Council of Texas (ERCOT) for the year 2012 [15,56]. Table 1 presents the hourly marginal selling prices. Further, we use  $\bar{X} = 960\text{kWh}$ , with a 2V/1000Ah



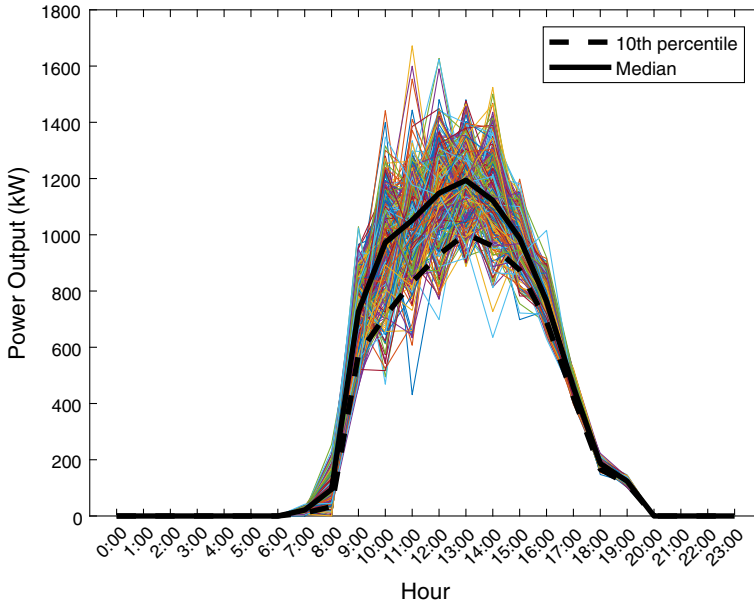
**Fig. 1** Bounds for model (2). An arrow from a source node to a sink node indicates the optimal value of the source node is smaller than the sink node. Here,  $z^*$  is the true optimal value, LB is the lower bound from a heuristic, LD is the optimal value of the Lagrangian dual,  $PH_{UB}$  is an upper bound for the Lagrangian dual obtained via PH, and  $\bar{z}^*$  and  $\underline{z}^*$  are the upper and lower bounds for  $z^*$  if solved sub-optimally. An optimality gap of  $\beta$  between an upper bound  $U$  and a lower bound  $L$  indicates  $\beta = \frac{U-L}{U}$ . The optimal PH value is not shown and could be larger or smaller than  $z^*$

**Table 1** Marginal revenue,  $R_t$ , earned by selling energy at hour  $t$  [15,56]

Hour	Selling price (\$/kWh)	Hour	Selling price (\$/kWh)
0:00	0.0189	12:00	0.0250
1:00	0.0172	13:00	0.0261
2:00	0.0155	14:00	0.0285
3:00	0.0148	15:00	0.0353
4:00	0.0146	16:00	0.0531
5:00	0.0151	17:00	0.0671
6:00	0.0173	18:00	0.0438
7:00	0.0219	19:00	0.0333
8:00	0.0227	20:00	0.0287
9:00	0.0226	21:00	0.0268
10:00	0.0235	22:00	0.0240
11:00	0.0242	23:00	0.0211

rating, from the system with 480 lead-acid batteries described in [68]. To compute the cost of charging or discharging the battery, we again use the formula given by [68]:  $390\alpha QI$ , where  $\alpha$  is an effective weighting factor,  $I$  is the initial investment cost to purchase the battery, and  $390Q$  is an approximation for the total Ah throughput of the battery [27]. For the system in [68], a  $\bar{X}$  kWh sized battery is equivalent to  $\frac{2000}{960} \bar{X}$  Ah rating. We assume  $\alpha = 0.5$ , and use a lead-acid battery cost of \$200/kWh from [52]. An approximately similiar battery cost is available from [46]. Then,  $C_c = C_d = \$0.0256/\text{kWh}$ . Further, we assume  $\underline{X} = 0.2\bar{X}$ ,  $\bar{P} = \bar{Q} = 0.5\bar{X}$ , and the battery is 50% charged initially (i.e.,  $x_0 = 0.5\bar{X}$ ).

To generate solar power scenarios, we take hourly year-long historical solar power output from a site described in [19]. We use ARMA( $p, q$ ) models for the fourteen hours with sunlight, [06:00-20:00). For each hour, we verify the stationarity of the time series and test a number of ARMA( $p, q$ ) models to find the best. For details on the forecasting method we used, see [57]. Then, using Monte Carlo sampling from the best ARMA models for each hour, we create a set of hourly scenarios. Figure 2 plots a sampling of 300 scenarios, as well as the hourly medians and the ten percent quantile.



**Fig. 2** 300 hourly scenarios (unscaled) for solar power generated using the scheme described in [57]. The solid black line is the median hourly value, and the dashed black line is the 10 percentile solar power value

## 6 Computational experiments

### 6.1 Computational setup

All tests in this article are carried out with GAMS 25.1.2 using CPLEX 12.8 [49] on an Intel Core i7 2.8 GHz processor with 16 GB of memory. We attempt to solve the original model (2) to a MIP gap of 0.1%, with a maximum time limit of 40 min. For the rest of the experiments in this article, we also consider an optimality gap of 0.1%; i.e., if the relative gap between the best upper and lower bounds is less than or equal to 0.1% we consider the problem solved. We use multiple sets of scenarios—the smallest instance with 150 scenarios and the largest instance with 1200 scenarios—with two reliability regimes having  $\varepsilon = 0.01, 0.05$ .

### 6.2 Analysis of naive solution method

In Table 2 we present our first results obtained by solving model (2) naively. The “ $z^*$ ” column denotes the 24-h profit obtained in dollars; if the problem is not solved to the optimal tolerance, we present the best feasible solution. The  $\varepsilon = 0.01$  instance is relatively easier (smaller computation time) than the  $\varepsilon = 0.05$  instance due to the fewer number of combinatorics involved, and possibly due to a smaller value of the big- $M$  via Corollary 1; the problem can be solved to the optimal tolerance for all instances except  $|\Omega| = 1200$ . The gaps increase consistently when  $|\Omega|$  increases; beyond 600 scenarios the  $\varepsilon = 0.05$  case cannot be solved to the optimal tolerance.

**Table 2** Computational results for model (2). We use a maximum time limit of 2400 seconds and a MIP gap of 0.1%. Entries with a “-” indicate that the problem is solved to the required gap within the time limit; i.e.,  $\underline{z}^* = \overline{z}^* = z^*$ . For details, see Sect. 6.2 and Fig. 1

$ \Omega $	$\varepsilon$	$\underline{z}^*$	Gap	Time
150	0.05	280.90	–	30
	0.01	273.01	–	2
300	0.05	283.79	–	870
	0.01	278.49	–	29
450	0.05	283.73	–	1932
	0.01	278.93	–	96
600	0.05	280.65	6.7%	2400
	0.01	277.58	–	234
900	0.05	278.66	7.6%	2400
	0.01	276.83	–	1341
1200	0.05	271.76	9.5%	2400
	0.01	270.95	4.6%	2400

### 6.2.1 Analysis: solution profile

Next, we provide a profile of an optimal solution. We choose the  $\varepsilon = 0.05$ ,  $|\Omega| = 300$  instance; i.e., the third row of Table 2. To this end, we first solve the instance to a MIP gap of 0%, with an indefinite time limit. The optimal profit for the 24 hour horizon is \$283.79; i.e., the value reported in Table 2 is indeed optimal to a MIP gap of 0%. The selling price,  $R_t$ , exceeds the operational cost of the battery,  $C_c = C_d$ , for the nine hours [13:00–22:00), see Table 1. Solar power is non-zero, for at least one scenario, only for the fourteen hours [6:00–19:00), see Fig. 2. We provide the characteristics of the optimal solution in Table 3.

No power is promised during the seven night hours from midnight to the hour 6:00AM (i.e., until 6:59AM) and the three evening hours from 20:00 to the hour 22:00 (i.e., until 22:59PM). This is because the cost of operating the battery is higher than the reward for the seven night hours, and further there is no solar power available in these hours. In the three evening hours, although the cost of operation is (slightly) less than the reward, the model still chooses to not make a promise. This is because there is no solar power, and the model prefers to completely drain the battery to its lower limit in the last hour. Most of the hours meet the energy promise largely through the solar power, which is dictated by the economics of model (2). However, there is a good synergy between the charging and discharging times of the battery; i.e., (i) the promise made does not rely entirely on the solar power, and (ii) for the same hour the model chooses to charge or discharge based on the observed solar power scenario. An extreme example is the hour 17:00–17:59. In this hour, the battery is discharged (on average) to over 80% of its discharge limit to increase profit—solar power is decreasing beyond this hour, and this hour has the largest marginal profit ( $R_t - C_c$ ).

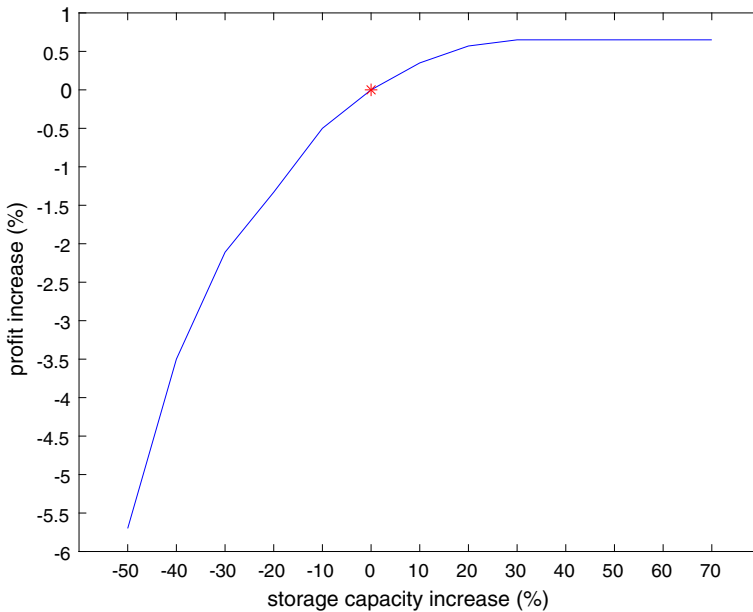
Also, we note that excess solar power in the afternoon hours is used to charge the battery. Hence, we see that the model is successfully using the battery to shift solar power to times when energy commands a higher price. Finally, as we mentioned before, at the last hour of 23:00–23:59, the battery is completely discharged to its permissible limit. To counter this end-effect, the model can be run in a rolling-time horizon or a boundary condition could be imposed; see, e.g., [30].

**Table 3** Profile of the optimal solution for model (2) with  $\varepsilon = 0.05$  and  $|\Omega| = 300$ . The average energy charged for hour  $t$  is  $\frac{\sum_{\omega \in \Omega} P_t^\omega}{|\Omega|}$ , the average energy discharged is  $\frac{\sum_{\omega \in \Omega} q_t^\omega}{|\Omega|}$ , and the average solar energy available is  $\frac{\sum_{\omega \in \Omega} s_t^\omega}{|\Omega|}$ . A blank in the table indicates zero. For details, see Sect. 6.2.1

Hour	Promised energy (kWh)	Average energy charged (kWh)	Average energy discharged (kWh)	Average solar energy available (kWh)
0:00				
1:00				
2:00				
3:00				
4:00				
5:00				
6:00		0.01		0.19
7:00	12.78	0.77	0.57	20.70
8:00	39.26	5.28	2.79	97.76
9:00	620.80	26.89	11.08	725.45
10:00	837.18	60.05	22.53	959.92
11:00	911.42	68.25	17.20	1042.44
12:00	1004.14	64.00	16.99	1147.96
13:00	1077.48	49.11	15.26	1183.66
14:00	946.56	80.69	7.04	1127.01
15:00	913.47	19.66	9.97	986.78
16:00	807.63	2.09	47.01	767.36
17:00	877.07		405.65	457.87
18:00	185.30	0.08	5.73	186.90
19:00	123.15		2.66	123.77
20:00				
21:00				
22:00				
23:00	480.00		480.00	

### 6.2.2 Analysis: storage size

Next, we analyze the impact of the size of the battery storage on the optimal profit. Similar to Sect. 6.2.1, we choose the  $\varepsilon = 0.05$ ,  $|\Omega| = 300$  instance. We recompute the optimal profit from model (2) by varying the maximum storage capacity of the battery,  $\bar{X}$ , in increments of 10%. Since we set  $x_0 = 0.5\bar{X}$  and  $\underline{X} = 0.2\bar{X}$  in our computational experiments, the lowest feasible limit of  $\bar{X}$  is  $0.5\bar{X}$  corresponding to a 50% decrease. Figure 3 presents the pareto-optimal curve for the percentage changes in the optimal profit by varying the maximum storage capacity of the battery. The red dot presents the base case; i.e., a profit of \$ 283.79 for a battery size of  $\bar{X} = 960$  kWh. There is no change in the optimal profit beyond a size of 1344 kWh; this suggests that for the given parameters of the model a battery of size  $\bar{X} = 1344$  kWh is sufficiently large.



**Fig. 3** Effect of varying the maximum storage capacity of the battery on the optimal profit

### 6.2.3 Analysis: quality of the solution

In this section, we perform an out-of-sample validation to analyze the performance of the optimal power promise. Since model (2) is solved using a given set of scenarios by a so-called sample average approximation (SAA) of the “true” model, this solution is naturally sub-optimal or infeasible to a new set of scenarios [4]. Thus, we seek to answer the following question: to what minimum reliability level can we guarantee a given power promise profile to safeguard us against new scenarios? Intuitively, we expect this reliability level to be lower than the chosen  $1 - \varepsilon$  level.

To this end, we first independently sample ten batches of 300 scenarios each. We fix the promised power output to the optimal power output of the base case instance with  $|\Omega| = 300$  instance. And, then we find the minimum number of violations required to ensure feasible operations (i.e., constraints (2c)–(2g)) in each of the ten batches. We construct a 95% confidence interval (CI) around the number of violations using a  $t$ -distribution. For the SAA solved with a 95% reliability regime, the 95% CI of the true reliability is (85.1%, 89.5%). Analogously for the 99% reliability regime, the 95% CI is (92.9%, 95.4%).

### 6.3 Analysis of lower bound heuristic

In this section, we briefly analyze the performance of the lower bounding heuristic mentioned in Sect. 3. Table 4 presents our results. The “LB” values are the lower bounds to model (2), see also Fig. 1. We solve all the scenario sub-problems using the Gather-Update-Solve-Scatter extension of GAMS [10]. This scheme is very efficient; 1200 scenario problems plus the fixed scenario problem can be solved in about 30 seconds. To compute the “Gap”, we use the best known upper bound—this is available from either the naive solution (Table 2),

**Table 4** Computational results for lower bounding heuristic. The “Gap” denotes the optimality gap of the LB from  $\min\{LD, z^*\}$ ; see, Fig. 1 for the notation. For details, see Sect. 6.3

$ \Omega $	$\varepsilon$	LB	Gap	Time
150	0.05	280.28	0.2%	6
	0.01	273.01	–	3
300	0.05	283.33	0.2%	7
	0.01	278.17	–	4
450	0.05	273.32	3.7%	7
	0.01	276.40	0.9%	7
600	0.05	275.58	5.0%	17
	0.01	274.53	1.1%	16
900	0.05	277.79	4.1%	22
	0.01	275.69	0.4%	29
1200	0.05	277.05	2.7%	31
	0.01	267.98	2.0%	35

or from the Lagrangian relaxation that we present later (Table 5). For example, for the  $|\Omega| = 1200, \varepsilon = 0.05$ , this upper bound is  $\min\{\frac{271.76}{1-0.095}, 284.77\}$ . A “–” in Table 4 indicates the gap is smaller than the optimality tolerance. Thus, six of the twelve problems have a gap under 1%, while two are solved to optimality.

### 6.4 Analysis of the Lagrangian relaxation scheme

In this section, we attempt to solve the Lagrangian relaxation of model (2) given by model (4) using Algorithm 1. We use a maximum of 10 iterations (i.e.,  $M = 10$ ). We use the same maximum time limit (*time*) and optimality tolerance ( $\psi$ ) of 2400 seconds and 0.1%, respectively, as for the results in Table 2. Theoretically, this gap might not be attained even with a larger time limit as (i) different schemes for obtaining a lower bound give different corresponding gaps, and (ii) even the tightest upper bound from the Lagrangian procedure might not be attained as our problem includes binary variables. However, as we show later in this section our empirical results demonstrate that in some instances an optimality gap smaller than the naive solution is achieved. We use the LB bound from Sect. 6.3 in Step 4 of Algorithm 1. Further, we “warm-start” the  $z$  variables for the first LR iteration, using the optimal  $z$  values from the LB heuristic. Also, at each subgradient iteration, we warm-start the  $y$  variables with their optimal value from the previous subgradient iteration. In CPLEX, this can be done by setting the `mipstart` parameter to 1.

Table 5 presents our results. All 10 subgradient iterations complete in the required time limit, except for  $|\Omega| = 900, 1200$ . In our experiments, if ten LR iterations perform within the time limit, the optimal Lagrangian dual (denoted by “LR”) is typically achieved around the sixth iteration. The four instances ( $\varepsilon = 0.05, |\Omega| = 600, 900, 1200$  and  $\varepsilon = 0.01, |\Omega| = 1200$ ) that are not solved to optimality in Table 3 all have smaller gaps in Table 5 than Table 2, suggesting the Lagrangian relaxation is better suited for larger problems. Also note, that for these four instances, only a few subgradient iterations are completed. This again lends credence to the observation that good Lagrangian bounds are obtained fairly early in the iterations.

The optimal solution provided by the Lagrangian relaxation is infeasible to model (2). Thus, the algorithm is useful only for providing good bounds to model (2), and not for obtain-

**Table 5** Computational results for model (2) using Algorithm 1. “LB” and “LR” refer to lower bound and Lagrangian relaxation, respectively. The “Gap” column is the relative gap between the LB and LR. We use a maximum time limit of 2400 seconds (cumulative sum of LP, LB and LR) and a gap of 0.1%. For details, see Sect. 6.4

Ω	ε	Bounds			Performance	
		LB	LR	Gap	LR Time	LR iterations
150	0.05	280.28	284.80	1.6%	56	10
	0.01	273.01	273.01	–	8	3
300	0.05	283.33	291.59	2.8%	272	10
	0.01	278.17	279.61	0.5%	210	10
450	0.05	273.32	291.09	3.9%	980	10
	0.01	276.40	279.14	1.0%	660	10
600	0.05	275.58	290.15	5.0%	1936	10
	0.01	274.53	277.98	1.2%	1581	10
900	0.05	277.79	289.68	4.1%	2369	2
	0.01	275.69	276.83	0.4%	2362	4
1200	0.05	277.05	284.77	2.7%	2356	3
	0.01	267.98	273.37	2.0%	2351	4

ing high quality feasible solutions. At best, the optimal  $y$  variables from the last iteration of Algorithm 1 could be used to warm-start model (2). This aspect of Lagrangian decomposition techniques is a well-known drawback; researchers have addressed this issue in a number of ways, see, e.g., [66,71]. A similar issue is encountered in Step 9 of Algorithm 2 as well; model (4) could be infeasible. However, if  $\bar{y}_t \leq \min\{\bar{Q}, \eta(\bar{X} - \underline{X})\} + s_t^{\omega(|N\varepsilon|+1,t)}$ ,  $\forall t \in T$ , then it follows from Proposition 3 that model (4) is assuredly feasible.

### 6.5 Analysis of the progressive hedging heuristic

The performance of the Lagrangian relaxation algorithm in Sect. 6.4 suggests little room for improving the optimality gaps. In this section, we pursue a different approach, specifically aimed at achieving fast bounds, albeit of a lower quality. To this end, we also analyze the advantages and disadvantages of using the PH heuristic presented in Sect. 4 and Algorithm 2. As we describe in Sect. 4, PH offers theoretical difficulties, however they are often marred by the algorithm’s practical performance. This makes PH a powerful heuristic for practical problems; similar trends have been reported earlier [51,61].

#### 6.5.1 Choice of ρ

Algorithm 2 is known to be sensitive to the choice of the parameter  $\rho$  [16,61]; no consensus seems to exist in the literature for deciding optimal values of  $\rho$ . We compare the sensitivity of our results using three values of  $\rho_t$ , in addition to the  $\rho_t = 1$  results reported above. In the first two runs, we use constant  $\rho_t$  values of 0.1 and 10, and in the third run we use a coefficient specific  $\rho$  value described in [61]. This value is  $\rho_t = \frac{R_t}{\max\{1, \sum_{\omega \in \Omega} (\bar{y}_t - y_t^\omega) / |\Omega|\}}$ , where the  $y$  values are those obtained in Steps 1 and 2 of Algorithm 2. We observe the following trends: larger values of  $\rho$  result in weaker upper bounds of Step 6 of Algorithm 2, but close the gap



in Step 7 faster. This is similar to the trend observed in [16]. However,  $\rho_t = 0.1$  results in weaker upper bounds than  $\rho_t = 1$ . The variable specific  $\rho$  closes the gap fastest, however the improvements compared to  $\rho_t = 10$  are minimal. Further, none of the  $\rho$  values completely close the gap; and,  $\rho_t = 1$  provides the best upper bounds.

### 6.5.2 Linearization

The quadratic proximal term in the PH objective can be linearized using piece-wise linear segments. While, there are many sophisticated techniques for this linearization [22,58], we attempt a basic first-order Taylor expansion. Dropping the  $t$  subscript, with  $L - 1$  segments, we have:  $y^2 \approx a^2 + 2a(y - a)$ , where  $a = \frac{(l-1)y_{\max}}{L-1}$ ,  $l = 1, 2, \dots, L$ . However, in our experiments, we observe that to achieve solutions comparable to that of the original mixed-integer quadratic constrained program (MIQCP), a large value for  $L$  — close to 100 — is needed. Then, the time required to solve the larger MIPs is no significantly different from that of the MIQCP. The reason for this is as follows.

There is only a single binary variable for each subproblem in Step 6 of Algorithm 2; thus, each MIP can be solved in at most two LP iterations (by branching on  $z^\omega = 0$  and  $z^\omega = 1$ ). When  $L$  increases, the time required for an LP solution increases due to the increased number of constraints in the LP. Poor values of  $y$  reported by PH cascade to poor values for  $z$  in Step 9 of the algorithm, resulting in poor values for  $\gamma$ , and ultimately poor performance for the Lagrangian dual. Two possible suggestions for the interested reader, that we do not investigate, are as follows. First, use a dynamic linearization as described in Veliz et al. [58] that gradually increments the number of piecewise segments. Second, use a subgradient step-size update rule that does not depend on  $\gamma$ .

### 6.5.3 Convergence of algorithm 2: Bundling

The number of PH iterations required for each subgradient iteration,  $\overline{iter}$ , can be large to force PH to converge. Although each iteration involves only a single binary variable, this can make the PH algorithm computationally inefficient. In our computational experiments, PH does not converge using the criteria in Step 7 of Algorithm 2, even with  $\overline{iter} = 100$ . As the number of scenarios grows, the time to solve the MIQCPs in Step 6 grows as well; solving MIQCPs can be significantly more challenging than solving MIPs. However, nearly always the best upper bound,  $PH_{UB}$ , is obtained in the first iteration itself. To this end, we begin our discussion by reporting results obtained by setting  $\overline{iter} = 1$ ; i.e., without even entering the `while` loop of Algorithm 2. Table 6 reports these results. The PH upper bounds are on average 10% (and, at most 11.7%) away from the optimal Lagrangian values (the fourth column of Table 5). For instances that are solved optimally in Table 2, the  $PH_{UB}$  values are on average 10.9% (and, at most 11.7%) away from the true optimal solutions. We obtain these very quickly, see column “Time” in Table 6; the 1200 scenario instances are solved in under half a minute.

However, if Step 4 of Algorithm 1 is replaced with Algorithm 2 without the `while` loop, the quality of the Lagrangian relaxation scheme deteriorates significantly due to the cascading effect we describe in Sect. 6.5.2. Again, there is a tradeoff between computational effort required (a large value for  $\overline{iter}$ ) versus the quality of the solution (a value close to zero for  $\gamma$ ). To this end, we investigate the following enhancement in the PH algorithm.

Instead of solving individual scenario sub-problems, we bundle together several scenarios and solve this “mini-extensive” form problem as our new sub-problem. Bundling scenarios

**Table 6** Computational results for model (4) using Algorithm 2 with  $\overline{iter} = 1$ .  $PH_{UB}$  refers to the upper bound obtained using the progressive hedging algorithm. See Sect. 4 for details

$ \Omega $	$\varepsilon$	$PH_{UB}$	Time
150	0.05	317.24	3
	0.01	309.33	3
300	0.05	320.29	5
	0.01	311.36	5
450	0.05	319.24	8
	0.01	310.32	8
600	0.05	318.60	10
	0.01	309.93	10
900	0.05	317.96	5
	0.01	308.99	16
1200	0.05	317.37	15
	0.01	308.56	23

**Table 7** Computational results for model (2) using Algorithm 2 with Algorithm 1 using bundling. Here, “size” indicates the size of a bundle, “LR” indicates the Lagrangian relaxation bound obtained using the PH algorithm, “iterations” indicates the number of completed LR iterations, and “Gap” is the relative gap between the LB column of Table 5 and the “LR”

$ \Omega $	$\varepsilon$	size = 100				size = 150			
		LR	Gap	Time	iterations	LR	Gap	Time	iterations
150	0.05	289.32	3.1%	79	10	284.80	1.6%	68	10
	0.01	280.79	2.8%	115	6	273.01	0.0%	9	3
300	0.05	293.94	3.6%	369	10	293.55	3.5%	251	10
	0.01	284.18	2.1%	95	2	281.66	1.2%	242	6
450	0.05	293.39	4.6%	537	10	292.26	4.3%	487	10
	0.01	283.43	4.8%	229	2	280.31	3.7%	299	10
600	0.05	292.50	5.8%	576	10	291.47	5.5%	279	10
	0.01	282.23	2.7%	203	1	279.30	1.7%	409	10
900	0.05	295.06	5.9%	676	4	292.47	5.0%	870	4
	0.01	283.70	2.8%	611	2	281.14	1.9%	613	2
1200	0.05	290.78	4.7%	781	2	289.21	4.2%	896	2
	0.01	277.78	3.5%	887	2	277.61	3.5%	747	2

could offer the advantage of speeding convergence of PH, but with increased computational effort per PH iteration. For details on implementation of bundling in progressive hedging, see, e.g. [16]. Further, the bounds obtained in Step 6 of Algorithm 2 also extend to the case with bundling [16]. Finally, we emphasize again that since the PH algorithm is a heuristic for solving model (4), the bounds obtained would be worse than those obtained using a naive solution method (Table 5); however, we are interested in the tradeoff between speed and quality.

Table 7 compares the results of our experiments using two different bundles. We use Algorithm 1 with Step 3 substituted by Algorithm 2. Further, to see the tradeoff between the computational time and the value of the optimal solution, we set a maximum time limit of 900 seconds. When the completion of a LR iteration would have resulted in the total time

**Table 8** Comparison of different upper bounds for model (2). “LP” denotes the linear programming relaxation, “LR” denotes the Lagrangian relaxation of model (4), and “LR-PH” denotes the Lagrangian relaxation using progressive hedging algorithm with bundling. The “Gap” columns denote the optimality gaps from  $\max\{z^*, LB\}$ ; see, Fig. 1 for the notation. The “LR-PH” has a maximum time limit of 900 seconds. For details, see Sect. 6

$\Omega$	$\varepsilon$	LP		LR		LR-PH	
		Gap	Time	Gap	Time	Gap	Time
150	0.05	6.6%	1	1.4%	56	1.4%	68
	0.01	4.5%	1	–	8	–	9
300	0.05	6.8%	1	2.7%	272	3.3%	251
	0.01	4.4%	1	0.4%	210	1.1%	242
450	0.05	6.7%	2	2.5%	980	2.9%	487
	0.01	3.8%	2	–	660	0.5%	299
600	0.05	7.5%	3	3.3%	1936	3.7%	279
	0.01	4.2%	3	–	1581	0.6%	409
900	0.05	8.0%	9	3.8%	2369	4.7%	870
	0.01	4.4%	9	–	2362	1.5%	613
1200	0.05	8.2%	10	2.7%	2356	4.2%	896
	0.01	5.9%	9	0.9%	2351	2.4%	747

being greater than a 900 seconds, we do not report it. For example, for the 900 scenario,  $\varepsilon = 0.01$  instance if the third iteration were to complete the total time would be greater than 900 seconds; thus, we report results only for two iterations. In this sense, our results are conservative. For the 150 scenario case, there is one bundle of size 100 and one of size 50. For the 450 scenario case, there are four bundles of size 100 and one of size 50. In all of the 12 instances, the 150 size bundles lead to better (lower) Lagrangian bounds than the 100 size bundles. However, there is no consistent trend in the time required for completion. For example, the 450 scenario instance with  $\varepsilon = 0.05$  took lesser time for the 150 bundle case than the 100 bundle case; but, the effect is the opposite for the 450 scenario instance with  $\varepsilon = 0.01$ .

### 6.5.4 Summary of bounds

From the preceding discussions, we have two valid upper bounds for model (2) available from plain LR and LR with PH. A third bound is obtained by solving the LP relaxation of model (2). We summarize these bounds in Table 8. The “Gap” here is the optimality gap of the bound from  $\max\{z^*, LB\}$ . The LP has a gap of 5.9% on average. From the LR column, we note that four of the twelve instances are solved to optimality—but this optimality is not verifiable (see Table 5). Imposing a reduced time limit and solving the LR with the bundled PH algorithm increases this optimality gap by at most 1.5% (with bundles of 150 scenarios). This again indicates that the PH algorithm achieves good bounds in fairly early iterations.

## 7 A generalization for multiple generators

### 7.1 Notation

*Indices and Sets:*

$t \in T$             Set of hours;  $\{1, 2, \dots, |T|\}$

$\omega \in \Omega$	set of scenarios; $\{\omega_1, \omega_2, \dots, \omega_{ \Omega }\}$
$s \in S$	Set of batteries
$g \in G$	Set of PV generators
$b \in B$	Set of buses; $b_{gr} \in B$ is the bus connected to the larger grid
$l \in L$	Set of branches or lines
$s \in S(b)$	Set of batteries at bus $b$
$g \in G(b)$	Set of PV generators at bus $b$
$l \in \delta^+(b)$	Set of lines to bus $b$
$l \in \delta^-(b)$	Set of lines from bus $b$

*Parameters: First Stage*

$\eta_s$	Efficiency of battery $s$ , $0 < \eta_s < 1$
$\overline{X}_s$	Maximum energy that can be stored in battery $s$ [kWh]
$\underline{X}_s$	Minimum energy that must remain in battery $s$ [kWh]
$R_t$	Marginal revenue earned at hour $t$ [\$/kWh]
$C_{c,s}$	Operational cost of charging battery $s$ [\$/kWh]
$C_{d,s}$	Operational cost of discharging battery $s$ [\$/kWh]
$\overline{P}_s$	Maximum charging rate of battery $s$ [kW]
$\overline{Q}_s$	Maximum discharging rate of battery $s$ [kW]
$L_{b,t}$	Load at bus $b$ at hour $t$ [kW]
$\overline{F}_l$	Thermal limit on line $l$ at hour $t$ [kW]
$\Delta$	Time step [h]

*Parameters: Second Stage*

$s_{g,t}^\omega$	Solar power available at hour $t$ from generator $g$ under scenario $\omega$ [kW]
------------------	---

*Decision Variables: First Stage*

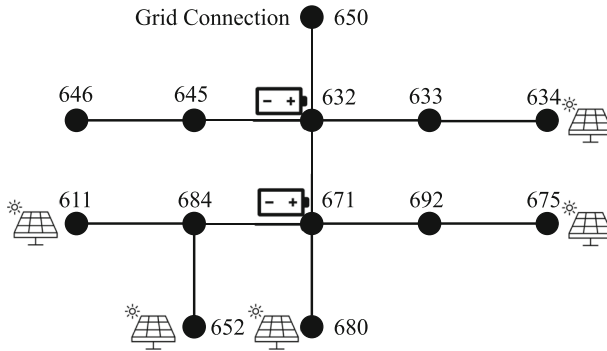
$y_t$	Power promised to deliver (or withdraw, if negative) at hour $t$ [kW]
-------	---

*Decision Variables: Second Stage*

$f_{l,t}^\omega$	Power flow on line $l$ during hour $t$ in scenario $\omega$ [kW]
$p_{s,t}^\omega$	Power charged to battery $s$ during hour $t$ in scenario $\omega$ [kW]
$q_{s,t}^\omega$	Power discharged from battery $s$ during hour $t$ in scenario $\omega$ [kW]
$x_{s,t}^\omega$	Energy stored in battery $s$ at hour $t$ in scenario $\omega$ [kWh]
$w_{s,t}^\omega$	Binary variable; 1 if battery $s$ is charging at hour $t$ and 0 if discharging
$u_t^\omega$	Power served to the grid at hour $t$ in scenario $\omega$ [kW]

## 7.2 Background

In this section, we present a proof-of-concept for a generalization of the models and analysis presented in the preceding sections of this article. To this end, we consider a microgrid with several photovoltaic units and two storage units. To build our test case, we consider a single-phase simplification of the IEEE 13-bus feeder [25,53]. We adopt a network flow approximation to model the power flow; this is equivalent to the commonly used “DC approximation” of power flow as the network considered is acyclic or *radial*, as is common for distribution systems. While this is a loose approximation [11], our aim is to demonstrate how the techniques described above can be applied in a different context.



**Fig. 4** The modified IEEE 13-bus feeder with two batteries and five PV units

We modify the IEEE 13-bus feeder system as follows. We place PV units at these five buses: 652, 680, 675, 611, 634. We assume the solar power available from the PV units at these buses are, respectively, 1, 2, 3, 4, 5 times  $\frac{1}{5} s_t^\omega$ , where  $s_t^\omega$  is the solar power data used in the rest of this article. We also place two storage devices: one at bus 632 and another at bus 671. The interface to the larger grid is at bus 650, which we denote at  $b_{gr}$ , and we buy and sell energy from this bus. Figure 4 gives a topological representation of the modified IEEE 13-bus feeder, with the placements of PV and battery storage devices. Further, we reduce the three-phase system to a single phase and set the line capacities,  $\bar{F}_l$ , to eight times the values in the IEEE 13-bus feeder system. The system is infeasible for lower values of the thermal line limit, and indeed for the computational instances we present in Sect. 7.3 this thermal limit is almost reached. Further, since this system does not provide load values indexed by time, we scale the load by computing load factors from [24, Table 3], to achieve a temporal resolution. Finally, we set  $\bar{X}_s$  to ten times the value in the rest of this article for each storage device  $s$ .

We modify model (1) as follows:

$$\max \sum_{t \in T} R_t \Delta y_t - \mathbb{E} \left[ \sum_{s \in S} (C_{c,s} \Delta p_{s,t}^\omega + C_{d,s} \Delta q_{s,t}^\omega) \right] \tag{6a}$$

$$\text{s.t. } \mathbb{P}(y_t \leq u_t^\omega, \forall t \in T) \geq 1 - \varepsilon \tag{6b}$$

$$x_{s,t+1}^\omega = x_{s,t}^\omega + \eta_s \Delta p_{s,t}^\omega - \frac{1}{\eta_s} \Delta q_{s,t}^\omega, \quad \forall s \in S, \forall t = 1, 2, \dots, |T| - 1, \forall \omega \in \Omega \tag{6c}$$

$$p_{s,t}^\omega \leq \bar{P}_s w_{s,t}^\omega, \quad \forall s \in S, \forall t \in T, \forall \omega \in \Omega \tag{6d}$$

$$q_{s,t}^\omega \leq \bar{Q}_s (1 - w_{s,t}^\omega), \quad \forall s \in S, \forall t \in T, \forall \omega \in \Omega \tag{6e}$$

$$\underline{X}_s \leq x_{s,t}^\omega \leq \bar{X}_s, \quad \forall s \in S, \forall t \in T, \forall \omega \in \Omega \tag{6f}$$

$$w_{s,t}^\omega \in \{0, 1\}, \quad \forall s \in S, \forall t \in T, \forall \omega \in \Omega \tag{6g}$$

$$\sum_{l \in \delta^+(b)} f_{l,t}^\omega - \sum_{l \in \delta^-(b)} f_{l,t}^\omega + \sum_{s \in S(b)} (p_{s,t}^\omega - q_{s,t}^\omega) + \sum_{g \in G(b)} s_{g,t}^\omega = L_{b,t}, \quad \forall b \in B \setminus \{b_{gr}\}, \forall t \in T, \forall \omega \in \Omega \tag{6h}$$

$$\sum_{l \in \delta^+(b_{gr})} f_{l,t}^\omega - \sum_{l \in \delta^-(b_{gr})} f_{l,t}^\omega = u_t^\omega, \quad \forall t \in T, \forall \omega \in \Omega \tag{6i}$$

$$-\bar{F}_l \leq f_{l,t}^\omega \leq \bar{F}_l, \quad \forall l \in L, \forall t \in T, \forall \omega \in \Omega \tag{6j}$$

$$p_{s,t}^\omega, q_{s,t}^\omega \geq 0, \quad \forall s \in S, \forall t \in T, \forall \omega \in \Omega. \tag{6k}$$

*Boundary condition:*  $x_{s,1}^\omega = x_0, \forall s \in S, \omega \in \Omega$ .

Constraint (6b) ensures we meet the promised load (or, withdraw no more than promised) for the entire time horizon with probability  $1 - \epsilon$ . Constraints (6c)–(6g), (6k) are exactly (1c)–(1h) except applied to multiple storage devices. Constraint (6h) ensures the load, generation, and flows are balanced at every bus except the grid-connected bus; constraint (6i) balances the flow to/from the grid-connected bus and the power to/from the grid  $u_t^\omega$ .

Model (6) permits a Lagrangian decomposition analogous to that demonstrated in Section 3, as follows.

$$\begin{aligned}
 L(\lambda) = \max \quad & \sum_{t \in T} \left( R_t \Delta y_t - \mathbb{E} \left[ \sum_{s \in S} (C_{c,s} \Delta p_{s,t}^\omega + C_{d,s} \Delta q_{s,t}^\omega) \right] \right) + \lambda (\lfloor N \epsilon \rfloor - \sum_{\omega \in \Omega} z^\omega) \\
 \text{s.t.} \quad & (6c) - (6k) \tag{7a} \\
 & y_t - u_t^\omega \leq z^\omega M_t^\omega, \quad \forall t \in T, \forall \omega \in \Omega \tag{7b} \\
 & z^\omega \in \{0, 1\}. \quad \forall \omega \in \Omega. \tag{7c}
 \end{aligned}$$

Similar to Corollary 1, a sufficiently large value for the big- $M$  for model (6) and model (7) is given by:  $M_t^\omega = \max_{t \in T, \omega \in \Omega} u_t^\omega = 2|L| \max_{t \in L} \bar{F}_t, \forall t \in T, \omega \in \Omega$ . By including constraint (6g), we disallow simultaneous charging and discharging of the battery. Thus, unlike model (4), model (7) includes additional binary variables; for details on when this restriction can be relaxed, see Proposition 1 and [17].

### 7.3 Computational results

Unlike Sect. 6, an instance of model (6) with  $|\Omega| = 300$  and  $\epsilon = 0.05$  is hard to solve. To demonstrate the generalization of the models we presented earlier, we thus use  $|\Omega| = 150$  and  $\epsilon = 0.05$ .

Analogous to Table 3, we present the profile of the optimal solution in Table 9. Between 20:00 and 08:59, when solar power is absent or in low quantities, the system buys energy from the main grid. In this sense, the solution is comparable to that in Sect. 6.2.1 where we do not complete any transactions between 00:00 and 6:59. Between 09:00 and 20:00, when solar power is more available, we sell energy to achieve a total profit of \$316.22. Since the data values are different than the ones we used before, the optimal objective function values are not directly comparable.

As the computational experiments are directly analogous to the ones we presented in detail in Sect. 6, here we only provide computational results for the hardest instance:  $|\Omega| = 1200$  and  $\epsilon = 0.05$ . We again compare the computational effort of the naive solution to the Lagrangian decomposition. The optimal solution for model (6) using the naive solution method is in the range [245.91, 66,726.55]—a MIP gap of 99.6%—with a new time limit three times that of the previously employed time limit (i.e., new time limit is 7200 seconds). This large MIP gap indicates the significantly larger computational effort required as opposed to Sect. 6.2. Next, we solve model (7). To this end, we first run the lower bounding heuristic of Sect. 3; we obtain LB = 247.91. The LP relaxation is 66,973.1. Seven iterations of the Lagrangian decomposition are completed and we obtain LR= 48,486.7, resulting in a gap of 99.5%. This suggests little benefit for a naive implementation of LR when multiple generators are present. Future efforts could concentrate on finding smaller values of big- $M$  and tighter reformulations of model (6).

**Table 9** Energy bought/sold at hour  $t$  for model (7) for  $|\Omega| = 150$  and  $\varepsilon = 0.05$ . A positive (negative) value indicates energy is sold (bought)

Hour	Energy bought/sold (kWh)	Hour	Energy bought/sold (kWh)
0:00	−871.50	12:00	2354.65
1:00	−819.20	13:00	2584.87
2:00	−754.99	14:00	2702.85
3:00	−657.62	15:00	2197.56
4:00	−600.51	16:00	1960.89
5:00	−512.42	17:00	1132.96
6:00	−446.21	18:00	537.52
7:00	−383.02	19:00	131.76
8:00	−74.15	20:00	−195.33
9:00	1532.04	21:00	−132.22
10:00	1421.08	22:00	−983.43
11:00	2273.46	23:00	−931.16

## 8 Conclusion

We developed a stochastic optimization model to schedule a standalone hybrid system consisting of battery storage with solar power. We used a chance-constrained formulation to ensure highly reliable operations under the uncertainty of solar power. The model makes charging/discharging decisions dictated by a mix of economics and the available solar power. For a few hundred scenarios, the model is relatively tractable benefiting from a strong LP relaxation. For larger number of scenarios, we demonstrate the applicability of a simple Lagrangian relaxation procedure. The procedure is remarkably effective, and tightens the optimality gap when a naive solution method does not. Finally, we present a heuristic, based on the progressive hedging algorithm, that we couple with the Lagrangian relaxation. We find significant benefit when scenarios are bundled and solved as one in the progressive hedging iterations; however, when solved plainly the PH heuristic does not achieve good quality solutions. We conclude with an extension to a larger system with multiple storage devices and batteries connected to form a microgrid. Future work could examine interconnected updates of the subgradient algorithm with the progressive hedging algorithm.

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

**Acknowledgements** Bismark Singh thanks David Pozo for discussions on the early stages of this manuscript and Frederik Fiand for assistance with the GUSS implementations. Parts of this work were completed while the authors were affiliated with Sandia National Laboratories, and supported by Sandia’s Laboratory Directed Research and Development (LDRD) program. The authors thank Sandia for their time.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory

regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Agnew, N., Agnew, R., Rasmussen, J., Smith, K.: An application of chance constrained programming to portfolio selection in a casualty insurance firm. *Manag. Sci.* **15**(10), 512 (1969)
2. Ahmed, S.: Two-stage stochastic integer programming: A brief introduction. Wiley Encyclopedia of Operations Research and Management Science (2010)
3. Ahmed, S., Luedtke, J., Song, Y., Xie, W.: Nonanticipative duality, relaxations, and formulations for chance-constrained stochastic programs. *Math. Program.* **162**(1–2), 51–81 (2017)
4. Ahmed, S., Shapiro, A.: Solving chance-constrained stochastic programs via sampling and integer programming. In: State-of-the-art decision-making tools in the information-intensive age, pp. 261–269. *Inform.* (2008)
5. Bahramirad, S., Reder, W., Khodaei, A.: Reliability-constrained optimal sizing of energy storage system in a microgrid. *IEEE Trans. Smart Grid* **3**(4), 2056–2062 (2012)
6. Bai, X., Sun, J., Sun, X., Zheng, X.: An alternating direction method for chance-constrained optimization problems with discrete distributions. Technical report, School of Management, Fudan University (2012)
7. Bekele, G., Palm, B.: Feasibility study for a standalone solar-wind-based hybrid energy system for application in Ethiopia. *Appl. Energy* **87**(2), 487–495 (2010)
8. Bhuiyan, M., Asgar, M.A., Mazumder, R., Hussain, M.: Economic evaluation of a stand-alone residential photovoltaic power system in Bangladesh. *Renew. Energy* **21**(3–4), 403–410 (2000)
9. Bilal, B.O., Sambou, V., Ndiaye, P., Kébé, C., Ndong, M.: Optimal design of a hybrid solar-wind-battery system using the minimization of the annualized cost system and the minimization of the loss of power supply probability (LPSP). *Renew. Energy* **35**(10), 2388–2390 (2010)
10. Bussieck, M.R., Ferris, M.C., Lohmann, T.: GUSS: Solving collections of data related models within GAMS. In: Algebraic Modeling Systems, pp. 35–56. Springer (2012)
11. Cain, M.B., O’Neill, R.P., Castillo, A.: History of optimal power flow and formulations (2012)
12. Carrión, M., Arroyo, J.M.: A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Trans. Power Syst.* **21**(3), 1371–1378 (2006)
13. Clare, G., Richards, A.: Air traffic flow management under uncertainty: Application of chance constraints. In: Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems, pp. 20–26. IRIT Press (2012)
14. Conejo, A.J., Arroyo, J.M., Contreras, J., Villamor, F.A.: Self-scheduling of a hydro producer in a pool-based electricity market. *IEEE Trans. Power Syst.* **17**(4), 1265–1272 (2002)
15. Electric Reliability Council of Texas, Inc (ERCOT): Market prices. <http://www.ercot.com/mktinfo/prices> (2016). Accessed: Feb 17, 2019
16. Gade, D., Hackebeil, G., Ryan, S.M., Watson, J.P., Wets, R.J.B., Woodruff, D.L.: Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Math. Program.* **157**(1), 47–67 (2016)
17. Garifi, K., Baker, K., Christensen, D., Touri, B.: Control of energy storage in home energy management systems: Non-simultaneous charging and discharging guarantees. arXiv preprint [arXiv:1805.00100](https://arxiv.org/abs/1805.00100) (2018)
18. Giraud, F., Salameh, Z.M.: Steady-state performance of a grid-connected rooftop hybrid wind-photovoltaic power system with battery storage. *IEEE Trans. Energy Convers.* **16**(1), 1–7 (2001)
19. Golestaneh, F., Gooi, H.B., Pinson, P.: Generation and evaluation of space-time trajectories of photovoltaic power. *Appl. Energy* **176**, 80–91 (2016)
20. Guan, X., Xu, Z., Jia, Q.S.: Energy-efficient buildings facilitated by microgrid. *IEEE Trans. Smart Grid* **1**(3), 243–252 (2010)
21. Held, M., Wolfe, P., Crowder, H.P.: Validation of subgradient optimization. *Math. Program.* **6**(1), 62–88 (1974)
22. Helseth, A.: Stochastic network constrained hydro-thermal scheduling using a linearized progressive hedging algorithm. *Energy Syst.* **7**(4), 585–600 (2016)
23. Henrion, R., Strugarek, C.: Convexity of chance constraints with independent random variables. *Computat. Optim. Appl.* **41**(2), 263–276 (2008)
24. Hsu, C., Tzeng, Y., Chen, C., Cho, M.: Distribution feeder loss analysis by using an artificial neural network. *Electric Power Syst. Res.* **34**(2), 85–90 (1995). [https://doi.org/10.1016/0378-7796\(95\)00959-x](https://doi.org/10.1016/0378-7796(95)00959-x)
25. IEEE PES AMPS DSAS Test Feeder Working Group: 1992 Test Feeder Cases. <https://site.ieee.org/pes-testfeeders/resources/> (2020). Accessed: August 5, 2020



26. Ito, M., Kato, K., Sugihara, H., Kichimi, T., Song, J., Kurokawa, K.: A preliminary study on potential for very large-scale photovoltaic power generation (VLS-PV) system in the Gobi desert from economic and environmental viewpoints. *Solar Energy Mater. Solar Cells* **75**(3–4), 507–517 (2003)
27. Jenkins, D., Fletcher, J., Kane, D.: Lifetime prediction and sizing of lead-acid batteries for microgeneration storage applications. *IET Renew. Power Gener.* **2**(3), 191–200 (2008)
28. Kargarian, A., Fu, Y., Wu, H.: Chance-constrained system of systems based operation of power systems. *IEEE Trans. Power Syst.* **31**(5), 3404–3413 (2016)
29. Knueven, B., Ostrowski, J., Ollis, B., Irminger, P., Starke, M., Herron, A., King, D., Xiao, B., Xue, Y., Karlson, P., et al.: Economic feasibility analysis and operational testing of a community energy storage system. In: *Energy Conversion Congress and Exposition (ECCE)*, 2016 IEEE, pp. 1–5. IEEE (2016)
30. Krishnan, V., McCalley, J.D.: Building foresight in long-term infrastructure planning using end-effect mitigation models. *IEEE Syst. J.* **11**(4), 2040–2051 (2017). <https://doi.org/10.1109/jsyst.2015.2410758>
31. Lasseter, R.H.: Microgrids. In: *Power Engineering Society Winter Meeting*, 2002. IEEE, vol. 1, pp. 305–308. IEEE (2002)
32. Li, S.: An insurance and investment portfolio model using chance constrained programming. *Omega* **23**(5), 577–585 (1995)
33. Lima, R.M., Conejo, A.J., Langodan, S., Hoteit, I., Knio, O.M.: Risk-averse formulations and methods for a virtual power plant. *Comput. Oper. Res.* **96**, 350–373 (2018)
34. Liu, X., Küçükyavuz, S., Luedtke, J.: Decomposition algorithms for two-stage chance-constrained programs. *Math. Program.* **157**(1), 219–243 (2016)
35. Løkketangen, A., Woodruff, D.L.: Progressive hedging and tabu search applied to mixed integer (0, 1) multistage stochastic programming. *J. Heuristics* **2**(2), 111–128 (1996)
36. Luedtke, J.: A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Math. Program.* **146**(1–2), 219–244 (2014)
37. Luedtke, J., Ahmed, S., Nemhauser, G.L.: An integer programming approach for linear programs with probabilistic constraints. *Math. Program.* **122**(2), 247–272 (2010)
38. Ma, T., Yang, H., Lu, L.: A feasibility study of a stand-alone hybrid solar-wind-battery system for a remote island. *Appl. Energy* **121**, 149–158 (2014)
39. Ma, T., Yang, H., Lu, L., Peng, J.: Technical feasibility study on a standalone hybrid solar-wind system with pumped hydro storage for a remote island in Hong Kong. *Renew. Energy* **69**, 7–15 (2014)
40. Morales, J.M., Conejo, A.J., Madsen, H., Pinson, P., Zugno, M.: Virtual power plants. In: *Integrating Renewables in Electricity Markets*, pp. 243–287. Springer (2014)
41. Ostrowski, J., Anjos, M.F., Vannelli, A.: Tight mixed integer linear programming formulations for the unit commitment problem. *IEEE Trans. Power Syst.* **27**(1), 39–46 (2012)
42. Ozturk, U.A., Mazumdar, M., Norman, B.A.: A solution to the stochastic unit commitment problem using chance constrained programming. *IEEE Trans. Power Syst.* **19**(3), 1589–1598 (2004)
43. Parida, B., Iniyar, S., Goic, R.: A review of solar photovoltaic technologies. *Renew. Sustain. Energy Rev.* **15**(3), 1625–1636 (2011)
44. Patel, M.R.: *Wind and solar power systems: design, analysis, and operation*. CRC Press, Boca Raton (2005)
45. Polyak, B.T.: A general method for solving extremal problems. In: *Doklady Akademii Nauk*, vol. 174, pp. 33–36. Russian Academy of Sciences (1967)
46. PowerTech Systems: Lithium-ion vs lead-acid cost analysis. <https://www.powertechsystems.eu/home/tech-corner/lithium-ion-vs-lead-acid-cost-analysis/> (2015). Accessed: May 5, 2019
47. Prékopa, A.: *Stochastic Programming*. Kluwer Academic Publishers, Amsterdam (1995)
48. Rockafellar, R.T., Wets, R.J.B.: Scenarios and policy aggregation in optimization under uncertainty. *Math. Oper. Res.* **16**(1), 119–147 (1991)
49. Rosenthal, R.E.: *GAMS—A User’s Guide*. ALS-NSCORT (2004)
50. Ruszczyński, A., Shapiro, A.: Optimization of convex risk functions. *Math. Oper. Res.* **31**(3), 433–452 (2006)
51. dos Santos, M.L., da Silva, E.L., Finardi, E.C., Gonçalves, R.E.: Practical aspects in solving the medium-term operation planning problem of hydrothermal power systems by using the progressive hedging method. *Int. J. Electric. Power Energy Syst.* **31**(9), 546–552 (2009)
52. Saurorja = Solar Energy: Lead acid is the cheapest battery: Conditions apply. <https://saurorja.org/2011/08/30/lead-acid-is-the-cheapest-battery-conditions-apply/> (2011)
53. Schneider, K.P., Mather, B.A., Pal, B.C., Ten, C.W., Shirek, G.J., Zhu, H., Fuller, J.C., Pereira, J.L.R., Ochoa, L.F., de Araujo, L.R., Dugan, R.C., Matthias, S., Paudyal, S., McDermott, T.E., Kersting, W.: Analytic considerations and design basis for the IEEE distribution test feeders. *IEEE Trans. Power Syst.* **33**(3), 3181–3188 (2018). <https://doi.org/10.1109/tpwrs.2017.2760011>

54. Shor, N.: The rate of convergence of the generalized gradient descent method. *Cybern. Syst. Anal.* **4**(3), 79–80 (1968)
55. Singh, B.: Optimal spatiotemporal resource allocation in public health and renewable energy. Ph.D. thesis, The University of Texas at Austin (2016)
56. Singh, B., Morton, D.P., Santoso, S.: An adaptive model with joint chance constraints for a hybrid wind-conventional generator system. *Computational Management Science* pp. 1–20 (2017)
57. Singh, B., Pozo, D.: A guide to solar power forecasting using ARMA models. In: 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), pp. 1–4 (2019). 10.1109/ISGTEurope.2019.8905430
58. Veliz, F.B., Watson, J.P., Weintraub, A., Wets, R.J.B., Woodruff, D.L.: Stochastic optimization models in forest planning: a progressive hedging solution approach. *Ann. Oper. Res.* **232**(1), 259–274 (2015)
59. Waller, S.T., Ziliaskopoulos, A.K.: A chance-constrained based stochastic dynamic traffic assignment model: analysis, formulation and solution algorithms. *Transp. Res. Part C Emerg. Technol.* **14**(6), 418–427 (2006)
60. Watson, J.P., Wets, R.J., Woodruff, D.L.: Scalable heuristics for a class of chance-constrained stochastic programs. *INFORMS J Comput* **22**(4), 543–554 (2010)
61. Watson, J.P., Woodruff, D.L.: Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Comput. Manag. Sci.* **8**(4), 355–370 (2011)
62. Wu, H., Shahidehpour, M., Li, Z., Tian, W.: Chance-constrained day-ahead scheduling in stochastic power system operation. *IEEE Trans. Power Syst.* **29**(4), 1583–1591 (2014)
63. Wu, L.: A tighter piecewise linear approximation of quadratic cost curves for unit commitment problems. *IEEE Trans. Power Syst.* **26**(4), 2581–2583 (2011)
64. Xu, H., Caramanis, C., Mannor, S.: Optimization under probabilistic envelope constraints. *Oper. Res.* **60**(3), 682–699 (2012)
65. Yang, H., Zhou, W., Lu, L., Fang, Z.: Optimal sizing method for stand-alone hybrid solar-wind system with LPSP technology by using genetic algorithm. *Solar Energy* **82**(4), 354–367 (2008)
66. Yunjun, H., Xiangdong, Y., Dan, W.: Lagrangian relaxation based feasible solution algorithm. In: 2012 24th Chinese Control and Decision Conference (CCDC), pp. 875–878. IEEE (2012)
67. Zhang, D., Shah, N., Papageorgiou, L.G.: Efficient energy consumption and operation management in a smart building with microgrid. *Energy Convers. Manag.* **74**, 209–222 (2013)
68. Zhao, B., Zhang, X., Chen, J., Wang, C., Guo, L.: Operation optimization of standalone microgrids considering lifetime characteristics of battery energy storage system. *IEEE Trans. Sustain. Energy* **4**(4), 934–943 (2013)
69. Zhao, B., Zhang, X., Li, P., Wang, K., Xue, M., Wang, C.: Optimal sizing, operating strategy and operational experience of a stand-alone microgrid on Dongfushan island. *Appl. Energy* **113**, 1656–1666 (2014)
70. Zhou, W., Lou, C., Li, Z., Lu, L., Yang, H.: Current status of research on optimum sizing of stand-alone hybrid solar-wind power generation systems. *Appl. Energy* **87**(2), 380–389 (2010)
71. Zhuang, F., Galiana, F.D.: Towards a more rigorous and practical unit commitment by lagrangian relaxation. *IEEE Trans. Power Syst.* **3**(2), 763–773 (1988)