COMBINING PROGRESSIVE HEDGING WITH A FRANK-WOLFE METHOD TO COMPUTE LAGRANGIAN DUAL BOUNDS IN STOCHASTIC MIXED-INTEGER PROGRAMMING[*]

NATASHIA BOLAND[†], JEFFREY CHRISTIANSEN[‡], BRIAN DANDURAND[‡], ANDREW EBERHARD[‡], JEFF LINDEROTH[§], JAMES LUEDTKE[§], AND FABRICIO OLIVEIRA[‡]

Abstract. We present a new primal-dual algorithm for computing the value of the Lagrangian dual of a stochastic mixed-integer program (SMIP) formed by relaxing its nonanticipativity constraints. This dual is widely used in decomposition methods for the solution of SMIPs. The algorithm relies on the well-known progressive hedging method, but unlike previous progressive hedging approaches for SMIP, our algorithm can be shown to converge to the optimal Lagrangian dual value. The key improvement in the new algorithm is an inner loop of optimized linearization steps, similar to those taken in the classical Frank-Wolfe method. Numerical results demonstrate that our new algorithm empirically outperforms the standard implementation of progressive hedging for obtaining bounds in SMIP.

Key words. mixed-integer stochastic programming, Lagrangian duality, progressive hedging, Frank-Wolfe method

1. Introduction.

Stochastic programming with recourse provides a framework for modeling problems where decisions are made in stages. Between stages, some uncertainty in the problem parameters is unveiled, and decisions in subsequent stages may depend on the outcome of this uncertainty. When some decisions are modeled using discrete variables, the problem is known as a Stochastic Mixed-Integer Programming (SMIP) problem. The ability to simultaneously model uncertainty and discrete decisions make SMIP a powerful modeling paradigm for applications. Important applications employing SMIP models include unit commitment and hydro-thermal generation scheduling [25, 34], military operations [32], vaccination planning [28, 35], air traffic flow management [4], forestry management and forest fire response [5, 27], and supply chain and logistics planning [19, 21]. However, the combination of uncertainty and discreteness makes this class of problems extremely challenging from a computational perspective. In this paper, we present a new and effective algorithm for computing lower bounds that arise from a Lagrangian-relaxation approach.

The mathematical statement of a two-stage SMIP is

$$\zeta^{SMIP} := \min_{x} \left\{ c^\top x + \mathcal{Q}(x) \ : \ x \in X \right\}, \tag{1}$$

where the vector $c \in \mathbb{R}^{n_x}$ is known, and X is a mixed-integer linear set consisting of linear constraints and integer restrictions on some components of x. The function $\mathcal{Q} : \mathbb{R}^{n_x} \mapsto \mathbb{R}$ is the expected recourse value

$$\mathcal{Q}(x) := \mathbb{E}_\xi \left[\min_y \left\{ q(\xi)^\top y \ : \ W(\xi) y = h(\xi) - T(\xi) x, y \in Y(\xi) \right\} \right].$$

[*] The work of authors Boland, Christiansen, Dandurand, Eberhard, Linderoth, and Oliveira was supported in part or in whole by the Australian Research Council (ARC) grant ARC DP140100985. The work of authors Linderoth and Luedtke was supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract number DE-AC02-06CH11357.

[†] Georgia Institute of Technology, Atlanta, Georgia, USA (natashia.boland@gmail.com)

[‡] RMIT University, Melbourne, Victoria, Australia (s3507717@student.rmit.edu.au, brian.dandurand@rmit.edu.au, andy.eberhard@rmit.edu.au, fabricio.oliveira@rmit.edu.au)

[§] Department of Industrial and Systems Engineering, Wisconsin Institutes of Discovery, University of Wisconsin-Madison, Madison, Wisconsin, USA (linderoth@wisc.edu, jim.luedtke@wisc.edu)

We assume that the random variable ξ is taken from a discrete distribution indexed by the finite set $\mathcal{S}$, consisting of the realizations, $\xi_1, \dots, \xi_{|\mathcal{S}|}$, with strictly positive corresponding probabilities of realization, $p_1, \dots, p_{|\mathcal{S}|}$. When ξ is not discrete, a finite scenario approximation can be obtained via Monte Carlo sampling [18, 23] or other methods [10, 9]. Each realization ξ_s of ξ, is called a *scenario* and encodes the realizations observed for each of the random elements $(q(\xi_s), h(\xi_s), W(\xi_s), T(\xi_s), Y(\xi_s))$. For notational brevity, we refer to this collection of random elements respectively as $(q_s, h_s, W_s, T_s, Y_s)$. For each $s \in \mathcal{S}$, the set $Y_s \subset \mathbb{R}^{n_y}$ is a mixed-integer set containing both linear constraints and integrality constraints on a subset of the variables, y_s.

The problem (1) may be reformulated as its *deterministic equivalent*

$$\zeta^{SMIP} = \min_{x,y} \left\{ \ c^\top x + \sum_{s \in \mathcal{S}} p_s q_s^\top y_s \ : \ (x, y_s) \in K_s, \forall s \in \mathcal{S} \ \right\}, \tag{2}$$

where $K_s := \{(x, y_s) \ : \ W_s y_s = h_s - T_s x, x \in X, y_s \in Y_s\}$. Problem (2) has a special structure that can be algorithmically exploited by decomposition methods. To induce a decomposable structure, scenario-dependent copies x_s for each $s \in S$ of the first-stage variable x are introduced to create the following reformulation of (2):

$$\zeta^{SMIP} = \min_{x,y,z} \left\{ \sum_{s \in \mathcal{S}} p_s (c^\top x_s + q_s^\top y_s) \ : \ (x_s, y_s) \in K_s, x_s = z, \ \forall s \in \mathcal{S}, z \in \mathbb{R}^{n_x} \right\}. \tag{3}$$

The constraints $x_s = z$, $s \in \mathcal{S}$, enforce nonanticipativity for first-stage decisions; the first-stage decisions x_s must be the same (z) for each scenario $s \in \mathcal{S}$. Applying Lagrangian relaxation to the nonanticipativity constraints in problem (3) yields the *nonanticipative Lagrangian dual function*

$$\phi(\mu) := \min_{x,y,z} \left\{ \begin{array}{c} \sum_{s \in \mathcal{S}} \left[p_s (c^\top x_s + q_s^\top y_s) + \mu_s^\top (x_s - z) \right] \ : \\ (x_s, y_s) \in K_s, \ \forall s \in \mathcal{S}, z \in \mathbb{R}^{n_x} \end{array} \right\}, \tag{4}$$

where $\mu = (\mu_1, \dots, \mu_{|\mathcal{S}|}) \in \prod_{s \in \mathcal{S}} \mathbb{R}^{n_x}$ is the vector of multipliers associated with the relaxed constraints $x_s = z$, $s \in \mathcal{S}$. By setting $\omega_s := \frac{1}{p_s} \mu_s$, (4) may be rewritten as

$$\phi(\omega) := \min_{x,y,z} \left\{ \sum_{s \in \mathcal{S}} p_s L_s(x_s, y_s, z, \omega_s) \ : \ (x_s, y_s) \in K_s, \ \forall s \in \mathcal{S}, \ z \in \mathbb{R}^{n_x} \right\}, \tag{5}$$

where

$$L_s(x_s, y_s, z, \omega_s) := c^\top x_s + q_s^\top y_s + \omega_s^\top (x_s - z).$$

Since z is unconstrained in the optimization problem in the definition (5), in order for the Lagrangian function $\phi(\omega)$ to be bounded from below, we require as a condition of dual feasibility that $\sum_{s \in \mathcal{S}} p_s \omega_s = 0$. Under this assumption, the z term vanishes, and the Lagrangian dual function (5) decomposes into separable functions,

$$\phi(\omega) = \sum_{s \in \mathcal{S}} p_s \phi_s(\omega_s), \tag{6}$$

where for each $s \in \mathcal{S}$,

$$\phi_s(\omega_s) := \min_{x,y} \left\{ (c + \omega_s)^\top x + q_s^\top y \ : \ (x, y) \in K_s \right\}. \tag{7}$$

The reformulation (6) is the basis for parallelizable approaches for computing dual bounds that are used, for example, in the dual decomposition methods developed in [8, 22].

For any choice of $\omega = (\omega_1, \ldots, \omega_{|\mathcal{S}|})$, it is well-known that the value of the Lagrangian provides a lower bound on the optimal solution to (1): $\phi(\omega) \leq \zeta^{SMIP}$. The problem of finding the best such lower bound is the *Lagrangian dual problem*:

$$\zeta^{LD} := \sup_{\omega} \left\{ \phi(\omega) \ : \ \sum_{s \in \mathcal{S}} p_s \omega_s = 0 \right\}. \tag{8}$$

The primary contribution of this work is a new and effective method for solving (8), thus enabling a practical and efficient computation of high-quality lower bounds for ζ^{SMIP}.

The function $\phi(\omega)$ is a piecewise-affine concave function, and many methods are known for maximizing such functions. These methods include the subgradient method [33], the augmented Lagrangian (AL) method [15, 29], and the alternating direction method of multipliers (ADMM) [13, 11, 7]. The subgradient method has mainly theoretical significance, since it is difficult to develop reliable and efficient step-size rules for the dual variables ω (see, e.g., Section 7.1.1 of [31]). As iterative primal-dual approaches, methods based on the AL method or ADMM are more effective in practice. However, in the context of SMIP, both methods require convexification of the constraints K_s, $s \in \mathcal{S}$ to have a meaningful theoretical support for convergence to the best lower bound value ζ^{LD}. Furthermore, both methods require the solution of additional mixed-integer linear programming (MILP) subproblems in order to recover the Lagrangian lower bounds associated with the dual values, ω [14]. ADMM has a more straightforward potential for decomposability and parallelization than the AL method, and so in this work, we develop a theoretically-supported modification of a method based on ADMM.

When specialized to the deterministic equivalent problem (2) in the context of stochastic programming, ADMM is referred to as Progressive Hedging (PH) [30, 37]. When the sets K_s, $s \in \mathcal{S}$, are convex, the limit points of the sequence of solution-multiplier pairs $\left\{((x^k, y^k, z^k), \omega^k)\right\}_{k=1}^{\infty}$ generated by PH are saddle points of the deterministic equivalent problem (2), whenever such saddle points exist. When the constraints $(x_s, y_s) \in K_s$, $s \in \mathcal{S}$, enforce nontrivial mixed-integer restrictions, the set K_s is not convex and PH becomes a heuristic approach with no guarantees of convergence [20]. Nevertheless, some measure of success, in practice, has been observed in [37] while applying PH to problems of the form (3). More recently, [14] showed that valid Lagrangian lower bounds can be calculated from the iterates of the PH algorithm when the sets K_s are not convex. However, their implementation of the algorithm does not offer any guarantee that the lower bounds will converge to the optimal value ζ^{LD}. Moreover, additional computational effort, in solving additional MILP subproblems, must be expended, in order to compute the lower bound. Our contribution is to extend the PH-based approach in [14], creating an algorithm whose lower bound values converge to ζ^{LD}, in theory, and for which lower bound calculations do not require additional computational effort. Computational results in Section 4 demonstrate that the new method outperforms the existing PH-based method, in terms of both quality of bound and efficiency of computation.

To motivate our approach, we first consider the application of PH to the following well-known primal characterization of ζ^{LD}:

$$\zeta^{LD} = \min_{x,y,z} \left\{ \sum_{s \in \mathcal{S}} p_s (c^\top x_s + q_s^\top y_s) \ : \ (x_s, y_s) \in \operatorname{conv}(K_s), \ x_s = z, \ \forall s \in \mathcal{S} \right\}, \tag{9}$$

where $\mathrm{conv}(K_s)$ denotes the convex hull of K_s for each $s \in \mathcal{S}$. (See, for example, Theorem 6.2 of [24].) The sequence of Lagrangian bounds $\{\phi(\omega^k)\}$ generated by the application of PH to (9) is known to be convergent. Thus, the value of the Lagrangian dual, ζ^{LD}, may, in theory, be computed by applying PH to (9). However, in practice, an explicit polyhedral description of $\mathrm{conv}(K_s)$, $s \in \mathcal{S}$ is, generally, not available, thus raising the issue of implementability.

The absence of such an explicit description motivates an application of a solution approach to the PH primal update step that iteratively constructs an improved inner approximation of each $\mathrm{conv}(K_s)$, $s \in \mathcal{S}$. For this purpose, we apply a solution approach to the PH primal update problem that is based on the Frank-Wolfe (FW) method [12]. Our approach has the additional benefit of providing Lagrangian bounds at no additional computational cost.

One simple, theoretically-supported integration of a FW-like method and PH is realized by having the PH primal updates computed using a method called the Simplicial Decomposition Method (SDM) [16, 36]. SDM is an extension of the FW method that makes use of progressively-improving inner approximations to each set $\mathrm{conv}(K_s)$, $s \in \mathcal{S}$. The finite optimal convergence of each application of SDM follows directly from the polyhedral structure $\mathrm{conv}(K_s)$, and the (practically reasonable) assumption that $\mathrm{conv}(K_s)$ is bounded for each $s \in \mathcal{S}$.

For computing improvements in the Lagrangian bound efficiently, convergence of SDM to the optimal solution of the subproblem is too costly and not necessary. We thus develop a modified integration whose theoretically-supported convergence analysis is based not on the optimal convergence of SDM, but rather on its ability to adequately extend the inner approximations of each $\mathrm{conv}(K_s)$, $s \in \mathcal{S}$.

The main contribution of this paper is the development, convergence analysis, and application of a new algorithm, called FW-PH, which is used to compute high-quality Lagrangian bounds for SMIPs efficiently and with a high potential for parallelization. FW-PH is efficient in that, under mild assumptions, each dual update and Lagrangian bound computation may be obtained by solving, for each $s \in \mathcal{S}$, just one MILP problem and one continuous convex quadratic problem. In contrast, each dual update of PH requires the solution of a mixed-integer quadratic programming (MIQP) subproblem for each $s \in \mathcal{S}$, and each PH Lagrangian bound computation requires the solution of one MILP subproblem for each $s \in \mathcal{S}$. In our convergence analysis, conditions are provided under which the sequence of Lagrangian bounds generated by FW-PH converges to the optimal Lagrangian bound ζ^{LD}. To the best of our knowledge, the combination of PH and FW in a manner that is theoretically supported, computationally efficient, and parallelizable is new, in spite of the convergence analyses of both PH and FW being well-developed.

This paper is organized as follows. In Section 2, we present the theoretical background of PH and a brief technical lemma regarding the inner approximations generated by SDM; this background is foundational for the proposed FW-PH method. In Section 3, we present the FW-PH method and a convergence analysis. The results of numerical experiments comparing the Lagrangian bounds computed with PH and those with FW-PH are presented in Section 4. We conclude in Section 5 with a discussion of the results obtained and with suggested directions for further research.

2. Progressive Hedging and Frank-Wolfe-Based Methods.

The Augmented Lagrangian (AL) function based on the relaxation of the nonanticipativity constraints $x_s = z$, $s \in \mathcal{S}$, is

$$L^\rho(x, y, z, \omega) := \sum_{s \in \mathcal{S}} p_s L_s^\rho(x_s, y_s, z, \omega_s),$$

where

$$L_s^\rho(x_s, y_s, z, \omega_s) := c^\top x_s + q_s^\top y_s + \omega_s^\top (x_s - z) + \frac{\rho}{2} \|x_s - z\|_2^2$$

and $\rho > 0$ is a penalty parameter. By changing the feasible region, denoted here by D_s, $s \in \mathcal{S}$, the Augmented Lagrangian can be used in a Progressive Hedging approach to solve either problem (3) or problem (9). Pseudocode for the PH algorithm is given in Algorithm 1.

Algorithm 1 PH applied to problem (3) ($D_s = K_s$) or (9) ($D_s = \text{conv}(K_s)$).

```
1: Precondition: $\sum_{s \in \mathcal{S}} p_s \omega_s^0 = 0$
2: function PH($\omega^0$, $\rho$, $k_{max}$, $\epsilon$)
3:     for $s \in \mathcal{S}$ do
4:         $(x_s^0, y_s^0) \in \text{argmin}_{x,y} \left\{ (c + \omega_s^0)^\top x + q_s^\top y \ : \ (x, y) \in D_s \right\}$
5:     end for
6:     $\phi^0 \leftarrow \sum_{s \in \mathcal{S}} p_s \left[ (c + \omega_s^0)^\top x_s^0 + q_s^\top y_s^0 \right]$
7:     $z^0 \leftarrow \sum_{s \in \mathcal{S}} p_s x_s^0$
8:     $\omega_s^1 \leftarrow \omega_s^0 + \rho(x_s^0 - z^0)$ for all $s \in \mathcal{S}$
9:     for $k = 1, \dots, k_{max}$ do
10:        for $s \in \mathcal{S}$ do
11:            $\phi_s^k \leftarrow \min_{x,y} \left\{ (c + \omega_s^k)^\top x + q_s^\top y \ : \ (x, y) \in D_s \right\}$
12:            $(x_s^k, y_s^k) \in \text{argmin}_{x,y} \left\{ L_s^\rho(x, y, z^{k-1}, \omega_s^k) \ : \ (x, y) \in D_s \right\}$
13:        end for
14:        $\phi^k \leftarrow \sum_{s \in \mathcal{S}} p_s \phi_s^k$
15:        $z^k \leftarrow \sum_{s \in \mathcal{S}} p_s x_s^k$
16:        if $\sqrt{\sum_{s \in \mathcal{S}} p_s \left\| x_s^k - z^{k-1} \right\|_2^2} < \epsilon$ then
17:            return $(x^k, y^k, z^k, \omega^k, \phi^k)$
18:        end if
19:        $\omega_s^{k+1} \leftarrow \omega_s^k + \rho(x_s^k - z^k)$ for all $s \in \mathcal{S}$
20:    end for
21:    return $(x^{k_{max}}, y^{k_{max}}, z^{k_{max}}, \omega^{k_{max}}, \phi^{k_{max}})$
22: end function
```

In Algorithm 1, $k_{max} > 0$ is the maximum number of iterations, and $\epsilon > 0$ parameterized the convergence tolerance. The initialization of Lines 3–8 provides an initial target primal value z^0 and dual values ω_s^1, $s \in \mathcal{S}$, for the main iterations $k \geq 1$. Also, an initial Lagrangian bound ϕ^0 can be computed from this initialization. For $\epsilon > 0$, the Algorithm 1 termination criterion $\sqrt{\sum_{s \in \mathcal{S}} p_s \left\| x_s^k - z^{k-1} \right\|_2^2} < \epsilon$ is motivated by the addition of the squared norms of the primal and dual residuals associated with problem (3). (See Section 3.3 of [7].) In summing the squared norm primal residuals $p_s \left\| x_s^k - z^k \right\|_2^2$, $s \in \mathcal{S}$, and the squared norm dual residual $\left\| z^k - z^{k-1} \right\|_2^2$, we have

$$\sum_{s \in \mathcal{S}} p_s \left[\left\| x_s^k - z^k \right\|_2^2 + \left\| z^k - z^{k-1} \right\|_2^2 \right] = \sum_{s \in \mathcal{S}} p_s \left\| x_s^k - z^{k-1} \right\|_2^2 \tag{10}$$

The equality in (10) follows since, for each $s \in \mathcal{S}$, the cross term resulting from the expansion of the squared norm $\left\|(x_s^k - z^k) + (z^k - z^{k-1})\right\|_2^2$ vanishes; this is seen in the equality $\sum_{s\in\mathcal{S}} p_s(x_s^k - z^k) = 0$ due to the construction of z^k.

The Line 11 subproblem of Algorithm 1 is an addition to the original PH algorithm. Its purpose is to compute Lagrangian bounds (Line 14) from the current dual solution ω^k [14]. Thus, the bulk of computational effort in Algorithm 1 applied to problem (3) (the case with $D_s = K_s$) resides in computing solutions to the MILP (Line 11) and MIQP (Line 12) subproblems. Note that Line 11 may be omitted if the corresponding Lagrangian bound for ω^k is not desired.

2.1. Convergence of PH.

The following proposition addresses the convergence of PH applied to problem (9).

Proposition 1. *Assume that problem (9) is feasible with conv(K_s) bounded for each $s \in \mathcal{S}$, and let Algorithm 1 be applied to problem (9) (so that $D_s =$ conv(K_s) for each $s \in \mathcal{S}$) with tolerance $\epsilon = 0$ for each $k \geq 1$. Then, the limit $\lim_{k\to\infty} \omega^k = \omega^*$ exists, and furthermore,*

1. $\lim_{k\to\infty} \sum_{s\in\mathcal{S}} p_s(c^\top x_s^k + q_s^\top y_s^k) = \zeta^{LD}$,
2. $\lim_{k\to\infty} \phi(\omega^k) = \zeta^{LD}$,
3. $\lim_{k\to\infty} (x_s^k - z^k) = 0$ *for each* $s \in \mathcal{S}$,

and each limit point $(((x_s^*, y_s^*)_{s\in\mathcal{S}}, z^*)$ *is an optimal solution for* (9).

Proof. Since the constraint sets $D_s = \text{conv}(K_s)$, $s \in \mathcal{S}$, are bounded, and problem (9) is feasible, problem (9) has an optimal solution $((x_s^*, y_s^*)_{s\in\mathcal{S}}, z^*)$ with optimal value ζ^{LD}. The feasibility of problem (9), the linearity of its objective function, and the bounded polyhedral structure of its constraint set $D_s = \text{conv}(K_s)$, $s \in \mathcal{S}$, imply that the hypotheses for PH convergence to the optimal solution are met (See Theorem 5.1 of [30]). Therefore, $\left\{\omega^k\right\}$ converges to some ω^*, $\lim_{k\to\infty} \sum_{s\in\mathcal{S}} p_s(c^\top x_s^k + q_s^\top y_s) = \zeta^{LD}$, $\lim_{k\to\infty} \phi(\omega^k) = \zeta^{LD}$, and $\lim_{k\to\infty}(x_s^k - z^k) = 0$ for each $s \in \mathcal{S}$ all hold. The boundedness of each $D_s = \text{conv}(K_s)$, $s \in \mathcal{S}$, furthermore implies the existence of limit points $((x_s^*, y_s^*)_{s\in\mathcal{S}}, z^*)$ of $\left\{((x_s^k, y_s^k)_{s\in\mathcal{S}}, z^k)\right\}$, which are optimal solutions for (9). □

Note that the convergence in Proposition 1 applies to the continuous problem (9) but *not* to the mixed-integer problem (3). In problem (3), the constraint sets K_s, $s \in \mathcal{S}$, are not convex, so there is no guarantee that Algorithm 1 will converge when applied to (3). However, the application of PH to problem (9) requires, in Line 12, the optimization of the Augmented Lagrangian over the sets conv(K_s), $s \in \mathcal{S}$, for which an explicit linear description is unlikely to be known. In the next section, we demonstrate how to circumvent this difficulty by constructing inner approximations of the polyhedral sets conv(K_s), $s \in \mathcal{S}$.

2.2. A Frank-Wolfe approach based on Simplicial Decomposition.

To use Algorithm 1 to solve (9) requires a method for solving the subproblem

$$(x_s^k, y_s^k) \in \operatorname*{argmin}_{x,y} \left\{L_s^\rho(x, y, z^{k-1}, \omega_s^k) : (x, y) \in \text{conv}(K_s)\right\} \tag{11}$$

appearing in Line 12 of the algorithm. Although an explicit description of conv(K_s) is not readily available, if we have a *linear* objective function, then we can replace conv(K_s) with K_s. This motivates the application of a FW algorithm for solving (11), since the FW algorithm solves a sequence of problems in which the nonlinear objective is linearized using a first-order approximation.

The Simplicial Decomposition Method (SDM) is an extension of the FW method, where the line searches of FW are replaced by searches over polyhedral inner approximations. SDM can be applied to solve a feasible, bounded problem of the general form

$$\zeta^{FW} := \min_x \{ f(x) : x \in D \}, \tag{12}$$

with nonempty compact convex set D and continuously differentiable convex function f. Generically, given a current solution x^{t-1} and inner approximation $D^{t-1} \subseteq D$, iteration t of the SDM consists of solving

$$\widehat{x} \in \operatorname*{argmin}_x \left\{ \nabla_x f(x^{t-1})^\top x : x \in D \right\}$$

updating the inner approximation as $D^t \leftarrow \operatorname{conv}(D^{t-1} \cup \{\widehat{x}\})$, and finally choosing

$$x^t \in \operatorname*{argmin}_x \left\{ f(x) : x \in D^t \right\}.$$

The algorithm terminates when the bound gap is small, specifically, when

$$\Gamma^t := -\nabla_x f(x^{t-1})^\top (\widehat{x} - x^{t-1}) < \tau,$$

where $\tau \geq 0$ is a given tolerance.

The application of SDM to solve problem (11), i.e., to minimize $L_s^\rho(x, y, z, \omega_s)$ over $(x, y) \in \operatorname{conv}(K_s)$, for a given $s \in \mathcal{S}$, is presented in Algorithm 2. Here, t_{max} is the maximum number of iterations and $\tau > 0$ is a convergence tolerance. Γ^t is the bound gap used to measure closeness to optimality, and ϕ_s is used to compute a Lagrangian bound as described in the next section. The inner approximation to $\operatorname{conv}(K_s)$ at iteration $t \geq 1$ takes the form $\operatorname{conv}(V_s^t)$, where V_s^t is a finite set of points, with $V_s^t \subset \operatorname{conv}(K_s)$. The points added by Algorithm 2 to the initial set, V_s^0, to form V_s^t, are all in K_s: here $\mathcal{V}(\operatorname{conv}(K_s))$ is the set of extreme points of $\operatorname{conv}(K_s)$ and, of course, $\mathcal{V}(\operatorname{conv}(K_s)) \subseteq K_s$.

Algorithm 2 SDM applied to problem (11).

```
Precondition: $V_s^0 \subset \operatorname{conv}(K_s)$ and $z = \sum_{s \in \mathcal{S}} p_s x_s^0$
function SDM($V_s^0$, $x_s^0$, $\omega_s$, $z$, $t_{max}$, $\tau$)
    for $t = 1, \ldots, t_{max}$ do
        $\widehat{\omega}_s^t \leftarrow \omega_s + \rho(x_s^{t-1} - z)$
        $(\widehat{x}_s, \widehat{y}_s) \in \operatorname{argmin}_{x,y} \left\{ (c + \widehat{\omega}_s^t)^\top x + q_s^\top y \ : \ (x, y) \in \mathcal{V}(\operatorname{conv}(K_s)) \right\}$
        if $t = 1$ then
            $\phi_s \leftarrow (c + \widehat{\omega}_s^t)^\top \widehat{x}_s + q_s^\top \widehat{y}_s$
        end if
        $\Gamma^t \leftarrow -[(c + \widehat{\omega}_s^t)^\top (\widehat{x}_s - x_s^{t-1}) + q_s^\top (\widehat{y}_s - y_s^{t-1})]$
        $V_s^t \leftarrow V_s^{t-1} \cup \{(\widehat{x}, \widehat{y})\}$
        $(x_s^t, y_s^t) \in \operatorname{argmin}_{x,y} \{ L_s^\rho(x, y, z, \omega_s) \ : \ (x, y) \in \operatorname{conv}(V_s^t) \}$
        if $\Gamma^t \leq \tau$ then
            return $(x_s^t, y_s^t, V_s^t, \phi_s)$
        end if
    end for
    return $(x_s^{t_{max}}, y_s^{t_{max}}, V_s^{t_{max}}, \phi_s)$
end function
```

Observe that

$$\nabla_{(x,y)} L_s^\rho(x,y,z,\omega_s)|_{(x,y)=(x_s^{t-1},y_s^{t-1})} = \begin{bmatrix} c+\omega_s+\rho(x_s^{t-1}-z) \\ q_s \end{bmatrix} = \begin{bmatrix} c+\widehat{\omega}_s \\ q_s \end{bmatrix},$$

and so the optimization at Line 5 is minimizing the gradient approximation to $L_s^\rho(x,y,z,\omega_s)$ at the point (x_s^{t-1}, y_s^{t-1}). Since this is a linear objective function, optimization over $\mathcal{V}(\text{conv}(K_s))$ can be accomplished by optimization over K_s (see, e.g., [24], Section I.4, Theorem 6.3). Hence Line 5 requires a solution of a single-scenario MILP.

The optimization at Line 11 can be accomplished by expressing (x,y) as a convex combination of the finite set of points, V_s^t, where the weights $a \in \mathbb{R}^{|V_s^t|}$ in the convex combination are now also decision variables. That is, the Line 11 problem is solved with a solution to the following convex continuous quadratic subproblem

$$(13) \qquad (x_s^t, y_s^t, a) \in \operatorname*{argmin}_{x,y,a} \left\{ \begin{array}{c} L_s^\rho(x,y,z,\omega_s) \ : \ (x,y) = \sum_{(\widehat{x}^i,\widehat{y}^i)\in V_s^t} a_i(\widehat{x}^i,\widehat{y}^i), \\ \sum_{i=1,\dots,|V_s^t|} a_i = 1, \text{ and } a_i \geq 0 \text{ for } i=1,\dots,|V_s^t| \end{array} \right\}.$$

For implementational purposes, the x and y variables may be substituted out of the objective of problem (13), leaving a as the only decision variable, with the only constraints being nonnegativity of the a components and the requirement that they sum to 1.

The Simplicial Decomposition Method is known to terminate finitely with an optimal solution when D is polyhedral [16], so the primal update step Line 12, Algorithm 1 with $D_s = \text{conv}(K_s)$ could be accomplished with the SDM, resulting in an algorithm that converges to a solution giving the Lagrangian dual bound ζ^{LD}. However, since each inner iteration of Line 5, Algorithm 2 requires the solution of a MILP, using $t_{\max}$ large enough to ensure SDM terminates optimally is not efficient for our purpose of computing Lagrangian bounds. In the next section, we give an adapation of the algorithm that requires the solution of only *one* MILP subproblem per scenario at each major iteration of the PH algorithm.

3. The FW-PH method.

In order to make the SDM efficient when used with PH to solve the problem (9), the minimization of the Augmented Lagrangian can be done approximately. This insight can greatly reduce the number of MILP subproblems solved at each inner iteration and forms the basis of our algorithm FW-PH. Convergence of FW-PH relies on the following lemma, which states an important expansion property of the inner approximations employed by the SDM.

LEMMA 2. *For any scenario $s \in \mathcal{S}$ and iteration $k \geq 1$, let Algorithm 2 be applied to the minimization problem (11) for any $t_{max} \geq 2$. For $1 \leq t < t_{max}$, if*

$$(14) \qquad (x_s^t, y_s^t) \notin \operatorname*{argmin}_{x,y} \left\{ L_s^\rho(x,y,z^{k-1},\omega_s^k) \ : \ (x,y) \in conv(K_s) \right\}$$

holds, then $conv(V_s^{t+1}) \supset conv(V_s^t)$.

Proof. For $s \in \mathcal{S}$ and $k \geq 1$ fixed, we know that by construction

$$(x_s^t, y_s^t) \in \operatorname*{argmin}_{x,y} \left\{ L_\rho^s(x,y,z^{k-1},\omega_s^k) : (x,y) \in \text{conv}(V_s^t) \right\}$$

for $t \geq 1$. Given the convexity of $(x,y) \mapsto L_\rho^s(x,y,z^{k-1},\omega_s^k)$ and the convexity of $\text{conv}(V_s^t)$, the necessary and sufficient condition for optimality

$$(15) \qquad \nabla_{(x,y)} L_\rho^s(x_s^t, y_s^t, z^{k-1}, \omega_s^k) \begin{bmatrix} x - x_s^t \\ y - y_s^t \end{bmatrix} \geq 0 \quad \text{for all } (x,y) \in \text{conv}(V_s^t)$$

is satisfied. By assumption, condition (14) is satisfied, $\mathrm{conv}(K_s)$ is likewise convex, and so the resulting *non*-satisfaction of the necessary and sufficient condition of optimality for the problem in (14) takes the form

$$\min_{x,y} \left\{ \nabla_{(x,y)} L^s_\rho(x^t_s, y^t_s, z^{k-1}, \omega^k_s) \begin{bmatrix} x - x^t_s \\ y - y^t_s \end{bmatrix} \; : \; (x,y) \in \mathrm{conv}(K_s) \right\} < 0. \tag{16}$$

In fact, during SDM iteration $t+1$, an optimal solution $(\widehat{x}_s, \widehat{y}_s)$ to the problem in condition (16) is computed in Line 5 of Algorithm 2. Therefore, by the satisfaction of condition (15) and the optimality of $(\widehat{x}_s, \widehat{y}_s)$ for the problem of condition (16), which is also satisfied, we have $(\widehat{x}_s, \widehat{y}_s) \notin \mathrm{conv}(V^t_s)$. By construction, $V^{t+1}_s \leftarrow V^t_s \cup \{(\widehat{x}_s, \widehat{y}_s)\}$, so that $\mathrm{conv}(V^{t+1}_s) \supset \mathrm{conv}(V^t_s)$ must hold. □

The FW-PH algorithm is stated in pseudocode-form in Algorithm 3. Similar to Algorithm 1, the parameter ϵ is a convergence tolerance, and k_{max} is the maximum number of (outer) iterations. The parameter t_{max} is the maximum number of (inner) SDM iterations in Algorithm 2.

The parameter $\alpha \in \mathbb{R}$ affects the initial linearization point $\tilde{x}_s$ of the SDM method. Any value $\alpha \in \mathbb{R}$ may be used, but the use of $\tilde{x}_s = (1-\alpha)z^{k-1} + \alpha x^{k-1}_s$ in Line 6 is a crucial component in the efficiency of the FW-PH algorithm, as it enables the computation of a valid dual bound, ϕ^k, at *each* iteration of FW-PH without the need for additional MILP subproblem solutions. Specifically, we have the following result.

Proposition 3. *Assume that the precondition $\sum_{s \in \mathcal{S}} p_s \omega^0_s = 0$ holds for Algorithm 3. At each iteration $k \geq 1$ of Algorithm 3, the value, ϕ^k, calculated at Line 9, is the value of the Lagrangian relaxation $\phi(\cdot)$ evaluated at a Lagrangian dual feasible point, and hence provides a finite lower bound on ζ^{LD}.*

Proof. Since $\sum_{s \in \mathcal{S}} p_s \omega^0_s = 0$ holds and, by construction, $0 = \sum_{s \in \mathcal{S}} p_s (x^0_s - z^0)$, we have $\sum_{s \in \mathcal{S}} p_s \omega^1_s = 0$ also. We proceed by induction on $k \geq 1$. At iteration k, the problem solved for each $s \in \mathcal{S}$ at Line 5 in the first iteration ($t = 1$) of Algorithm 2 may be solved with the same optimal value by exchanging $\mathcal{V}(\mathrm{conv}(K_s))$ for K_s; this follows from the linearity of the objective function. Thus, an optimal solution computed at Line 5 may be used in the computation of $\phi_s(\widetilde{\omega}^k_s)$ carried out in Line 7, where

$$\begin{aligned} \widetilde{\omega}^k_s &:= \widehat{\omega}^1_s = \omega^k_s + \rho(\widetilde{x}_s - z^{k-1}) = \omega^k_s + \rho((1-\alpha)z^{k-1} + \alpha x^{k-1}_s - z^{k-1}) \\ &= \omega^k_s + \alpha\rho(x^{k-1}_s - z^{k-1}). \end{aligned}$$

By construction, we have at each iteration $k \geq 1$ in Algorithm 3 that

$$\sum_{s \in \mathcal{S}} p_s (x^{k-1}_s - z^{k-1}) = 0 \quad \text{and} \quad \sum_{s \in \mathcal{S}} p_s \omega^k_s = 0,$$

which establishes that $\sum_{s \in \mathcal{S}} p_s \widetilde{\omega}^k_s = 0$. Thus, $\widetilde{\omega}^k$ is feasible for the Lagrangian dual problem, so that $\phi(\widetilde{\omega}^k) = \sum_{s \in \mathcal{S}} p_s \phi^k_s$, and, since each ϕ^k_s is the optimal value of a bounded and feasible mixed-integer linear program, we have $-\infty < \phi(\widetilde{\omega}^k) < \infty$. □

We establish convergence of Algorithm 3 for any $\alpha \in \mathbb{R}$ and $t_{max} \geq 1$. For the special case where we perform only one iteration of SDM for each outer iteration ($t_{max} = 1$), we require the additional assumption that the initial scenario vertex sets share a common point. More precisely, we require the assumption

$$\bigcap_{s \in \mathcal{S}} \mathrm{Proj}_x(\mathrm{conv}(V^0_s)) \neq \emptyset \tag{17}$$

Algorithm 3 FW-PH applied to problem (9).

```
 1: function FW-PH($(V_s^0)_{s\in\mathcal{S}}$, $(x_s^0, y_s^0)_{s\in\mathcal{S}}$, $\omega^0$, $\rho$, $\alpha$, $\epsilon$, $k_{max}$, $t_{max}$)
 2:     $z^0 \leftarrow \sum_{s\in\mathcal{S}} p_s x_s^0$
 3:     $\omega_s^1 \leftarrow \omega_s^0 + \rho(x_s^0 - z^0)$, for $s \in \mathcal{S}$
 4:     for $k = 1, \ldots, k_{max}$ do
 5:         for $s \in \mathcal{S}$ do
 6:             $\widetilde{x}_s \leftarrow (1-\alpha)z^{k-1} + \alpha x_s^{k-1}$
 7:             $[x_s^k, y_s^k, V_s^k, \phi_s^k] \leftarrow \text{SDM}(V_s^{k-1}, \widetilde{x}_s, \omega_s^k, z^{k-1}, t_{max}, 0)$
 8:         end for
 9:         $\phi^k \leftarrow \sum_{s\in\mathcal{S}} p_s \phi_s^k$
10:         $z^k \leftarrow \sum_{s\in\mathcal{S}} p_s x_s^k$
11:         if $\sqrt{\sum_{s\in\mathcal{S}} p_s \left\|x_s^k - z^{k-1}\right\|_2^2} < \epsilon$ then
12:             return $((x_s^k, y_s^k)_{s\in\mathcal{S}}, z^k, \omega^k, \phi^k)$
13:         end if
14:         $\omega_s^{k+1} \leftarrow \omega_s^k + \rho(x_s^k - z^k)$, for $s \in \mathcal{S}$
15:     end for
16:     return $\big((x_s^{k_{max}}, y_s^{k_{max}})_{s\in\mathcal{S}}, z^{k_{max}}), \omega^{k_{max}}, \phi^{k_{max}}\big)$
17: end function
```

which can, in practice, be effectively handled through appropriate initialization, under the standard assumption of relatively complete recourse. We describe one initialization procedure in Section 4.

PROPOSITION 4. *Let the convexified separable deterministic equivalent SMIP* (9) *have an optimal solution, and let Algorithm 3 be applied to* (9) *with* $k_{max} = \infty$, $\epsilon = 0$, $\alpha \in \mathbb{R}$, *and* $t_{max} \geq 1$. *If either* $t_{max} \geq 2$ *or* (17) *holds, then* $\lim_{k\to\infty} \phi^k = \zeta^{LD}$.

Proof. First note that for any $t_{max} \geq 1$, the sequence of inner approximations $\text{conv}(V_s^k)$, $s \in \mathcal{S}$, will stabilize, in that, for some threshold $0 \leq \bar{k}_s$, we have for all $k \geq \bar{k}_s$

$$\text{(18)} \qquad \text{conv}(V_s^k) =: \overline{D}_s \subseteq \text{conv}(K_s).$$

This follows due to the assumption that each expansion of the inner approximations $\text{conv}(V_s^k)$ take the form $V_s^k \leftarrow V_s^{k-1} \cup \{(\widehat{x}_s, \widehat{y}_s)\}$, where $(\widehat{x}_s, \widehat{y}_s)$ is a vertex of $\text{conv}(K_s)$. Since each polyhedron $\text{conv}(K_s)$, $s \in \mathcal{S}$ has only a finite number of such vertices, the stabilization (18) must occur at some $\bar{k}_s < \infty$.

For $t_{\text{max}} \geq 2$, the stabilizations (18), $s \in \mathcal{S}$, are reached at some iteration $\bar{k} := \max_{s\in\mathcal{S}} \left\{\bar{k}_s\right\}$. Noting that $\overline{D}_s = \text{conv}(V_s^k)$ for $k > \bar{k}$ we must have

$$\text{(19)} \qquad (x_s^k, y_s^k) \in \underset{x,y}{\text{argmin}} \left\{L_s^\rho(x, y, z^{k-1}, \omega_s^k) : (x, y) \in \text{conv}(K_s)\right\}.$$

Otherwise, due to Lemma 2, the call to SDM on Line 7 must return $V_s^k \supset V_s^{k-1}$, contradicting the finite stabilization (18). Therefore, the $k \geq \bar{k}$ iterations of Algorithm 3 are identical to Algorithm 1 iterations, and so Proposition 1 implies that $\lim_{k\to\infty} x_s^k - z^k = 0$, $s \in \mathcal{S}$, and $\lim_{k\to\infty} \phi(\omega^k) = \zeta^{LD}$. By the continuity of $\omega \mapsto \phi_s(\omega)$ for each $s \in \mathcal{S}$, we have $\lim_{k\to\infty} \phi^k = \lim_{k\to\infty} \sum_{s\in\mathcal{S}} p_s \phi_s(\omega_s^k + \alpha(x_s^{k-1} - z^{k-1})) = \lim_{k\to\infty} \sum_{s\in\mathcal{S}} p_s \phi_s(\omega_s^k) = \lim_{k\to\infty} \phi(\omega^k) = \zeta^{LD}$ for all $\alpha \in \mathbb{R}$.

In the case $t_{max} = 1$, we have at each iteration $k \geq 1$ the optimality

$$(x_s^k, y_s^k) \in \underset{x,y}{\text{argmin}} \left\{L_s^\rho(x_s, y_s, z^{k-1}, \omega_s^k) \ : \ (x_s, y_s) \in \text{conv}(V_s^k)\right\}.$$

By the stabilization (18), the iterations $k \geq \bar{k}$ of Algorithm 3 are identical to PH iterations applied to the restricted problem

$$\min_{x,y,z} \left\{ \sum_{s \in \mathcal{S}} p_s(c^\top x_s + q_s^\top y_s) \ : \ (x_s, y_s) \in \overline{D}_s, \ \forall s \in \mathcal{S}, x_s = z, \ \forall s \in \mathcal{S} \right\}. \tag{20}$$

We have initialized the sets $(V_s^0)_{s \in \mathcal{S}}$ such that $\cap_{s \in \mathcal{S}} \text{Proj}_x \text{conv}(V_s^0) \neq \emptyset$, so since the inner approximations to $\text{conv}(K_s)$ only expand in the algorithm, $\cap_{s \in \mathcal{S}} \text{Proj}_x(\overline{D}_s) \neq \emptyset$. Therefore, problem (20) is a feasible and bounded linear program, and so the PH convergence described in Proposition 1 with $D_s = \overline{D}_s$, $s \in \mathcal{S}$, holds for its application to problem (20). That is, for each $s \in \mathcal{S}$, we have 1) $\lim_{k \to \infty} \omega_s^k = \omega_s^*$ and $\lim_{k \to \infty}(x_s^k - z^k) = 0$; and 2) for all limit points $((x_s^*, y_s^*)_{s \in \mathcal{S}}, z^*)$, we have the feasibility and optimality of the limit points, which implies $x_s^* = z^*$ and

$$\min_{x,y} \left\{ (c + \omega_s^*)^\top (x - x^*) + q_s^\top (y - y^*) \ : \ (x, y) \in \overline{D}_s \right\} = 0 \tag{21}$$

Next, for each $s \in \mathcal{S}$ the compactness of $\text{conv}(K_s) \supseteq \overline{D}_s$, the continuity of the minimum value function

$$\omega \mapsto \min_{x,y} \left\{ (c + \omega)^\top x + q_s^\top y \ : \ (x, y) \in \overline{D}_s \right\},$$

and the limit $\lim_{k \to \infty} \widetilde{\omega}_s^{k+1} = \lim_{k \to \infty} \omega_s^{k+1} + \alpha\rho(x_s^k - z^k) = \omega_s^*$, together imply that

$$\lim_{k \to \infty} \min_{x,y} \left\{ (c + \widetilde{\omega}_s^{k+1})^\top (x - x^k) + q_s^\top (y - y^k) \ : \ (x, y) \in \overline{D}_s \right\} = 0. \tag{22}$$

Recall that $\widetilde{\omega}_s^k = \omega_s^k + \rho\alpha(x_s^{k-1} - z^{k-1})$ is the $t = 1$ value of $\widehat{\omega}_s^t$ defined in Line 4 of Algorithm 2. Thus, for $k + 1 > \bar{k}$, we have due to the stabilization (18) that

$$\begin{aligned} &\min_{x,y} \left\{ (c + \widetilde{\omega}_s^{k+1})^\top (x - x^k) + q_s^\top (y - y^k) \ : \ (x, y) \in \overline{D}_s \right\} = \\ &\qquad \min_{x,y} \left\{ (c + \widetilde{\omega}_s^{k+1})^\top (x - x^k) + q_s^\top (y - y^k) \ : \ (x, y) \in \text{conv}(K_s) \right\} \end{aligned} \tag{23}$$

If equality (23) does not hold, then the inner approximation expansion $\overline{D}_s \subset \text{conv}(V_s^{k+1})$ must occur, since a point $(\widehat{x}_s, \widehat{y}_s) \in \text{conv}(K_s)$ that can be strictly separated from $\overline{D}_s$ would have been discovered during the iteration $k + 1$ execution of Algorithm 2, Line 5, $t = 1$. The expansion $\overline{D}_s \subset \text{conv}(V_s^{k+1})$ contradicts the finite stabilization (18), and so (23) holds. Therefore, the equalities (22) and (23) imply that

$$\lim_{k \to \infty} \min_{x,y} \left\{ (c + \widetilde{\omega}_s^{k+1})^\top (x - x^k) + q_s^\top (y - y^k) \ : \ (x, y) \in \text{conv}(K_s) \right\} = 0. \tag{24}$$

Our argument has shown that for all limit points (x_s^*, y_s^*), $s \in \mathcal{S}$, the following stationarity condition is satsfied:

$$(c + \omega_s^*)^\top (x - x_s^*) + q_s^\top (y - y_s^*) \geq 0 \quad \forall (x, y) \in \text{conv}(K_s), \tag{25}$$

which together with the feasibility $x_s^* = z^*$, $s \in \mathcal{S}$ implies that each limit point $((x_s^*, y_s^*)_{s \in \mathcal{S}}, z^*)$ is optimal for problem (9) and ω^* is optimal for the dual problem (8).

Thus, for all $t_{max} \geq 1$, we have shown $\lim_{k \to \infty}(x_s^k - z^k) = 0$, $s \in \mathcal{S}$, and $\lim_{k \to \infty} \phi(\omega^k) = \zeta^{LD}$. By similar reasoning used in the $t_{max} \geq 2$ case, it is straightforward that for all $\alpha \in \mathbb{R}$, we also have $\lim_{k \to \infty} \phi^k = \zeta^{LD}$. □

While using a large value of t_{max} more closely matches Algorithm 3 to the original PH algorithm as described in Algorithm 1, we are motivated to use a small value of t_{max} since the work per iteration is proportional to t_{max}. Specifically, each iteration requires solving $t_{max}|\mathcal{S}|$ MILP subproblems, and $t_{max}|\mathcal{S}|$ continuous convex quadratic subproblems. (For reference, Algorithm 1 applied to problem (3) requires the solution of $|\mathcal{S}|$ MIQP subproblems for each ω update and $|\mathcal{S}|$ MILP subproblems for each Lagrangian bound ϕ computation.)

4. Numerical Experiments.

We performed computations using a C++ implementation of Algorithm 1 ($D_s = K_s$, $s \in \mathcal{S}$) and Algorithm 3 using CPLEX 12.5 [17] as the solver for all subproblems. For reading SMPS files into scenario-specific subproblems and for their interface with CPLEX, we used modified versions of the COIN-OR [3] Smi and Osi libraries. The computing environment is the Raijin cluster maintained by Australia's National Computing Infrastructure (NCI) and supported by the Australian Government [1]. The Raijin cluster is a high performance computing (HPC) environment which has 3592 nodes (system units), 57472 cores of Intel Xeon E5-2670 processors with up to 8 GB PC1600 memory per core (128 GB per node). All experiments were performed in a serial setting using a single node and one thread per CPLEX solve.

In the experiments with Algorithms 1 and 3, we set the convergence tolerance at $\epsilon = 10^{-3}$. For Algorithm 3, we set $t_{max} = 1$. Also, for all experiments performed, we set $\omega^0 = 0$. In this case, convergence of our algorithm requires that (17) holds, which can be guaranteed during the initialization of the inner approximations $(V_s^0)_{s\in\mathcal{S}}$. Under the standard assumption of *relatively complete resource*, i.e., the assumption that for all $x \in X$ and $s \in \mathcal{S}$ there exists y_s such that $(x, y_s) \in K_s$, a straightforward mechanism for ensuring this assumption is to solve the recourse problems for any fixed $\widehat{x} \in X$. Specifically, for each $s \in \mathcal{S}$, let

$$\widehat{y}_s \in \arg\min_{y}\{q_s^\top y \;:\; (\widehat{x}, y) \in K_s\},$$

and initialize V_s^0 for each $s \in \mathcal{S}$ so that $\{(\widehat{x}, \widehat{y}_s)\} \in V_s^0$. Observe also that this initialization corresponds to a technique for computing a feasible solution to the original problem (2), which is independently useful for obtaining an *upper bound* on ζ^{SMIP}.

For the computational experiments, we run the following initialization to obtain $(V_s^0)_{s\in\mathcal{S}}$ and $(x_s^0, y_s^0)_{s\in\mathcal{S}}$ that are input into Algorithm 3:

Algorithm 4 Initialization step for FW-PH

```
Precondition: Problem (2) has relatively complete recourse
function FW-PH-Initialization($\omega^0$)
    for $s \in \mathcal{S}$ do
        $(x_s^0, y_s^0) \leftarrow \text{argmin}_{x,y}\left\{(c + \omega_s^0)^\top x + q_s^\top y \;:\; (x, y) \in K_s\right\}$
        $V_s^0 \leftarrow \left\{(x_s^0, y_s^0)\right\}$
        if $s \neq 1$ then
            $\overline{y}_s \leftarrow \text{argmin}_y\left\{q_s^\top y \;:\; (x_1^0, y) \in K_s\right\}$
            $V_s^0 \leftarrow V_s^0 \cup \left\{(x_1^0, \overline{y}_s)\right\}$
        end if
    end for
    return $\left(V_s^0, (x_s^0, y_s^0)\right)_{s\in\mathcal{S}}$
end function
```

If problem (2) does not have relatively complete recourse, then any means to compute a feasible solution to (2) may be employed to initialize each V_s^0, $s \in \mathcal{S}$, in a way to satisfy (17).

Two sets of Algorithm 3 experiments correspond to variants considering $\alpha = 0$ and $\alpha = 1$. Computations were performed on four problems: the CAP instance 101 with the first 250 scenarios (CAP-101-250) [6], the DCAP instance DCAP233_500 with 500 scenarios, the SSLP instances SSLP5.25.50 with 50 scenarios (SSLP-5-25-50) and SSLP10.50.100 with 100 scenarios (SSLP-10-50-100). The latter three problems are described in detail in [26, 2] and accessible at [2]. For each problem, computations were performed for different penalty values $\rho > 0$. The penalty values used in the experiments for the SSLP-5-25-50 instance were chosen to include those penalties that are tested in a computational experiment with PH whose results are depicted in Figure 2 of [14]. For the other problem instances, the set of penalty values ρ tested is chosen to capture a reasonably wide range of performance potential for both PH and FW-PH. All computational experiments were allowed to run for a maximum of two hours in wall clock time.

	Percentage gap			# Iterations			Termination		
Penalty	PH	FW-PH		PH	FW-PH		PH	FW-PH	
		$\alpha = 0$	$\alpha = 1$		$\alpha = 0$	$\alpha = 1$		$\alpha = 0$	$\alpha = 1$
20	0.08%	0.10%	0.11%	466	439	430	T	T	T
100	0.01%	0.00%	0.00%	178	406	437	C	T	T
500	0.07%	0.00%	0.00%	468	92	93	T	C	C
1000	0.15%	0.00%	0.00%	516	127	130	T	C	C
2500	0.34%	0.00%	0.00%	469	259	274	T	C	C
5000	0.66%	0.00%	0.00%	33	431	464	C	T	T
7500	0.99%	0.00%	0.00%	28	18	19	C	C	C
15000	1.59%	0.00%	0.00%	567	28	33	T	C	C

TABLE 1

Result summary for CAP-101-250, with the absolute percentage gap based on the known optimal value 733827.3

	Percentage gap			# Iterations			Termination		
Penalty	PH	FW-PH		PH	FW-PH		PH	FW-PH	
		$\alpha = 0$	$\alpha = 1$		$\alpha = 0$	$\alpha = 1$		$\alpha = 0$	$\alpha = 1$
2	0.13%	0.12%	0.12%	1717	574	600	T	T	T
5	0.22%	0.09%	0.09%	2074	589	574	T	T	T
10	0.23%	0.07%	0.07%	2598	592	587	T	T	T
20	0.35%	0.07%	0.07%	1942	590	599	T	T	T
50	1.25%	0.06%	0.06%	2718	597	533	T	T	T
100	1.29%	0.06%	0.06%	2772	428	438	T	C	C
200	2.58%	0.06%	0.06%	2695	256	262	T	C	C
500	2.58%	0.07%	0.07%	2871	244	246	T	C	C

TABLE 2

Result summary for DCAP-233-500, with the absolute percentage gap based on the best known lower bound 1737.7.

	Percentage gap			# Iterations			Termination		
Penalty	PH	FW-PH		PH	FW-PH		PH	FW-PH	
		$\alpha = 0$	$\alpha = 1$		$\alpha = 0$	$\alpha = 1$		$\alpha = 0$	$\alpha = 1$
1	0.30%	0.00%	0.00%	105	115	116	C	C	C
2	0.73%	0.00%	0.00%	51	56	56	C	C	C
5	0.91%	0.00%	0.00%	25	26	27	C	C	C
15	3.15%	0.00%	0.00%	12	16	17	C	C	C
30	6.45%	0.00%	0.00%	12	18	18	C	C	C
50	9.48%	0.00%	0.00%	18	25	26	C	C	C
100	9.48%	0.00%	0.00%	8	45	45	C	C	C

TABLE 3

Result summary for SSLP-5-25-50, with the absolute percentage gap based on the known optimal value -121.6

	Percentage gap			# Iterations			Termination		
Penalty	PH	FW-PH		PH	FW-PH		PH	FW-PH	
		$\alpha = 0$	$\alpha = 1$		$\alpha = 0$	$\alpha = 1$		$\alpha = 0$	$\alpha = 1$
1	0.57%	0.22%	0.23%	126	234	233	T	T	T
2	0.63%	0.03%	0.03%	127	226	228	T	T	T
5	1.00%	0.00%	0.00%	104	219	220	C	T	T
15	2.92%	0.00%	0.00%	33	45	118	C	C	C
30	4.63%	0.00%	0.00%	18	21	22	C	C	C
50	4.63%	0.00%	0.00%	11	26	27	C	C	C
100	4.63%	0.00%	0.00%	9	43	45	C	C	C

TABLE 4

Result summary for SSLP-10-50-100, with the absolute percentage gap based on the known optimal value -354.2

Tables 1–4 provide a summary indicating the quality of the Lagrangian bounds ϕ computed at the end of each experiment for the four problems with varying penalty parameter ρ. In each of these tables, the first column lists the values of the penalty parameter ρ, while the following are presented for PH and FW-PH (for both $\alpha = 0$ and $\alpha = 1$) computations in the remaining columns: 1) the absolute percentage gap $\left|\frac{\zeta^*-\phi}{\zeta^*}\right| * 100\%$ between the computed Lagrangian bound ϕ and some reference value ζ^* that is either a known optimal value for the problem, or a known best upper bound thereof (column "Percentage Gap"); 2) the total number of dual updates ("# Iterations"); and 3) the indication of whether the algorithm terminated due to the time limit, indicated by letter "T", or the satisfaction of the convergence criterion $\sqrt{\sum_{s\in\mathcal{S}} p_s \left\|x_s^k - z^{k-1}\right\|_2^2} < \epsilon$, indicated by letter "C" (column "Termination").

The following observations can be made from the results presented in Tables 1–4. First, for small values of the penalty ρ, there is no clear preference between the bounds ϕ generated by PH and FW-PH. However, for higher penalties, the bounds ϕ obtained by FW-PH are consistently of better quality (i.e., higher) than those obtained by PH, regardless of the variant used (i.e. $\alpha = 0$ or $\alpha = 1$). This tendency is typically illustrated, for example, in Table 2, where the absolute percentage gap of the Lagrangian lower bound with the known optimal value was 0.06% with $\rho = 200$ for FW-PH ($\alpha = 0$), while it was 2.58% for the same value of ρ for PH. This improvement is consistently observed for the other problems and the other values of ρ that are not too close to zero. Also, FW-PH did not terminate with suboptimal convergence or display cycling behavior for any of the penalty values ρ in any of the problems considered. For example, all experiments considered in Table 3 terminated due to convergence. The percentage gaps suggest that the convergence for PH was suboptimal, while it was optimal for FW-PH. Moreover, it is possible to see from these tables that the quality of the bounds ϕ obtained using FW-PH were not as sensitive to the value of the penalty parameter ρ as obtained using PH.

The FW-PH with $\alpha = 0$ versus PH convergence profiles for the experiments performed are given in Figures 1–4, in which we provide plots of wall time versus Lagrangian bound values based on profiling of varying penalty. The times scales for the plots have been set such that trends are meaningfully depicted (1000s for CAP-101-250 and DCAP-233-500, 100 seconds for SSLP-5-25-50, and 3000s for SSLP-10-50-100). The trend of the Lagrangian bounds is depicted with solid lines for FW-PH with $\alpha = 0$ and with dashed lines for PH. Plots providing the same comparison for FW-PH with $\alpha = 1$ are provided in Appendix A.

As seen in the plots of Figures 1–4, the Lagrangian bounds ϕ generated with PH tend to converge suboptimally, often displaying cycling, for large penalty values. In terms of the quality of the bounds obtained, while there is no clear winner when low penalty ρ values are used, for large penalties, the quality of the bounds ϕ generated with FW-PH is consistently better than for the bounds generated with PH, regardless of the α value. This last observation is significant because the effective use of large penalty values ρ in methods based on augmented Lagrangian relaxation tends to yield the most rapid early iteration improvement in the Lagrangian bound; this point is most clearly illustrated in the plot of Figure 3.

5. Conclusions and future work. In this paper, we have presented an alternative approach to compute nonanticipativity Lagrangian bounds associated with SMIPs that combines ideas from the Progressive Hedging (PH) and the Frank-Wolfe (FW) based methods. We first note that while Lagrangian bounds can be recovered with PH, this requires—for each iteration and each scenario—the solution of an additional MILP subproblem in addition to the MIQP subproblem. Furthermore, when using the PH method directly, the Lagrangian bounds may converge suboptimally, cycle (for large penalties), or converge very slowly (for small penalties).

To overcome the lack of theoretical support for the above use of PH, we first described a straightforward integration of PH and a FW-like approach such as the Simplicial Decomposition Method (SDM), where SDM is used to compute the primal updates in PH. Its convergence only requires noting that SDM applied to a convex problem with a bounded polyhedral constraint set terminates finitely with optimal convergence. However, for the stated goal of computing high-quality Lagrangian bounds efficiently, the benefits of relying on the optimal convergence of SDM is far outweighed by the computational costs incurred.

As an alternative, we propose the contributed algorithm, FW-PH, that is analyzed under general assumptions on how the Lagrangian bounds are computed and on the number of SDM iterations used for each dual update. Furthermore, under mild assumptions on the initialization of the algorithm, FW-PH only requires the solution of a MILP subproblem and a continuous convex quadratic subproblem for each iteration and each scenario. FW-PH is versatile enough to handle a wide range of

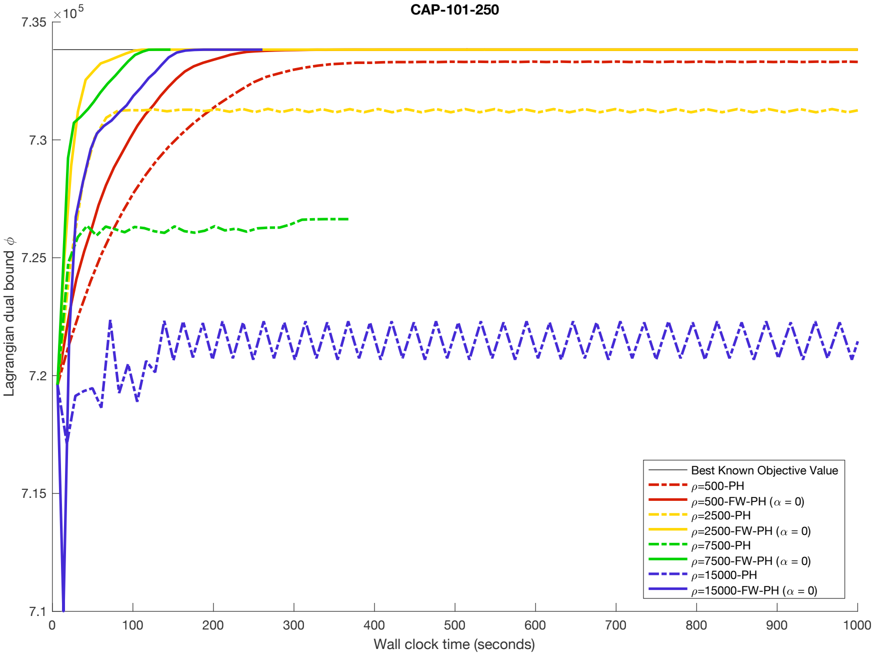

FIG. 1. *Convergence profile for CAP-101-250 (PH and FW-PH with $\alpha = 0$)*

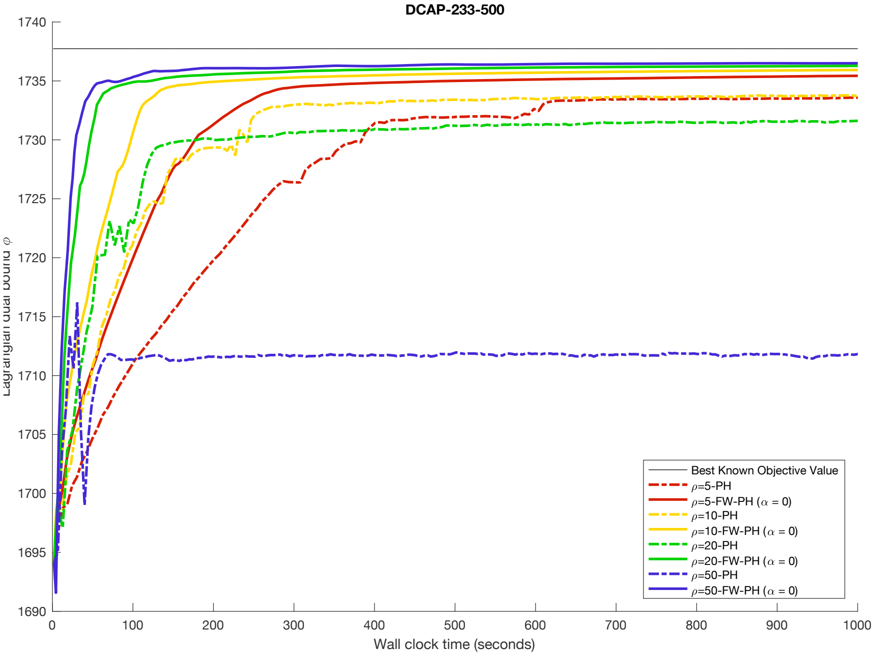

FIG. 2. *Convergence profile for DCAP-233-500 (PH and FW-PH with $\alpha = 0$)*

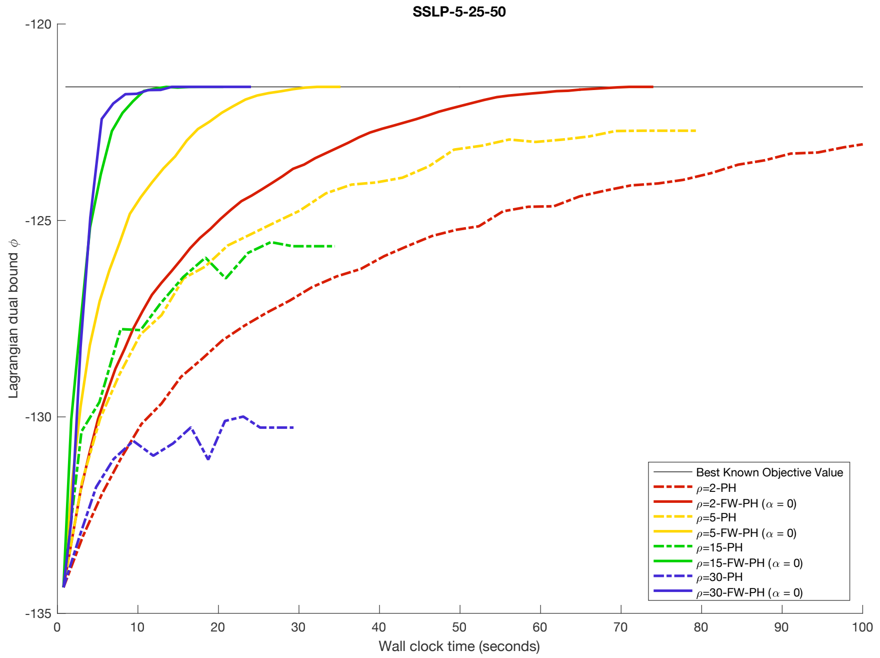

FIG. 3. *Convergence profile for SSLP-5-25-50 (PH and FW-PH with $\alpha = 0$)*

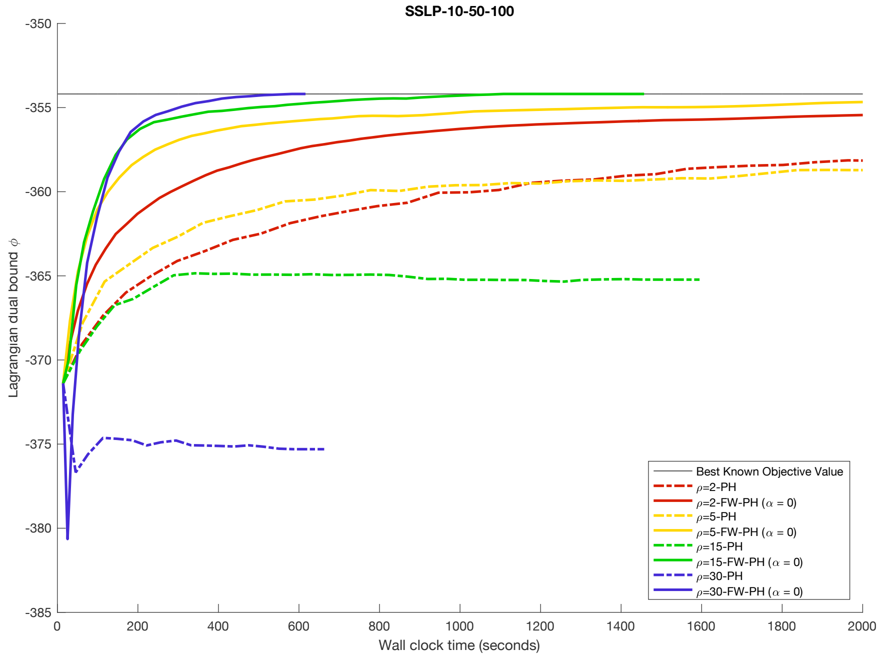

FIG. 4. *Convergence profile for SSLP-10-50-100 (PH and FW-PH with $\alpha = 0$)*

SMIPs with integrality restrictions in any stage, while providing rapid improvement in the Lagrangian bound in the early iterations that is consistent across a wide range of penalty parameter values. Although we have opted to focus on two-stage problems with recourse, the generalization of the proposed approach to the multi-stage case is also possible.

Numerical results are encouraging as they suggest that the proposed FW-PH method applied to SMIP problems usually outperforms the traditional PH method with respect to how quickly the quality of the generated Lagrangian bounds improves. This is especially true with the use of larger penalty values. For all problems considered and for all but the smallest penalties considered, the FW-PH method displayed better performance over PH in terms of the quality of the final Lagrangian bounds at the end of the allotted wall clock time.

The improved performance of FW-PH over PH for large penalties is significant because it is the effective use of large penalties enabled by FW-PH that yields the most rapid initial dual improvement. This last feature of FW-PH would be most helpful in its use within a branch-and-bound or branch-and-cut framework for providing strong lower bounds (in the case of minimization). In addition to being another means to compute Lagrangian bounds, PH would still have a role in such frameworks as a heuristic for computing a primal feasible solution to the SMIP, thus providing (in the case of minimization) an upper bound on the optimal value.

Future research on this subject includes the following. First, FW-PH inherits the potential for parallelization from PH. Experiments for exploring the benefit of parallelization are therefore warranted. Second, the theoretical support of FW-PH can be strengthened with a better understanding of the behavior of PH (and its generalization ADMM) applied to infeasible problems. Finally, FW-PH can benefit from a better understanding of how the proximal term penalty coefficient can be varied to improve performance.

REFERENCES

[1] http://www.nci.org.au. Last accessed 20 April, 2016.

[2] http://www2.isye.gatech.edu/~sahmed/siplib/sslp/sslp.html. Last accessed 23 December, 2015.

[3] *COmputational INfrastructure for Operations Research*, http://www.coin-or.org/. Last accessed 28 January, 2016.

[4] A. Agustín, A. Alonso-Ayuso, L. F. Escudero, and C. Pizarro, *On air traffic flow management with rerouting. Part II: Stochastic case*, European Journal of Operational Research, 219 (2012), pp. 167–177.

[5] F. Badilla Veliz, J.-P. Watson, A. Weintraub, R. J.-B. Wets, and D. L. Woodruff, *Stochastic optimization models in forest planning: a progressive hedging solution approach*, Annals of Operations Research, 232 (2015), pp. 259–274.

[6] M. Bodur, S. Dash, O. Günlük, and J. Luedtke, *Strengthened benders cuts for stochastic integer programs with continuous recourse*, (2014), http://www.optimization-online.org/DB_FILE/2014/03/4263.pdf. Last accessed on 13 January 2015.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundation and Trends in Machine Learning, 3 (2011), pp. 1–122.

[8] C. C. Carøe and R. Schultz, *Dual decomposition in stochastic integer programming*, Operations Research Letters, 24 (1999), pp. 37–45.

[9] J. Chen, C. H. Lim, P. Qian, J. Linderoth, and S. J. Wright, *Validating sample average approximation solutions with negatively dependent batches*, http://arxiv.org/abs/1404.7208. Last accessed 30 March, 2016.

[10] T. H. de Mello, *On rates of convergence for stochastic optimization problems under non-I.I.D. sampling*, SIAM Journal on Optimization, 19 (2008), pp. 524–551.

[11] J. Eckstein and D. Bertsekas, *On the Douglas-Rachford splitting method and the proximal*

point algorithm for maximal monotone operators, Mathematical Programming, 55 (1992), pp. 293–318.

[12] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Research Logistics Quarterly, 3 (1956), pp. 149–154.

[13] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Computers and Mathematics with Applications, 2 (1976), pp. 17–40.

[14] D. Gade, G. Hackebeil, S. M. Ryan, J.-P. Watson, R. J.-B. Wets, and D. L. Woodruff, *Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs*, Mathematical Programming, 157 (2016), pp. 47–67.

[15] M. R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization Theory and Applications, (1969), pp. 303–320.

[16] C. Holloway, *An extension of the Frank and Wolfe method of feasible directions*, Mathematical Programming, 6 (1974), pp. 14–27.

[17] IBM Corporation, *IBM ILOG CPLEX V12.5*, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/. Last accessed 28 Jan 2016.

[18] A. Kleywegt, A. Shapiro, and T. Homem-de-Mello, *The sample average approximation method for stochastic discrete optimization*, SIAM Journal on Optimization, 2 (2001), pp. 479–502.

[19] G. Laporte, F. V. Louveaux, and H. Mercure, *The vehicle routing problem with stochastic travel times*, Transportation Science, 26 (1992), pp. 161–170.

[20] A. Løkketangen and D. Woodruff, *Progressive hedging and tabu search applied to mixed integer (0,1) multi-stage stochastic programming*, Journal of Heuristics, 2 (1996), pp. 111–128.

[21] F. V. Louveaux, *Discrete stochastic location models*, Annals of Operations Research, 6 (1986), pp. 23–34.

[22] M. Lubin, K. Martin, C. Petra, and B. Sandkçi, *On parallelizing dual decomposition in stochastic integer programming*, Operations Research Letters, 41 (2013), pp. 252–258.

[23] W. K. Mak, D. P. Morton, and R. K. Wood, *Monte Carlo bounding techniques for determining solution quality in stochastic programs*, Operations Research Letters, 24 (1999), pp. 47–56.

[24] G. Nemhauser and L. Wolsey, *Integer and combinatorial optimization*, Wiley-Interscience series in discrete mathematics and optimization, Wiley, 1988.

[25] M. P. Nowak and W. Römisch, *Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty*, Annals of Operations Research, 100 (2000), pp. 251–272.

[26] L. Ntaimo, *Decomposition Algorithms for Stochastic Combinatorial Optimization: Computational Experiments and Extensions*, PhD thesis, 2004.

[27] L. Ntaimo, J. A. G. Arrubla, C. Stripling, J. Young, and T. Spencer, *A stochastic programming standard response model for wildfire initial attack planning*, Canadian Journal of Forest Research, 42 (2012), pp. 987–1001.

[28] O. Y. Özaltin, O. A. Prokopyev, A. J. Schaefer, and M. S. Roberts, *Optimizing the societal benefits of the annual influenza vaccine: A stochastic programming approach*, Operations Research, 59 (2011), pp. 1131–1143.

[29] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., New York: Academic Press, 1969.

[30] R. T. Rockafellar and R. J.-B. Wets, *Scenarios and policy aggregation in optimization under uncertainty*, Mathematics of Operations Research, 16 (1991), pp. 119–147.

[31] A. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, 2006.

[32] J. Salmeron, R. K. Wood, and D. P. Morton, *A stochastic program for optimizing military sealift subject to attack*, Military Operations Research, 14 (2009), pp. 19–39.

[33] N. Shor, *Minimization Methods for Non-differentiable Functions*, Springer-Verlag, New York, 1985.

[34] S. Takriti, J. Birge, and E. Long, *A stochastic model for the unit commitment problem*, IEEE Transactions on Power Systems, 11 (1996), pp. 1497–1508.

[35] M. W. Tanner, L. Sattenspiel, and L. Ntaimo, *Finding optimal vaccination strategies under parameter uncertainty using stochastic programming*, Mathematical Biosciences, 215 (2008), pp. 144–151.

[36] B. Von Hohenbalken, *Simplicial decomposition in nonlinear programming algorithms*, Mathematical Programming, 13 (1977), pp. 49–68.

[37] J. Watson and D. Woodruff, *Progressive hedging innovations for a class of stochastic resource allocation problems*, Computational Management Science, 8 (2011), pp. 355–370.

Appendix A. Additional plots for PH vs. FW-PH for $\alpha = 1$.

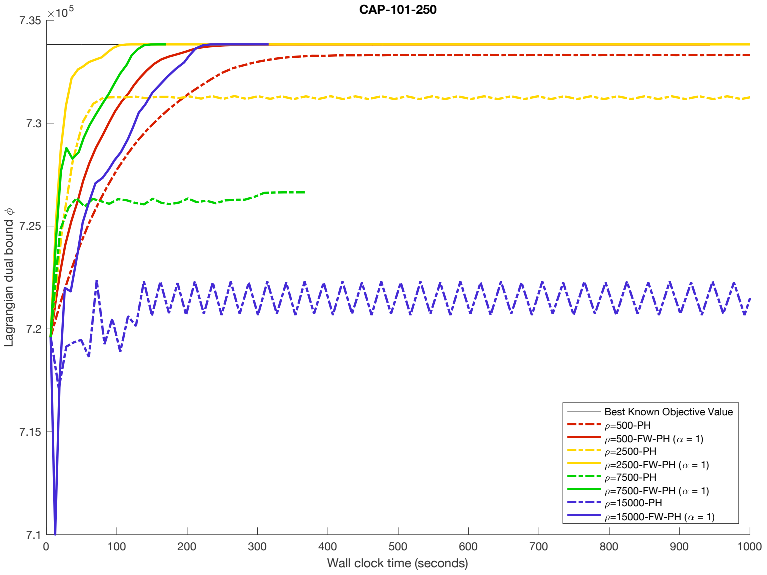

Fig. 5. *Convergence profile for CAP-101-250 (PH and FW-PH with* $\alpha = 1$*)*

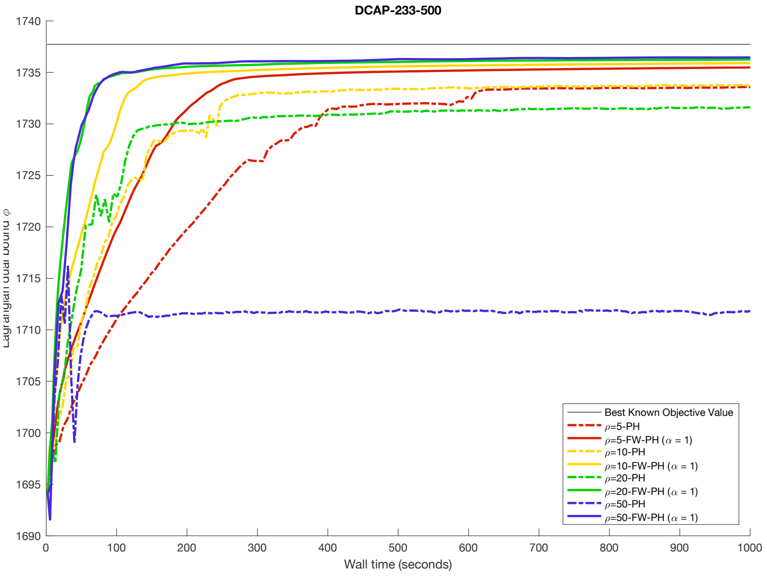

Fig. 6. *Convergence profile for DCAP-233-500 (PH and FW-PH with* $\alpha = 1$*)*

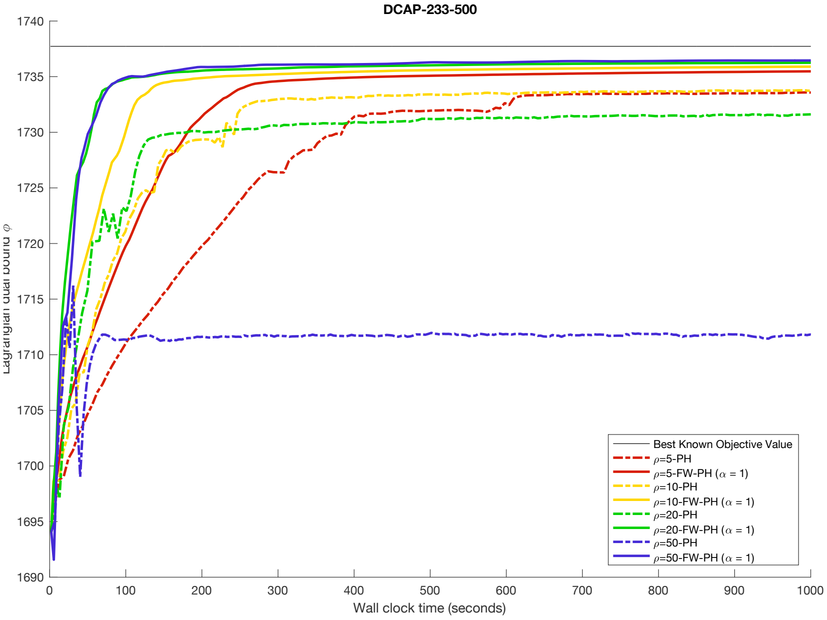

FIG. 7. *Convergence profile for SSLP-5-25-50 (PH and FW-PH with $\alpha = 1$)*

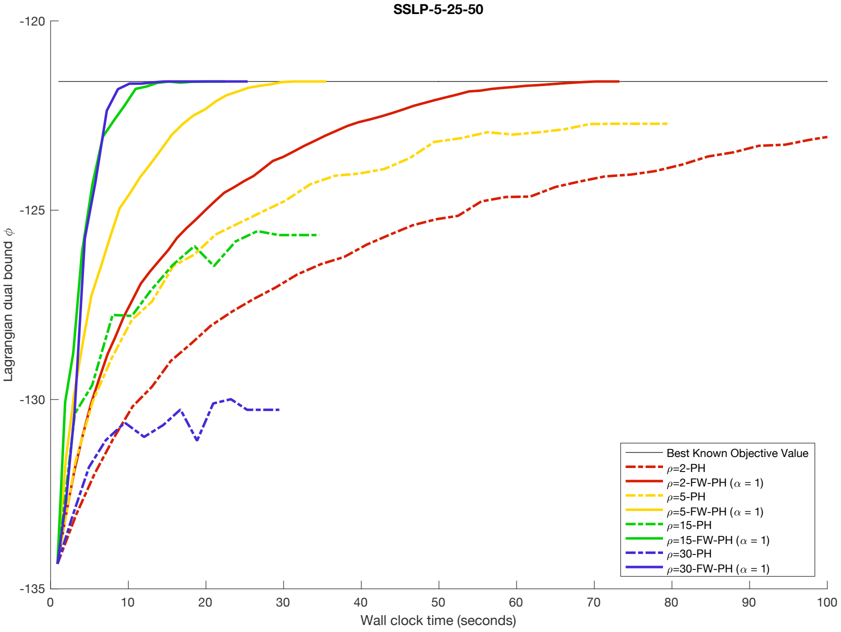

FIG. 8. *Convergence profile for SSLP-10-50-100 (PH and FW-PH with $\alpha = 1$)*