Noname manuscript No.
(will be inserted by the editor)

Progressive hedging for stochastic programs with cross-scenario inequality constraints

Ellen Krohn Aasgård · Hans Ivar Skjelbred

Received: date / Accepted: date

Abstract In this paper, we show how progressive hedging may be used to solve stochastic programming problems that involve cross-scenario *inequality* constraints. In contrast, standard stochastic programs involve cross-scenario *equality* constraints that describe the non-anticipative nature of the optimal solution. The standard progressive hedging algorithm (PHA) iteratively manipulates the objective function coefficients of the scenario subproblems to reflect the costs of non-anticipativity and penalize deviations from a non-anticipative, aggregated solution. Our proposed algorithm follows the same principle, but works with cross-scenario inequality constraints. Specifically, we focus on the problem of determining optimal bids for hydropower producers that participate in wholesale electricity auctions. The cross-scenario inequality constraints arise from the fact that bids are required to be non-decreasing. We show that PHA for inequality constraints have the same convergence properties as standard PHA, and illustrate our algorithm with results for an instance of the hydropower bidding problem.

Keywords Progressive hedging · Stochastic programming · Hydropower · Unit commitment · Electricity auctions

1 Introduction

We start with similar situation as described by [1] and [2]: we are confronted with a very large or difficult (possibly mixed integer) stochastic program for

Ellen Krohn Aasgård
Norwegian University of Science and Technology
Alfred getz vei 3
7034 Trondheim. Norway
E-mail: Ellen.Aasgard@ntnu.no

Hans Ivar Skjelbred
SINTEF Energy Research

which there already exists effective techniques for solving individual scenario subproblems, i.e. for the deterministic version of the program. However, as opposed traditional stochastic programming problems where scenarios are linked by equality constraints that represent non-anticipativity, we consider a class of stochastic programming problems where scenarios are linked by *inequalities*. This paper adapts the well-known progressive hedging algorithm of Rockafellar and Wets, which has traditionally been applied to problems wherein a subset of decisions must be identical in all scenarios, to a problem wherein inequality constraints link decisions made in different scenarios.

Specifically, we work with the problem faced by power producers that sells their output to wholesale electricity auctions. The producer must submit a supply curve as a set of price-volume points that describe a non-decreasing bid curve. Determining these bids may be formulated as a stochastic program, as described in [3] for hydropower producers participating in the Nordic market. In the rest of this paper, we will use the hydropower bidding and scheduling problems as described by [3] as the basis of our modelling, but empahasize that the algorithm is general and may be applied to other problems where there is logical relationship between variables in different scenarios. The hydropower bidding problem will be large due to the physical dimensions of the production system and the number of scenarios necessary to build a good bid curve. The problem may also involve integer variables due to unit commitment. Currently, operators in the Nordic hydropower industry do not determine bids by solving the true stochastic program, but rather multiple runs of a well-established deterministic model that determines optimal production schedules. From the starting point of having a good deterministic model, we develop a solution method that utilizes the benefits of stochastic optimization, but still keep calculation time low.

An important concern in stochastic programming is non-anticipativity or implementability, which means that the final solution cannot depend on any particular scenario realization. In [1], implementability is defined as the fact that if two scenarios s and s' are indistinguishable at time t on the basis of information available about them at time t, then their solutions should be equal. Non-anticipativity thus leads to equality constraints between scenarios in stochastic programs. As PHA requires independent subproblems, these cross-scenario constraints are removed from the formulation and instead the objective function coefficients of the subproblems are iteratively changed to reflect the costs of non-anticipativity and penalize deviations from a non-anticipative, aggregated solution.

Our problem involves inequality constraints between scenarios rather than equality constraints. The inequalities arise from the requirement of a non-decreasing bid curve in the hydropower bidding problem. Producers are expected to be willing to supply more power as the prices increase. However, due to time-dependencies and other characteristics of power production and river chains, the increasing nature of bid curves must be explicitly included in the optimization model.

The dynamics of PHA is to solve perturbations of scenario subproblems in iterations, where the objective function coefficients of the scenario problems are updated based on the distance between the current scenario solution and an aggregated solution. For standard PHA, the aggregation operator is to calculate the average solution over all scenario solutions. We show that a version of PHA also may be applied to problems with inequality constraints, as long as an appropriate aggregation operator and definition of implementability is used. For the case of hydropower bidding, implementability is defined as a non-decreasing bid curve, i.e. that larger or equal volumes are offered for higher prices. The aggregation operator thus calculates a non-decreasing bid curve based on the scenario solutions.

This paper shows how standard PHA and PHA for inequality constraints may be applied to the hydropower scheduling and bidding problems, respectively. For the scheduling problem or unit commitment problem, which involves equality constraints, standard PHA is a well suited method and has been applied successfully by other researchers, for instance [4–8]. Adaptations and improvements for standard PHA are reported by, among others, [2], who propose improvements for efficient selection of parameter values and acceleration of convergence, and [9] who describe how PHA may be solved more efficiently by considering bundles of scenarios. [10] show how lower bounds may be calculated during PHA iterations. For the bidding problem, which involve inequality constraints, we do not find many applications of PHA. However, [11], states that PHA is "not the only (method) that could be viewed as an implementation of the principle of scenario aggregation". We present such a method for stochastic programs with inequality constraints. The method is presented side-by-side with standard PHA, and we show that the convergence properties of PHA for inequality constraints is similar to that of standard PHA.

The rest of this paper is organized as follows: In Section 2, we discuss the scheduling and bidding problem for hydropower producers, before the mathematical models for both problems are presented in Section 3. In Section 4, the algorithms for standard PHA and PHA for inequality constraints are presented and compared. The convergence properties of both algorithms are discussed in Section 5. Results for standard PHA and PHA with inequality constraints applied to an instance of the hydropower scheduling and bidding problems are presented in Section 6. Conclusions are drawn in Section 7.

2 The hydropower scheduling and bidding problem

Over the last 20 years, wholesale electricity auctions have become a cornerstone in modern, deregulated power systems. Producers and consumers are required to submit price-volume bids indicating their willingness to sell and buy power for the next trading period. We only consider supply-side bids in this work, and focus on producers of hydropower in particular. The algorithm we present is however applicable to any type of power producer. In fact, with small adaptations, the algorithm may be used for any market agent that is required to submit a non-decreasing or non-increasing bid curve that is based on schedules of future production.

Determining market bids is an important aspect of day-to-day operations for power companies. We define the bidding problem as the task of determining bids for the next trading period, based on the current system state and forecasts for future prices and inflows. In the case of a day-ahead market, the next trading period will be the next day, while for shorter-term markets the next operating period will be a few hours ahead depending on the rules of the market. In the rest of this paper we will use the context of a day-ahead market.

The bidding problem is intimately linked to the scheduling problem, which we define as determining a production schedule, given an initial system state and forecasts for future prices and inflows. The reason why the scheduling and bidding problems are so tightly linked, is because the bids sent to the market should result in feasible operating schedules. For hydro, there are environmental, hydraulic and technical constraints that must be fulfilled to ensure safe and legal operation of multi-reservoir, cascaded river chains. Some of these constraints involve time-dependencies, such as river flow time delays and start-up or shut-down costs. Start-ups and shut-downs are also important for thermal producers, and is considered in the unit-commitment problem, which is defined as the problem of determining which generating units are on and producing in any given time step. The unit-commitment problem is an integral part of both the scheduling and bidding problem, and necessitates binary variables in the formulation for correct modelling of the on/off state of generating units. The difference between the scheduling and bidding problems lies mostly in the result we are after; a production schedule in the scheduling problem and a set of bids in the bidding problem, see the right part of Figs. 1 and 2, respectively. In the scheduling problem, the production schedule of the first day cannot depend on any particular scenario realization tomorrow, so we require that the production y_{ts} is equal across scenarios. In the bidding problem, the bids must have non-decreasing volumes for increasing prices, so we require that the production volume y_{ts} is less than or equal to the next element in a list where the scenarios are sorted by increasing prices.

From the view of a single power producer, both the scheduling and bidding problems involve uncertainty regarding the future market price, i.e. the outcome of the market, as well as inflows, i.e. future resource availability. Uncertainty makes stochastic programming a promising method for optimal scheduling and bidding, and has been applied by several researchers [3,12–16]. The benefits of stochastic optimization is, among others, more robust schedules, which are important in the case of inflow uncertainty, and being able to exploit price peaks and therefore increase profits. Comparisons between stochastic and deterministic models for hydropower scheduling and bidding are reported in [3, 14, 17, 18], which all show that stochastic programming yields improvements. For bidding, price uncertainty or a range of prices is a prerequisite to be able to formulate a model that results in optimal bids.

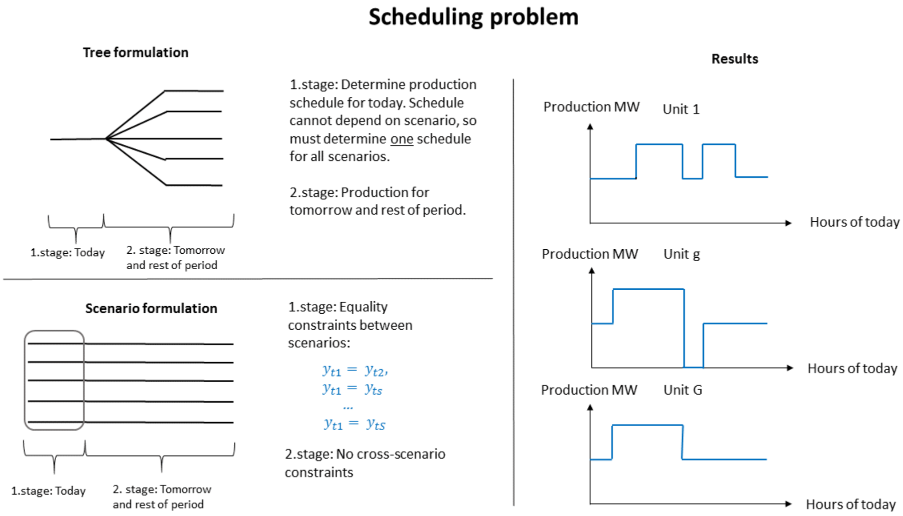

Fig. 1 Upper left: We define a two-stage scenario tree that have the decisions for today in the first stage and decisions for tomorrow and rest of the period in the second stage. Lower left: Problem structure of the deterministic equivalent used in PHA. The grey boxes indicate where cross-scenario constraints are added. Right: An illustration of the result from solving the scheduling problem; production schedules for each generating unit in the system.

This is why bidding is based on multi-scenario runs of the deterministic model today.

However, for industrial operations and large data sets, models based on stochastic programming are notoriously hard to solve, especially if they also involve binary variables, which indeed are present in both the scheduling and bidding problem. To provide decision support for industry operations, models for optimal scheduling and bidding must be solved fast in order to give operators the answers they need, when they need it.

Currently, the Nordic hydropower industry use a deterministic model in their daily scheduling. The scheduling model considers state-dependencies and nonlinear effects typical and important for hydropower by successive linear programming, where mixed-integer or linear problems are solved in iterations [19, 20]. The bidding problem is solved by heuristics based on multiple runs of the deterministic scheduling problem, as explained in [18]. The deterministic model has been extended to a stochastic programming model in [15] and to formal optimization of bids in [16]. The aim of the current work is to determine algorithms that makes it possible to solve the stochastic scheduling and bidding problem in parallel by dividing the problem into subproblems that may be solved separately. A successful parallel implementation may reduce the runtime of the stochastic models, making them applicable in industry operations.

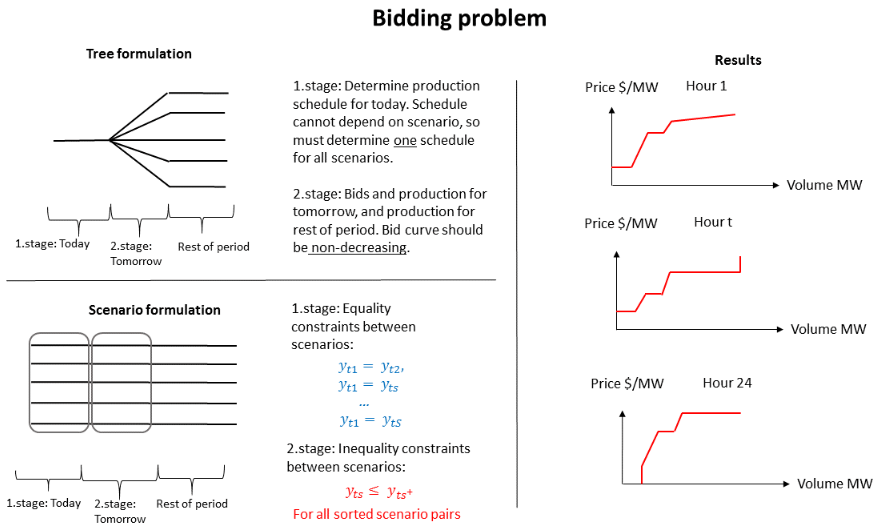

Fig. 2 Upper left: We define a two-stage scenario tree that have the decisions for today in the first stage and decisions for tomorrow and rest of the period in the second stage. Bids for tomorrow are determined before tomorrow's operations, but are calculated from possible production schedules for tomorrow. Lower left: Problem structure of the deterministic equivalent used in PHA. The grey boxes indicate where cross-scenario constraints are added. Right: An illustration of the results from solving the bidding problem; bid curves for each hour of tomorrow.

3 Mathematical formulation

Our mathematical formulation is based on [16], but is repeated here for completeness and to point out the differences between the scheduling and bidding problem. Focus is on the aspects of the model that are necessary to understand the differences between standard PHA and PHA for inequality constraints. Difficult topologies, optimization of pressure height and other aspects typical and important for hydropower scheduling are therefore not considered.

We define T to be the set of time steps within the model horizon. For consistent linking to longer-term scheduling models and to avoid end-of-horizon effects, both the scheduling and bidding problems have a horizon of about one week even though only the decisions for the first day are implemented. We define S to be the set of scenarios. Each scenario occurs with a known probability π_s and contain possible values for future prices and inflow. We consider a river chain with R reservoirs, and assume for simplicity that all reservoirs have an associated generating unit. The objective is to maximize the revenues from selling power within the model horizon and the value of water left in the reservoirs at the end of the horizon, less any costs related to start-ups. Revenues are calculated as the price of electricity (ρ_{ts}) times the sold volume (y_{ts}). Price is considered to be a stochastic variable. V_r describes the marginal value of storage (s_{tsr}), and may be interpreted as the marginal cost of production. In this work, the value of storage is a known input to the scheduling and bidding models, assumed to be calculated from longer-term hydropower scheduling models. Each time a generating unit is started, the variable d_{tsr} will take on a value of 1 and a cost of C_r^{Start} is incurred. The objective may thus be expressed by

$$max \sum_S \pi_s (\sum_T \rho_{ts} y_{ts} + \sum_R V_r s_{Tsr} - \sum_{TR} C_r^{Start} d_{tsr}) \tag{1}$$

Storage in reservoirs must adhere to a mass balance equation between time steps, as expressed by

$$s_{tsr} = s_{t-1sr} - q_{tsr}^P - q_{tsr}^B - q_{tsr}^S + \sum_{j \in R^{Up}} (q_{tsj}^P + q_{tsj}^B + q_{tsj}^S) + \Gamma_{tsr}. \tag{2}$$

Eq. (2) states that the storage level at the end of this hour, s_{tsr} , must equal the storage level at the end of the previous hour less any discharge used for production (q_{tsr}^P), bypass (q_{tsr}^B), or spillage (q_{tsr}^S) within this hour. The system consists of a series of connected reservoirs, and any discharge from connected upstream reservoirs, $j \in R^{Up}$, must be added to the storage level of this reservoir. Inflow, Γ_{tsr} , which may be stochastic, is also included in the mass balance equation. Reservoir levels also have to be within known maximum and minimum bounds.

The release of water is linked to power production by a piecewise linear concave production function for each generating unit. This production function has M line segments, each represented with a discharge volume q_{tsrm} limited by an upper bound Q_{rm}^{Max} and a power output rate E_{rm} . The sum of power produced from all segments must equal the power produced and requirements for minimum generation level, P_r^{Min} , if the unit is on. This is expressed by

$$\sum_M E_{rm} q_{tsrm} = y_{tsr} + P_r^{Min} u_{tsr}. \tag{3}$$

Here, u_{tsr} is a binary variable that describes the on/off decision of each generating unit. The sum of discharge through the segments is equal to the production discharge from this reservoir,

$$\sum_M q_{tsrm} = q_{tsr}^P. \tag{4}$$

The binary variable u_{tsr} is also used to determine if a start-up has occurred between two consecutive time steps. If so, the auxiliary variable d_{tsr} takes on a value of 1 according to

$$d_{tsr} \geq u_{tsr} - u_{t-1r} \tag{5}$$

and a cost is incurred in the objective function. Finally, the total volume produced from the system is the sum of production from the individual reservoirs,

$$y_{ts} = \sum_R y_{tsr} \tag{6}$$

Eqs. (1)-(6) describe the basic production scheduling aspects of the hydropower scheduling and bidding problems. Given a set of price or combined price-inflow scenarios with known probability, these equations determines an optimal volume to produce in each hour of each scenario. However, as explained in Section 1, we must require that the production schedule does not anticipate and adapt to a certain scenario. Therefore, we must chose a production schedule that is best for all scenarios, which is already indicated by averaging the objective function over scenarios in Eq. (1). However, the production schedule should not only be best on average, it should be equal for all scenarios. In Eq. (6), it is evident that both the total production y_{ts} and the production per reservoir y_{tsr} is dependent on scenario. To obtain a solution that is truly nonanticipative, we must require that all scenario solutions are equal. There are several ways to formulate this, and we have chosen to use the first scenario as reference and require the production in all other scenarios to be equal production in the first scenario, as in

$$y_{t1r} = y_{t2r} \ , \ldots , \ y_{t1r} = y_{tsr} \ , \ldots , \ y_{t1r} = y_{tSr}. \tag{7}$$

Note that the non-anticipativity constraints are imposed on unit level and not for the total production, that is, on y_{tsr} instead of y_{ts}. If the restrictions in Eq. (7) had been applied to the total production, the optimization would still be free to allocate the total volume differently among generation units depending on scenario. For large, multi-reservoir river chains this might lead to very different production schedules.

To summarize, Eqs. (1)-(7) describe a stochastic programming model for hydropower scheduling. The left part of Fig. 1 shows the scenario tree structure for the scheduling problem, both when formulated as a tree formulation, or as the deterministic equivalent used in PHA. The only constraints that link scenario problems is the non-anticipativity constraints on y_{tsr} as expressed in Eq. (7). Non-anticipativity is only imposed on for the first-stage production variables, i.e. for the hours where we require an implementable schedule.

The formulation for the bidding problem builds on the formulation of the scheduling problem in Eqs. (1)-(6). However, we will now show that the bidding problem also involves inequality constraints that link otherwise independent scenario subproblems.

In [3], a stochastic optimization model that determines optimal bids for hydropower is formulated. Their model use a scenario tree to describe stochastic prices and inflow, and consider how market commitments are determined for an individual producer. They consider the Nordic market setting where bid curves are piecewise linear and described by a set of price-volume pairs, $(P_i, x_i,)$. The commitment volume y_{ts} is found by interpolation between neighbouring points on the producer's bid curve and the market price. This is expressed as

$$y_{ts} = \frac{\rho_{ts} - P_{i-1})}{(P_i - P_{i-1})} x_{ti} + \frac{(P_i - \rho_{ts})}{(P_i - P_{i-1})} x_{ti-1} \qquad \text{if} \ \ P_{i-1} \leq \rho_{ts} \leq P_i, \tag{8}$$

where y_{ts} is the commitment volume and ρ_{ts} is the market clearing price. ρ_{ts} is only known after market clearing, so in the bid optimization problem the producer use a set of scenarios to describe possible prices for tomorrow. To avoid the nonlinear formulation that would result if both x_{ti} and P_i were variables, a set $i \in I$ of fixed price points P_i for which to bid is determined prior to the optimization. The optimization then determines optimal bid volumes, x_{ti}, for each price point. A constraint to make sure that the bids are non-decreasing must also be added,

$$x_{ti-1} \leq x_{ti}. \tag{9}$$

where the index i correspond to price scenarios sorted in increasing order for each hour. As stated in Section 1, it is natural for a producer to be willing to supply a larger volume at higher prices. However, due to time-dependencies and other characteristics of power production and river-chains, ensuring a non-decreasing bid curve must be explicitly formulated in the optimization model.

The model in [3] further includes production scheduling aspects to determine how the commitment volume y_{ts} should be produced from the generating units in the system. This is similar to Eqs. (1)-(6) in the scheduling problem above.

The model in [3] has been implemented, with slight modifications, in the framework of models that is used by most hydropower producers in the Nordic region [20]. In terms of bid optimization, the difference from [3] is that the set of fixed price points P_i are chosen from the scenario prices ρ_{ts}. This leads to a formulation that involves inequalities between scenarios. To see this, consider the case when the price $\rho_{ts} = P_i$. Eq. (8) is then reduced to its first term

$$y_{ts} = \frac{\rho_{ts} - P_{i-1}}{P_i - P_{i-1}} x_{ti} = \frac{P_i - P_{i-1}}{P_i - P_{i-1}} x_{ti} = x_{ti}. \tag{10}$$

So we are left with $y_{ts} = x_{ti}$. Recognizing that the index s for scenarios and i for price points and are interchangeable when we use the scenario prices as bid price points, we see that, according to Eq. (10), the commitment volume y_{ts} and the bid volume x_{ti} are equal. In this way, the bidding problem may be solved as a set of nearly independent scenario problems that determines production schedules and therefore the production/bid volumes y_{ts}. The only constraint that connects the different scenarios is the fact that we need a non-decreasing bid curve, as expressed in Eq. (9) for the model in [3]. Writing the equivalent constraints for our current formulation problem yields

$$y_{ts-1} \leq y_{ts}, \tag{11}$$

where scenarios are sorted in increasing order for each hour. As the price scenarios may cross each other, it is not certain that the ordering of scenarios is equal from hour to hour, see Fig 4.

To be able to correctly impose the restrictions in Eq. (11), we include in our formulation restrictions of the form

$$y_{ts} \leq y_{ts}^{+} \qquad \text{if} \;\; \rho_{ts} \leq \rho_{ts}^{+}. \tag{12}$$

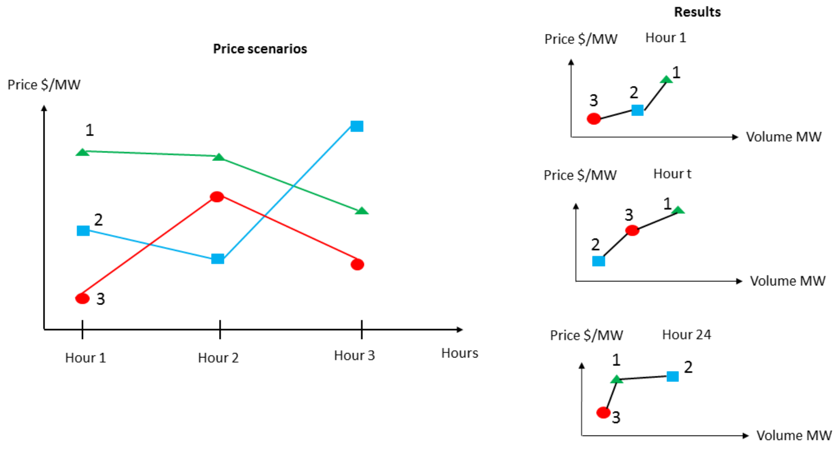

Fig. 3 Left: An illustration of the time profile of three price scenarios over three hours. In the first hour, Scenario 3 (red) has the lowest price and thus should have the lowest production volume. So, for the first hour, we require that production in Scenario 3 must be lower than in Scenario 2 (blue), and production in Scenario 2 must be lower than in Scenario 1 (green). In the next hour, the order is changed, and the production in Scenario 2 should be lower than in Scenario 3, which again should be lower than in Scenario 1. In the final hour, production in Scenario 3 should be lower than in Scenario 1, which should be lower than in Scenario 2. Right: An illustration of the results from bid optimization, comparable to Fig 2. Bid curves are piecewise linear and given by a set of price-volume points. The price and optimized volume in each scenario gives a point on the bid curve, as seen here for the scenarios to the left.

Scenario s^+ is defined as the scenario with the lowest price that is still greater or equal to ρ_{ts} in that hour. In the case where $\rho_{ts} = \rho^+_{ts}$, the production is required to be equal in the two scenarios. Further details about the implementation of the bidding model may be found in [16].

To summarize, the bidding model is defined by Eqs. (1)-(6), refnonanti and (12), where the constraints in Eq. (7) are imposed for the first stage production variables, and Eq. (12) is imposed for the hours for which we want to determine bids. Notice that the bid constraints link scenarios by inequality constraints, as opposed to equalities in the non-anticipatively constraints. The constraints in Eq. (12) are the only constraints that are added in order to go from scheduling to bidding. The reason for using this formulation instead of the formulation by [3] that uses Eqs. (8) and (9), is to have a closer fit with the scheduling model. With our formulation, there is no need for additional variables for bidding (i.e. the x_{ti} in Eq. (8)), we simply add a set of constraints to the scheduling model. The result is that, for each hour, the total production volume, y_{ts}, in each scenario, will give a point on the bid curve. Each bid point will be specified by the scenario price, ρ_{ts} and the corresponding optimized volume y_{ts}, see the right side of Fig. 3. The points need to specify a non-decreasing curve, which is ensured by Eq. (12).

Also notice that while the non-anticipativity constraints in Eq. (7) are defined per unit, i.e. on y_{tsr}, the inequality constraints for bidding is defined on system level, i.e. on y_{ts}. The reason is that when determining bids, the producer still have flexibility regarding how to allocate market commitments among units after commitments and price becomes known in the market clearing. Our modelling of the bidding problem maintains this flexibility.

4 Algorithm descriptions

PHA works with the deterministic equivalent formulation of stochastic programs, in which there is one copy of each variable belonging to every scenario. For the scheduling problem, this formulation is illustrated in Fig. 1, where the grey box indicate the non-anticipativity constraints.

The bidding problem, which involves inequality constraints, does not fit directly in to standard PHA. However, seeing the similarities with the scheduling problem, we propose a PHA-inspired method to tackle the bidding problem. Fig. 2 illustrates the deterministic equivalent formulation of the bidding problem. The grey boxes illustrate that there exists two types of cross-scenario constraints in the model; in the first stage, we impose non-anticipativity through equality constraints. In the second stage, we impose the non-decreasing bid curve constraints as defined in Eq. (12) for the hours we want to determine bids.

The algorithms for standard PHA and PHA with inequalities are presented in Algorithm 1 and 2, respectively. Both algorithms take a penalty factor α and a termination threshold ϵ as the sole input parameters. The presentation of the algorithms follow the notation in [2], except that we use y for the first stage variables (production) and x for second stage variables (everything else) to better comply with the presentation of the mathematical models in Section 3. Thus, $c_s y_s$ is the objective function for the first stage, while $f_s x_s$ is objective for the second stage. F_s is the feasible region for scenario s. In the implementation of both algorithms, we follow the suggestion of [2] and use a variable-dependent penalty parameter instead of a fixed value.

The algorithms in Algorithm 1 and 2 look quite similar and both decompose the problem by scenarios. PHA aggregates the solutions from scenario subproblems into an overall solution that is implementable for the underlying stochastic program. For standard PHA, given in Algorithm 1, the aggregation operator in Steps 3 and 7 is taking the average over all scenario solutions. This results in an aggregated solution $\overline{y}^{(k)}$ that does not depend on scenario. The dual variables, $w_s^{(k)}$ are calculated for each scenario based on the difference between the scenario solution $y_s^{(k)}$ and the average solution $\overline{y}^{(k)}$, see steps 4 and 8. If the scenario value is larger than the average, the dual variable will take on a positive value (since $\alpha \geq 0$), which in the next iteration will contribute to a lower solution value $y_s^{(k)}$ (given a minimization problem). If the scenario value is lower than the average, the dual will take on a negative value, contributing to a higher solution value for $y_s^{(k)}$ in the next iteration. In addition to the duals, deviations from the current aggregated solution are penalized by the quadratic term in the PHA objective function in step 6 of the algorithm.

A variant of PHA may be used also for problems with inequality constraints, as long as an appropriate aggregation operator and definition of implementability is used. In our case, implementability is defined as a non-decreasing bid curve. For PHA with inequality constraints, given in Algorithm 2, this means that the implementable solution $\hat{y}_s^{(k)}$ may depend on scenario as long as the non-decreasing bid constraints are overheld. The aggregation operator in Steps 3 and 7 calculates a non-decreasing bid curve by reordering the scenario solutions. The rest of the algorithm is equal to standard PHA.

Algorithm 1 Standard PHA

1: $k \leftarrow 0$
2: For all $s \in S$, $y_s^{(k)} = argmin(c_s y_s + f_s x_s) : (y_s, x_s) \in F_s$
3: For all $s \in S$, $\overline{y}^{(k)} = \sum_S \pi_s y_s^{(k)}$
4: For all $s \in S$, $w_s^{(k)} = \alpha(y_s^{(k)} - \overline{y}^{(k)})$
5: $k \leftarrow k + 1$
6: For all $s \in S$,
$y_s^{(k)} = argmin(c_s y_s + w_s^{(k-1)} y_s + \alpha||y_s - \overline{y}^{(k-1)}||^2 + f_s x_s) : (y_s, x_s) \in F_s$
7: For all $s \in S$, $\overline{y}^{(k)} = \sum_S \pi_s y_s^{(k)}$
8: For all $s \in S$, $w_s^{(k)} = w_s^{(k-1)} + \alpha(y_s^{(k)} - \overline{y}^{(k)})$
9: $g^{(k)} = \sum_S \pi_s ||y_s - \overline{y}^{(k)}||$
10: If $g^{(k)} \geq \epsilon$, go to Step 5. Otherwise terminate.

Algorithm 2 PHA for inequality constraints

1: $k \leftarrow 0$
2: For all $s \in S$, $y_s^{(k)} = argmin(c_s y_s + f_s x_s) : (y_s, x_s) \in F_s$
3: For all $s \in S$ sorted by increasing prices, and $s' \in S'$ sorted by increasing volume, $\hat{y}_s^{(k)} = y_{s'}^{(k)}$
4: For all $s \in S$, $w_s^{(k)} = \alpha(y_s^{(k)} - \hat{y}_s^{(k)})$
5: $k \leftarrow k + 1$
6: For all $s \in S$,
$y_s^{(k)} = argmin(c_s y_s + w_s^{(k-1)} y_s + \alpha||y_s - \hat{y}_s^{(k-1)}||^2 + f_s x_s) : (y_s, x_s) \in F_s$
7: For all $s \in S$ sorted by increasing prices, and $s' \in S'$ sorted by increasing volume, $\hat{y}_s^{(k)} = y_{s'}^{(k)}$
8: For all $s \in S$, $w_s^{(k)} = w_s^{(k-1)} + \alpha(y_s^{(k)} - \hat{y}_s^{(k)})$
9: $g^{(k)} = \sum_S \pi_s ||y_s - \hat{y}^{(k)}||$
10: If $g^{(k)} \geq \epsilon$, go to Step 5. Otherwise terminate.

A closer look at the differences in the aggregation operator for standard PHA and PHA for inequality constraints is in order. For standard PHA, the aggregation operator is to take the average over all scenario solutions. This results in an aggregated, implementable solution $\overline{y}^{(k)}$ that is scenario independent. The individual scenario solutions will converge to the average solution over the PHA iterations, so we will be left with one solution to implement. Notice that we might as well have used the notation $\overline{y}_s^{(k)}$ to denote the aggregate solution, indicating that each scenario gets its own copy of the average solution. The individual scenario solutions would be compared to its own copy, but as all scenario copies have the same value, the value of the final solution would still be scenario independent and thus implementable.

For PHA with inequality constraints, each scenario also have its own implementable solution $\hat{y}_s^{(k)}$. The scenario solutions will converge to this solution over the PHA iterations. Note, however, that in the case of inequality constraints, we do not require the $\hat{y}_s^{(k)}$ to be equal for all scenarios. An implementable solution in the case of inequality constraints does not mean that all scenarios have the same solution; it rather means that the inequalities between scenarios are adhered to. In our case it means that the bid curve is non-decreasing, i.e. that the constraints in Eq. (12) are fulfilled. Therefore, in each aggregation step, we calculate an implementable solution, i.e. a non-decreasing bid curve, from the available scenario solutions by switching the order of scenario solutions. For each time step, we start with an ordered set S where scenarios are arranged by increasing prices. We define another ordered set S' where the scenarios are sorted according to increasing production volumes, i.e. according to the scenario solutions $y_s^{(k)}$. The sets S and S' have the same size as they both contain all scenarios in the problem, but the order may be rearranged depending on the values of the scenario solution, see Fig. 4 for an illustration. Using the two ordered sets of scenarios, the implementable solution for each scenario (indexed by price) is set equal to the solution value of the corresponding volume-sorted scenario.

The presentation so far has assumed that each PHA subproblem consists of a single scenario. In the implementation, however, it is possible to bundle any number of scenarios into the subproblems. A large problem might thus be solved as a set of independent subproblems each consisting of a set of scenarios of the original problem, with any cross-scenario constraints between the scenarios in each bundle present. This makes PHA a viable alternative when the original problem is too large to be solved as one problem. If the maximum number of scenarios that can be solved as one problem is 100, a problem with 1000 scenarios may still be solved as 10 subproblems with 100 scenarios each.

Current industry applications of the model is however likely to involve a smaller set of scenarios ($\leq$50), as the typical problem instance is already large due to the size of the physical production system. Large hydropower companies operate several interconnected river systems, and bid for their total portfolio of plants in a market area. The resulting stochastic problem could potentially be solved, but as operators are pressed for time in the bidding process, short calculation times are crucial. Hydropower companies are thus willing to invest in additional computing resources even for modest speed-ups. The re-

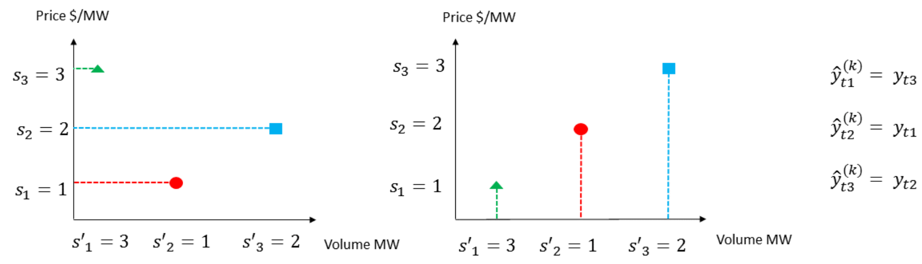

Fig. 4 An illustration of how an implementable bid curve is calculated in each iteration of PHA for inequality constraints. We consider a single hour and three scenarios. Scenarios are indexed by prices on the price axis and by bid volumes on the volume axis. To the left, where scenarios are sorted by increasing prices, the results do not represent a non-decreasing bid curve. The right plot shows the scenarios sorted by bid volume and this the non-decreasing, implementable solution. The resulting equations that determines $\hat{y}_s^{(k)}$ are seen to the right.

sults in the next section indicate that PHA achieves significant speed-ups for the scheduling problem but only moderate speed-ups for the bidding problem. The speed-ups increase as the model size increase. Industry applications of the scheuling and bidding problem are likely to involve larger production systems than the case in Section 6, and thus also larger problem instances. Feedback from users of the SHOP model is that the speed-ups observed so far may be useful in practice, especially for the scheduling problem.

5 Convergence

In [11], it is shown that PHA will converge to the optimal solution of the original stochastic program if

$$\sum_S \pi_s w_s^{(k)} = 0. \tag{13}$$

This is indeed true for standard PHA as we have

$$\sum_S \pi_s (w_s^{(k)} - w_s^{(k-1)}) = \alpha \pi_s \sum_S (y_s^{(k)} - \overline{y}^{(k)}) \quad \text{and} \quad w_s^{(0)} = 0. \tag{14}$$

The trick lies in recognizing that

$$\pi_s \sum_S (y_s^{(k)} - \overline{y}^{(k)}) = \sum_S \pi_s y_s^{(k)} - \sum_S \pi_s \overline{y}^{(k)} = \overline{y}^{(k)} - \overline{y}^{(k)} = 0, \tag{15}$$

as the average operator is idempotent. Further, as each scenario solution

$$y_s^{(k)} = argmin(c_s y_s + w_s^{(k)} y_s + \alpha ||y_s - \overline{y}_s^{(k-1)}||^2 + f_s x_s) : (y_s, x_s) \in F_s, \tag{16}$$

the average solution

$$\overline{y}^{(k)} = argmin \sum_S \pi_s(c_s y_s + w_s^{(k)} y_s + \alpha ||y_s - \overline{y}_s^{(k-1)}||^2 + f_s x_s) : (y_s, x_s) \in \bigcup_S F_s$$
$$= argmin \sum_S \pi_s(c_s y_s + \alpha ||y_s - \overline{y}_s^{(k-1)}||^2 + f_s x_s) : (y_s, x_s) \in \bigcup_S F_s, \quad (17)$$

where the previous result on $w_s^{(k)}$ is used in the last step. Assuming that the objective function is "nice" in the neighbourhood of $\overline{y}^{(k)}$, this in turn implies that

$$\overline{y}^{(k)} = argmin \sum_S \pi_s(c_s y_s + f_s x_s) : (y_s, x_s) \in \bigcup_S F_s \quad (18)$$

and therefore solves the original stochastic program in the convex case. The above proof is taken from [11]. A note should be made here about the feasible region for each scenario subproblem, F_s, in contrast to the feasible region for the total stochastic problem $\bigcup_S F_s$. A scenario solution is only guaranteed to be feasible for that particular scenario, while the final solution needs to be feasible for all scenarios. In the case of the hydropower scheduling and bidding problems, the feasible region is only affected by inflow uncertainty. Inflow occurs in the right-hand side of the reservoir balance constraints (Eq. (2)), and therefore affects the feasible regions of scenarios. However, the model always have the opportunity to spill excess water or obtain extra water at a high cost, so the feasible regions for all scenarios are in principle the same. Price is only present in the objective function, and therefore does not affect the feasible region.

The above paragraphs prove that standard PHA will converge to the optimal solution of the underlying stochastic program. If the equivalent of Eq. (13) is valid for PHA for inequality constraints, the rest of the convergence proof will follow. Similarly to standard PHA,

$$\sum_S \pi_s(w_s^{(k)} - w_s^{(k-1)}) = \alpha \pi_s \sum_S (y_s^{(k)} - \hat{y}_s^{(k)}) \quad \text{and} \quad w_s^{(0)} = 0. \quad (19)$$

so we need to show that

$$\pi_s \sum_S (y_s^{(k)} - \hat{y}_s^{(k)}) = 0. \quad (20)$$

Considering that the implementable solutions $\hat{y}_s^{(k)}$ is a rearrangement of the scenario solutions $y_s^{(k)}$, each individual scenario solution will appear one time with a positive sign and one time with a negative sign in the summation over all scenarios in Eq. (20). If all scenarios have equal probability, the scenario and implementable solutions will cancel each other out and the sum over all scenarios will be zero as needed. The convergence properties of PHA for inequality constraints are therefore the same as for standard PHA, as long as all scenarios have equal probability.

Table 1 Results for the scheduling problem

No Scenarios	5	20	35	50	64
No Variables	320,080	1,280,320	2,240,560	3,200,800	4,097,024
No Constraints	249,689	1,026,059	1,802,429	2,578,799	3,303,411
Objective stochastic	100.000	100.000	100.000	100.000	100.000
Objective PHA	99.998	99.999	99.998	99.997	99.998
Time stochastic	7.08	56.92	147.50	336.29	732.44
Time PHA	68.36	90.52	148.91	201.22	252.73

Equal probability for all scenarios is a limitation when considering that the scenarios in principle should approximate a distribution for the stochastic parameters. In practical applications, however, the number of scenarios used is often too small to in any way approximate a distribution, so the additional error of having equal probabilities is considered to be negligible. If the number of scenarios is larger, it is possible to solve PHA using bundles of scenarios as subproblems. It is then possible to choose these bundles such that each bundle has equal probability while still maintaining the true probabilities of the individual scenarios.

The convergence of standard PHA is proven only for linear programs, but promising results also for mixed-integer problems are reported by [2]. Both the scheduling and bidding problem involve binary variables due to unit commitment.

6 Case study and results

Results for PHA and PHA for inequality constraints is shown for a reservoir system with 17 reservoirs and 13 generating units. The total generation capacity is about 1000 MW. The time horizon is set to 168 hours and two-stage scenario trees as depicted in Figs 1 and 2 are used. We only consider price uncertainty in this work. In the scheduling problem, the main result is a production schedule for all units for the 24 hours of today. In the bidding problem, the main result is bid curves for the 24 hours of tomorrow. For both the scheduling and bidding problem, we solve the deterministic equivalent model as a full stochastic program and by the PHA algorithms presented in Section 4. PHA is meant to be used when the stochastic program is too large or complex to be solved within certain time constraints. Here, however, we want to ascertain that our new algorithm actually converges to the optimal solution of the stochastic program, so we limit our self to cases with a modest number of scenarios in order to be able to solve the full stochastic program.

Tables 1 and 2 show results for the scheduling and bidding problem, respectively. The objective function values are given as a percentage of the objective function value obtained by the full stochastic program. The objective function values obtained by PHA is very close to the true objective function value. Calculation time is given in seconds.

Table 2 Results for the bidding problem

No Scenarios	5	20	35	50	64
No Variables	290,976	1,163,976	2,036,976	2,909,976	3,724,776
No Constraints	213,004	867,919	1,522,834	2,177,749	2,789,003
Objective stochastic	100.000	100.000	100.000	100.000	100.000
Objective PHA	100.005	99.227	99.144	99.039	99.017
Time stochastic	9.30	71.56	231.98	348.31	958.20
Time PHA	39.39	107.34	295.81	388.73	734.69

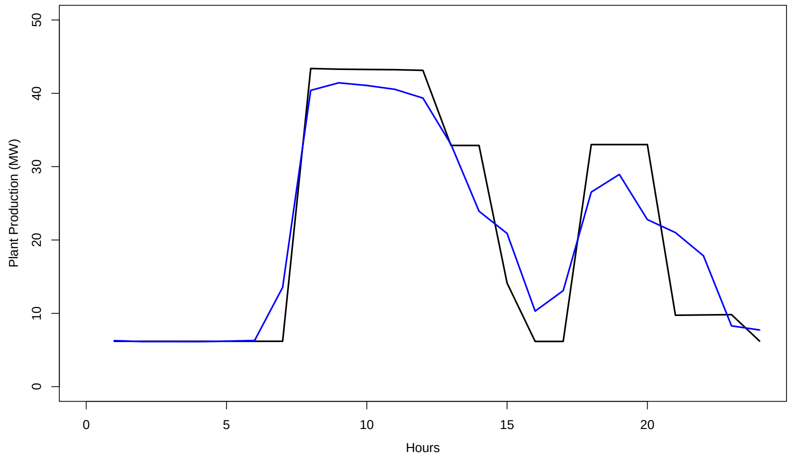

Fig. 5 The 24-hour production schedule for unit 4, as obtained from the full stochastic program (black) and standard PHA (blue).

We take a closer look at the results obtained from the 35-scenario case. For the scheduling problem, Fig. 4 shows the production schedule obtained from the full stochastic program and standard PHA for one of the generating units in the system. The solutions are quite similar, and follow the same general pattern with a morning and afternoon production peak. The solutions are however not identical, even though the objective function values are very close as seen from Table 1. This is because the objective function is relatively flat due to the flexibility of the production system. Results for the remaining units are comparable.

For the bidding problem, Fig. 5 shows the bid curves for all 24 hours of tomorrow, while Fig. 6 shows a single hour to better discern the differences. Again, the results are similar, but not identical even though the objective function values are close as seen in Table 2.

Regarding convergence, Figs. 8 and 9 shows plots of the distance calculated in Step 9 of the algorithms given in Section 4. This distance is used to decide whether to terminate the algorithms or continue with another iteration, and should approach zero. From the plots, it is evident that the scheduling problem has a smaller initial distance, which is driven to zero very fast, whereas the

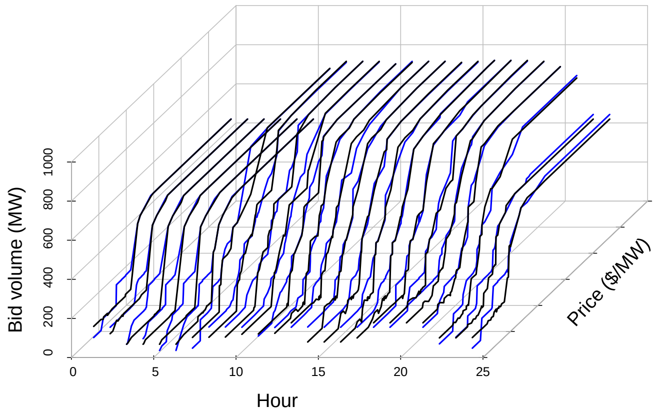

Fig. 6 Bid curves for all 24 hours of tomorrow, as obtained from the full stochastic program (black) and PHA for inequality constraints (blue). Values along the price-axis are removed for confidentiality.

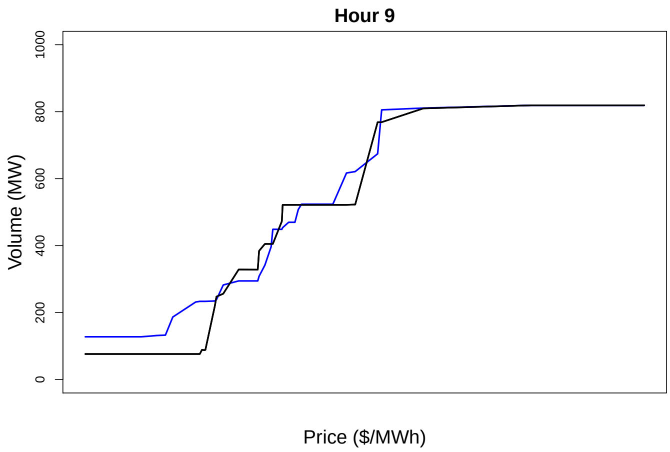

Fig. 7 Bid curves for Hour 9 of tomorrow, as obtained from the full stochastic program (black) and PHA for inequality constraints (blue). As seen from Fig. 6, the results for the other hours have similar differences. Values along the price-axis are removed for confidentiality.

bidding problem has a larger initial distance and uses several iterations to reduce it to zero. These results are however case dependent and we find other patterns of convergence for other problem instances or other values for the penalty parameter α. Our experience, however, is that the convergence of PHA for inequality constraints in the bidding problem is slower than for standard PHA in the scheduling problem.

The results presented so far indicate that standard PHA and PHA for inequality constrains give good results for the hydropower scheduling and bid-

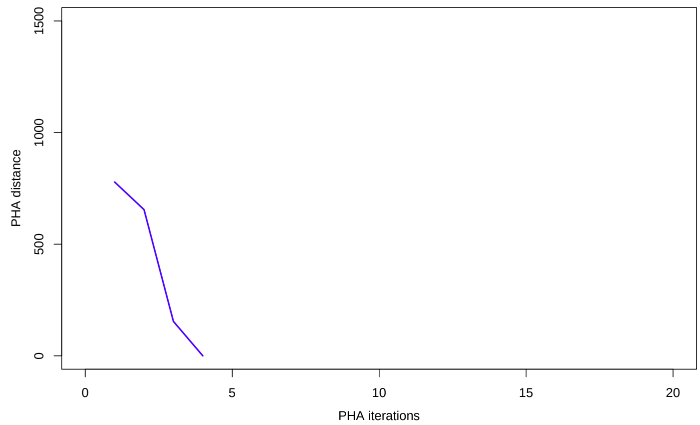

Fig. 8 The distance $\sum_S \pi_s ||y_s - \overline{y}^{(k)}||$ as calculated in Step 9 of standard PHA. .

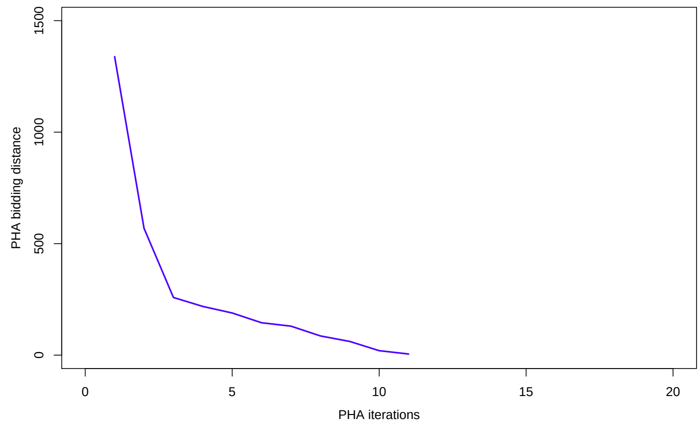

Fig. 9 The distance $\sum_S \pi_s ||y_s - \hat{y}_s^{(k)}||$ as calculated in Step 9 of PHA for inequality constraints.

ding problem, in that we are able to obtain solutions that are comparable to solving the full stochastic program. However, the reason for implementing PHA is to be able to solve subproblems in parallel in order to solve problems that are too large to be solved as a single problem or to speed up calculation time. For the cases considered here, we are able to solve the full stochastic program, so we consider the potential for speed-ups. Table 1 and 2 show the calculation time for the full stochastic program and PHA. For small problem instances, it is faster to solve the full stochastic programs than the parallell PHA algorithms due to overheads of decomposition. As the problem size increase, the calculation times of PHA is shorter than for the full stochastic program. The reduction in calculation time is substantial for standard PHA in the scheduling problem, whereas only a moderate reduction is observed for the largest case of PHA for inequality constraints in the bidding problem.

7 Conclusions

In this paper, we have shown that a variant of progressive hegding may be used to solve stochastic programming problems that involve cross-scenario inequality constraints. Standard progressive hedging solves large stochastic problems by iteratively manipulating the objective function coefficients of individual scenario subproblems to reflect the costs of non-anticipativity. Our algorithm follows the same principle, but works with cross-scenario inequality constraints. Specifically, we focus on the problem of determining optimal bids for hydropower producers that participate in wholesale electricity auctions. The cross-scenario inequality constraints arise from the fact that bids are required to be non-decreasing. We formulate a progressive hedging inspired algorithm for stochastic programs with inequality constraints and show that the algorithm has the same convergence properties as standard progressive hedging. Results for the new algorithm are demonstrated on a typical instance of the hydropower scheduling and bidding problem, and illustrate that we are able to obtain convergence to the optimal solution in a small number of iterations (≤ 15) of the PHA algorithm. In terms of calculation time, our results indicate that the parallell implementation standard progressive heding in the scheduling problem achives substantial reductions in calculation time, whereas reductions are moderate for progressive hedging for inequality constraints in the bidding problem.

Acknowledgements

This work was supported by the Research Council of Norway under project-number 255100/E20 MultiSharm.

References

1. R. T. Rockafellar and Roger J.-B. Wets, Scenarios and Policy Aggregation in Optimization Under Uncertainty, Mathematics of Operations Research, 16(1), 119-147, 1991
2. J.-P. Watson and D.L. Woodruff, Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems, Computational Management Science, 8(4), 355-370, 2011
3. S.-E. Fleten and T. K. Kristoffersen, Stochastic programming for optimizing bidding strategies of a Nordic hydropower producer, European Journal of Operational Research, 181(2), 916-928, 2007
4. S. Takriti, J. R. Birge and E. Long, A stochastic model for the unit commitment problem, IEEE Transactions on Power Systems , 11(3), 1497-1508, 1996
5. S. M. Ryan, R. J. B. Wets, D. L. Woodruff, Toward scalable, parallel progressive hedging for stochastic unit commitment, 2013 IEEE Power and Energy Society General Meeting, Vancouver, Canada, 2013

6. P.-L. Carpentier, M. Gendreau and F. Bastin, Long-term management of a hydroelectric multireservoir system under uncertainty using the progressive hedging algorithm, Water Resources Research, 49(5), 2812-2827, 2013
7. A. Helseth, Stochastic network constrained hydro-thermal scheduling using a linearized progressive hedging algorithm, Energy Systems, 7(4), 585-600, 2016
8. R.E.C. Gonçalves, E.C. Finardi and E. L. da Silva, Applying different decomposition schemes using the progressive hedging algorithm to the operation planning problem of a hydrothermal system, Electric Power Systems Research, 83(1), 19-27, 2012
9. T. G. Crainic, M. Hewitt and W. Re, Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design, Computers Operations Research, 43, 90-99, 2014
10. 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, 47-67, 2016
11. R.J.-B.Wets, The Aggregation Principle in Scenario Analysis and Stochastic Optimization, in Algorithms and Model Formulations in Mathematical Programming, Springer Berlin Heidelberg, Berlin, Heidelberg, 91-113, 1989
12. S. Séguin, S.-E. Fleten, P. Côté, A. Pichler and C. Audet, Stochastic short-term hydropower planning with inflow scenario trees, European Journal of Operational Research, 259(3), 1156-1168, 2017
13. D. De Ladurantaye, M. Gendreau, and J.-Y. Potvin, Strategic Bidding for Price-Taker Hydroelectricity Producers, IEEE Transactions on Power Systems, 22(4), 2007
14. S.-E. Fleten and T. K. Kristoffersen, Short-term hydropower production planning by stochastic programming, Computers & Operations Research, 35(8), 2656-2671, 2008
15. M.M. Belsnes, O. Wolfgang, T. Follestad and E.K. Aasgård, Applying successive linear programming for stochastic short-term hydropower optimization, Electric Power Systems Research, 130, 167-180, 2016
16. E.K. Aasgård , C. Øyn Naversen, M. Fodstad and H.I. Skjelbred, Optimizing day-ahead bid curves in hydropower production, Energy Systems, 2017
17. D. De Ladurantaye, M. Gendreau, and J.-Y. Potvin, Optimizing profits from hydroelectricity production, Computers & Operations Research, 36(2), 499-529, 2009
18. E. K. Aasgård, G. S. Andersen, S.-E Fleten and D. Haugstvedt, Evaluating a Stochastic-Programming-Based Bidding Model for a Multireservoir System, IEEE Transactions on Power Systems 29(4), 2014
19. O. B. Fosso and M. M. Belsnes, Short-term hydro scheduling in a liberalized power system, 2004 International Conference on Power System Technology, Singapore, 2004
20. SINTEF Energy Research SHOP website, accessed Aug 8, 2017, http://www.sintef.no/en/software/shop/