Available online at www.sciencedirect.com

ScienceDirect

Energy Procedia 87 (2016) 77 – 84

Energy Procedia

5th International Workshop on Hydro Scheduling in Competitive Electricity Markets

Short-term hydrothermal generation scheduling using a parallelized stochastic mixed-integer linear programming algorithm

Esteban Gil[a]*, Juan Araya[a]

[a]*Universidad Técnica Federico Santa María, Av. España 1680, Valparaíso, Chile*

Abstract

Short-term hydrothermal generation scheduling (STHTGS) is the optimization process through which decisions are made about the commitments of thermal generators and the allocation of hydro energy resources in the planning horizon (1 day to 1 week), while satisfying a large set of technical constraints. Uncertainty in this problem may appear in different modelling parameters, but the extended stochastic version of the STHTGS problem may lead to impractical solution times. This paper discusses the application of a parallelized stochastic mixed-integer linear program (SMILP) to solve the stochastic STHTGS problem. In order to decrease simulation time a scenario-based decomposition approach based on the progressive hedging (PH) algorithm is proposed. Computational experiments are conducted in two multi-processor nodes of a cluster for different numbers of stochastic scenarios. The algorithm is tested in the Chilean Central Interconnected System using a problem instance considering a weekly horizon with hourly resolution. Results show that the PH algorithm has good convergence properties, needing only a few iterations to converge. Furthermore, as PH generates similarly sized sub-problems, the parallel version of the algorithm scales up quite well as the number of scenarios is increased.

Peer-review under responsibility of SINTEF Energi AS

Keywords: Short-term hydrothermal generation scheduling; Unit commitment; Optimization; Stochastic programming; Progressive hedging

1. Introduction

Generation scheduling in hydrothermal power systems is a computationally challenging optimization problem whose purpose is to find the optimal allocation of thermal and hydro energy resources to minimize present and future operation costs. This work deals with short-term hydrothermal generation scheduling (STHTGS), where the

* Corresponding author. Tel.: +56-32-2654192; fax: +56-32-2654594.
E-mail address: esteban.gil@usm.cl

Peer-review under responsibility of SINTEF Energi AS
doi:10.1016/j.egypro.2015.12.360

decision variables are the unit commitments of thermal generators and the allocation of hydro energy resources in the planning horizon. Thus, STHTGS combines two power system optimization problems: the unit commitment problem (UC) [1] and the hydrothermal coordination problem (HC) [2].

STHTGS can be a very challenging problem. On the one hand, the unit commitments are integer decisions, making the optimization problem NP-complete. On the other hand, use of stored hydro energy in the current scheduling horizon will preclude its use in the future, thus increasing future operating costs and requiring coordination with medium and long-term operational planning models. Thus, optimal allocation of hydro resources must minimize the sum of the immediate operation costs (mostly fuel costs of thermal units) and the water cost (opportunity cost of using the water now instead of in the future), while satisfying a large set of technical constraints [2]. These constraints are related to energy balance in each bus, power flows in the transmission network, transmission losses modelling, technical operation limits of generation and transmission equipment, loading ramps, min-up times and min-down times, stability and security, primary and secondary reserve requirements, and water balance constraints for reservoirs and cascading hydro. Many different formulations and optimization methods have been proposed in the literature for the STHTGS problem (see [3] for a review).

Uncertainty in this problem is usually pervasive, as it may appear in different modelling parameters (e.g. demand, availability of hydro resources, wind/solar generation, and so on). Although uncertainty in the short-term operational planning horizon (1 day to 1 week) has traditionally been ignored, emergence of new technologies (e.g. wind and solar power, demand response, smarter grids) are now forcing system operators to modify their models to explicitly manage the uncertainty at the generation scheduling stage. The deterministic STHTGS problem is already quite complex and computationally intensive, and its extended stochastic version may lead to impractical solution times.

The objective of this work is to discuss the application of a parallelized stochastic mixed-integer linear program (SMILP) to solve the stochastic STHTGS problem in real-sized systems. The paper is structured as follows: Section 2 presents the mathematical formulation (based on SMILP) of the stochastic STHTGS; Section 3 discusses the decomposition of the stochastic STHTGS using the Progressive Hedging (PH) algorithm and the parallelization of the solution process; Section 4 provides details about the implementation of the algorithm; Section 5 describes the Chilean Central Interconnected System (*Sistema Interconectado Central*, SIC) on which the computational tests were conducted; Section 6 presents results focused on the computational performance of the parallelized stochastic SMILP formulation. Finally, Section 7 presents conclusions and directions for further research.

2. Problem formulation using SMILP

2.1. Deterministic Formulation of the STHTGS problem

The thermal generation costs for a single hour are shown in equation (1), where $P_{t,g}$ is generation for generator g; $Y_{t,g}$ is 1 if generator g is started in t; $Z_{t,g}$ is 1 if generator g is shutdown in time t. C^{op}, C^{on}, and C^{off} are the variable operation cost, the start cost and the shutdown cost for each generator, respectively.

$$y_t = \sum_{g\in G}\{C_g^{op}\cdot P_{t,g} + C_g^{on}\cdot Y_{t,g} + C_g^{off}\cdot Z_{t,g}\} \tag{1}$$

In the deterministic unit commitment problem (UC), the objective is to minimize the sum of the thermal generation costs and the cost of unserved energy during a certain horizon. When large hydro storages need to be considered, the STHTGS problem extends the UC problem using the following objective function:

$$Minimize\colon \quad \sum_{t\in T}\{y_t + \sum_{r\in R} FCF_r(Vol_{T,r}) + \sum_{n\in N} VoLL\cdot USE_{t,n}\} \tag{2}$$

where $Vol_{T,r}$ is hydro energy (in MWh) in reservoir r in the last period of the scheduling horizon, T; and $USE_{t,n}$ is the unserved energy in bus n. All hourly decision variables are indexed in time $t \leq T$. $VoLL$ is the value of lost load for the system. Hydro power plant efficiency is considered constant during the scheduling horizon.

The committed thermal generating units and the hydro generators must be able to meet the system load at every hour during the scheduling horizon. The cost minimization is also subject to a number of operating, transmission, security, and hydro constraints (see [1, 2] for details), including cascading hydro constraints. The transmission system is modelled as a DC power flow with all the transmission constraints formulated upfront.

If stored hydro energy is used during the current scheduling horizon, future operating costs will increase. In order to decouple long/mid-term decisions from the short-term hydrothermal coordination activity, hydro energy has a *Future Cost Function* FCF_r for each reservoir *r,* exogenously given by medium- and long-term models (see [2]).

2.2. SMILP formulation of the stochastic STHTGS problem

In SMILP [4,5], the uncertainty of the input variables is represented through a finite number of scenarios (*S*). Let define $\mathbf{Q}_s$ as the set of all constraints for scenario *s*. If we now group for each scenario *s* all first-stage decision variables in vector $\mathbf{x}_s$ and the second-stage variables in vector $\mathbf{y}_s$, the objective function in equation (2) can be rewritten for each scenario *s* as equation (3).

$$Minimize\colon \{c_x \cdot x_s + c_y \cdot y_s\} \mid x_s, y_s \in Q_s \tag{3}$$

Then, the deterministic formulation of the STHTGS is replicated for each of the *S* scenarios (with individual probability w_s) to define *S* sub-problems that are then linked in a master problem through an additional set of constraints called non-anticipativity constraints. The non-anticipativity constraints force those decisions than need to be made now (first-stage variables, that is thermal unit commitments and final volumes in the hydro storages) to be the same for all scenarios, as equation (4) shows.

$$Minimize\colon \sum_{s\in S} w_s \cdot \{c_x \cdot x_s + c_y \cdot y_s\} \mid x_s, y_s \in Q_s \ \forall s \in S$$

$$subject\ to \quad x_i = x_j \ \forall i, j \in S \tag{4}$$

The resulting optimization problem is a two-stage SMILP whose size is approximately *S*-times the original deterministic problem. Notice that the objective function is the expected value of the total system costs.

3. Problem decomposition using PH

Due to the large size of stochastic programming applications, decomposition methods are usually required. For a review of different decomposition methods applied to stochastic programming, see [6,7]. The Progressive Hedging (PH) algorithm was proposed by Rockefeller and Wets in [8] in order to allow scenario-based decomposition of large stochastic linear programs. If the solution times for each scenario are similar, the problem is well-suited for parallelism.

To solve the stochastic STHTGS in a reasonable time, a scenario-based decomposition approach based on the progressive hedging (PH) algorithm is proposed and used in this work. PH has recently been successfully used for the stochastic unit commitment problem [9-11] and for medium-term operational planning of hydrothermal systems [12].

PH is an augmented Lagrangian decomposition method, as it relaxes the non-anticipativity constraints of the SMILP while adding penalty terms to the objective function. This creates similarly-sized sub-problems, lending itself nicely to parallelism and high-performance computing (HPC). After solving all sub-problems, before starting a new iteration the PH algorithm updates the multipliers associated to the non-anticipativity constraints based on the distance between the first-stage variables. The process is repeated until all first-stage variables are identical.

First, an optimal solution is obtained independently for each scenario, without enforcing the non-anticipativity constraints. Then, a candidate non-anticipative solution is proposed as the average of the independent solutions. Next, a vector of multipliers associated to the non-anticipativity constraints are updated. These multipliers are included in the augmented objective function, together with a quadratic proximal term to accelerate convergence to a non-anticipative optimal solution. The algorithm stops when the first-stage variables (unit commitments of thermal units) for all scenarios converge to the average, that is, when the non-anticipativity constraints are satisfied. In our tests for the STHTGS problem, the algorithm may converge in as little as two iterations, although this may increase for different sets of scenarios and for different hydrological conditions. The convergence criterion is set by the size of the distance between the vectors of the first-stage variables.

Notice that in each iteration the sub-problems are solved independently, communicating with the master problem only when they have finished. Thus, the formulation, parametrization, solution, and result extraction for each sub-problem can be conducted in parallel.

4. Parallel implementation

The algorithm is implemented in Fortran 95 [13, 14] using the GNU compiler for x64 architecture. Fortran is a very popular programming language for HPC applications [15]. The solution of each sub-problem is obtained using CPLEX [16], using 2 cores per thread.

As in our implementation each parallel task (the solution of each sub-problem) is similar (only its parameterization is different), a domain decomposition is performed (Single Instruction Multiple Data, SIMD). Parallelization is conducted using a hybrid OpenMP [17] and MPI model, with Message Passing Interface (MPI) protocol through Open MPI [18] between the nodes and shared memory programming using OpenMP inside each node.

The computational experiments are conducted in two multi-processor nodes of a cluster. Each node has 2 Intel E5 Xeon processors with 6 cores each, so each of the two nodes has 12 cores available. As indicated before, CPLEX uses 2 cores per thread, so each node can run up to 6 CPLEX threads simultaneously.

We tested both a shared memory architecture (using a single node) and a distributed memory architecture (using both nodes of the cluster). In the shared memory architecture all processors have access to the same physical memory, while in the distributed memory architecture network communications are used to access memory on the different machines where tasks are executing.

First, the program reads the input files and parametrizes the model. Then, the tasks of creating, solving, and extracting the solutions of the sub-problems generated by the PH algorithm are executed. Parallel processing was implemented in the tasks of creating, solving, and extracting the solutions. The solutions of each optimization sub-problem are retrieved and average values and simplex multipliers needed in each iteration of the PH algorithm are calculated. Finally, if the algorithm has converged the program writes the solution files. Otherwise, the program goes back, updates the multipliers, and performs another iteration of the PH algorithm.

5. Test system and sub-problem size

The algorithm is tested in the Chilean Central Interconnected System (Sistema Interconectado Central, SIC), Chile's largest interconnected system serving nearly 92% of the country's population. It has an installed generation capacity of about 14.1 GW (2014), of which approximately 42.3% is hydro (reservoir and run-of-river, some in cascading hydro schemes) and 55.6% is thermal (coal-fired, combined-cycle and open-cycle gas-fired, and some fuel-oil and diesel-based peaking plants), with the remaining capacity being wind and solar.

Our model of the SIC is composed of 152 buses and 202 transmission lines, 330 generators, of which 205 are thermal and 11 hydro with significant water storage. The remaining generators are wind farms, run-of-river, and cascading hydro units.

An instance of the problem for the SIC considering a weekly horizon and using hourly resolution is solved for different numbers of stochastic scenarios. If S is the number of scenarios, the PH algorithm decomposes the full problem into S sub-problems, each with 33116 rows (constraints) and 427010 columns (variables). As there are integer variables, the optimization is formulated as a mixed-integer program (MIP) that can be solved using the Branch & Bound algorithm of CPLEX. After the *Presolve* and *Aggregator* routines of CPLEX, each sub-problem has 24691 rows, 367786 columns, with 962716 non-zero coefficients.

6. Computational simulation results

Our results show that in general the PH algorithm has nice convergence properties for the stochastic STHTGS problem. We have identified three main issues affecting the speed of convergence of PH applied to the stochastic STHTGS problem: (1) Diversity of the scenarios, as scenarios that are too similar will lead to solutions of first-stage variables that are too similar in the first iteration; (2) Amount of water available during the scheduling period, as wet conditions do not require switching intermediate and peaking units so much as dry conditions; (3) The size of the penalization factors for not complying with the non-anticipativity constraints in the PH algorithm.

In the case presented here, the PH algorithm only needed two iterations to converge, due to the wet conditions of the simulated case and partly due to the lack of diversity on the defined scenarios. Next, we will present computational results both for the shared memory and the distributed memory architectures.

6.1. Results with a shared memory architecture

Computational simulation results in terms of solution time are presented in Figures 1, 2, and 3 for a shared memory architecture. These results only use a single node of the cluster, hence having 12 cores available.

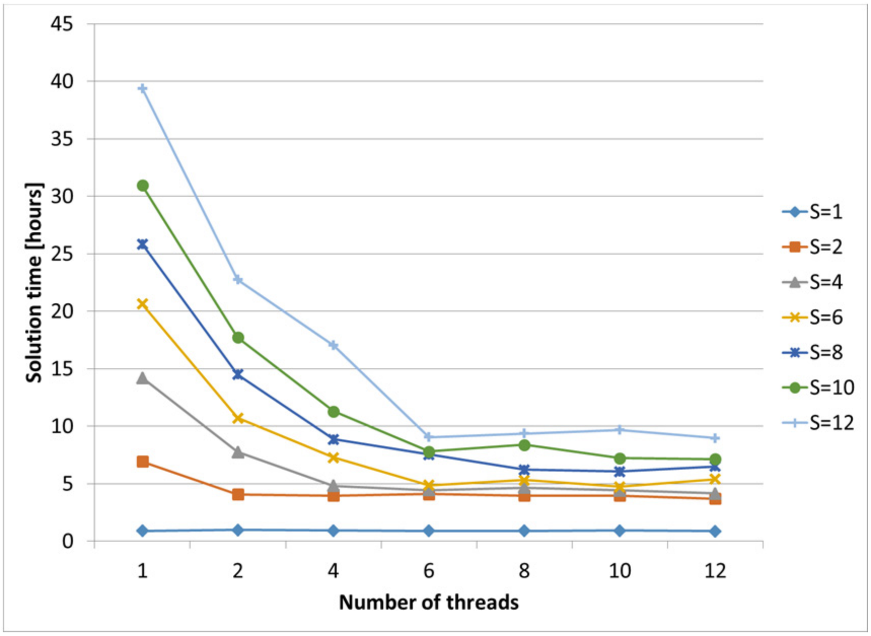

Fig. 1. Solution time against number of threads in use, for different numbers of stochastic scenarios *S*.

Figure 1 shows how the solution time decreases as we use a larger number of threads, considering different numbers of stochastic scenarios. For a single scenario (S=1) we observe no benefit in using more than a single thread, as there is nothing to parallelize. In general, using more threads reduces simulation time, but there is little or no benefit of using more threads than the number of scenarios modeled.

Beyond six scenarios, we observe that the incremental benefits of using a greater number of threads decrease. Although the node has 12 cores available, each thread of CPLEX uses 2 cores, so if we employ more than 6 scenarios we will have to wait for the first 6 scenarios to finish before we can solve the rest, severely hindering the performance of the parallelization.

Figure 2 shows the same results from a different perspective. The simulation time seems to increase nearly linearly with the number of scenarios. Although the benefits of the parallelization scheme in terms of solution time reduction are clear, we observe that there is little benefit of using more than 6 threads, even as the numbers of scenarios keeps increasing.

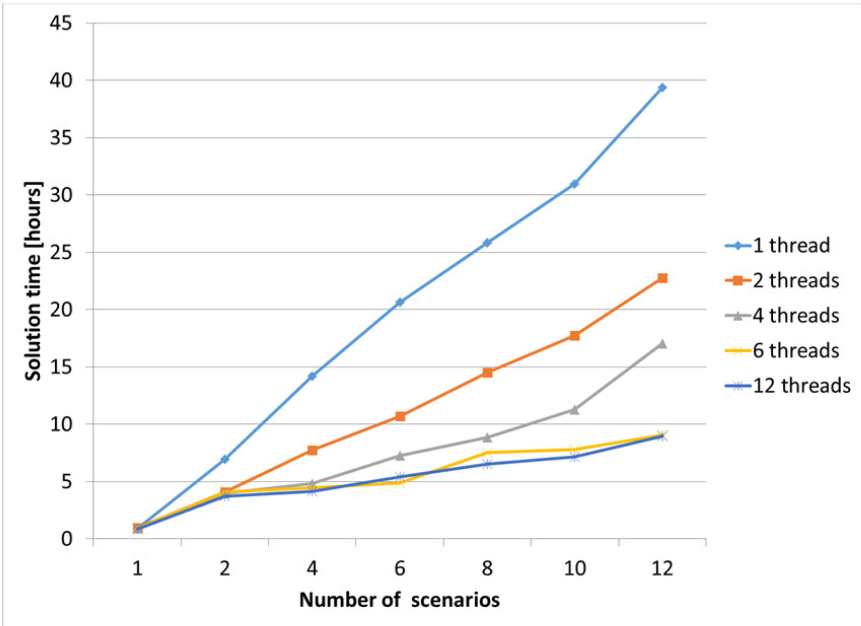

Fig. 2. Solution time against number of stochastic scenarios, for different numbers of threads in use.

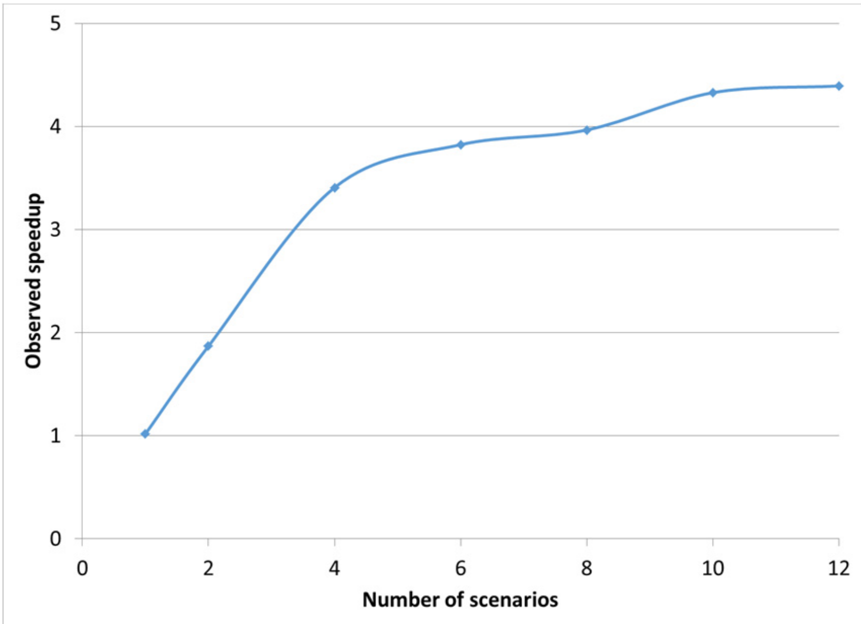

Fig. 3. Observed speedup using twelve threads for different number of stochastic scenarios.

Figure 3 measures up the performance of the parallel program showing the observed speedup as the number of scenarios increase. The observed speedup is calculated as the ratio between the solution time for the serial execution and the parallel execution. With a single scenario, there is almost no benefit for using the twelve threads. Furthermore, as PH generates similarly sized sub-problems, the parallel version of the algorithm scales up quite well as the number of scenarios is increased. Again, beyond six scenarios, the improvements in observed speedup are more modest.

6.2. Results with a distributed memory architecture

For up to six scenarios, running in a single node (i.e. shared memory architecture) or in both nodes of the cluster (i.e. distributed memory architecture) makes little difference, as in both cases there are sufficient cores available to run all the threads simultaneously. For more than 6 scenarios, however, the distributed memory scheme allowed to run all threads simultaneously as more cores were available (24 instead of 12), thus allowing to run 12 instead of 6 threads at the same time. As a consequence, as the number of scenarios and the number of parallel tasks increases, the distributed memory architecture showed better performance in terms of simulation time than the shared memory architecture.

When running with 8 scenarios, for example, the distributed memory architecture was 1.47 times faster than the shared memory architecture. For 10 scenarios, it was 1.53 times faster, and for 12 scenarios it was 1.78 times faster. Notice, however, that despite having twice as many cores available, the distributed memory architecture is not twice as fast as the shared memory architecture. This is a consequence of parallel overhead, that is, the amount of time required to coordinate the parallel tasks. There is also some overhead caused by the time necessary for the remote processors to communicate through the network and pass their results to the master problem, which needs to calculate the average of the first-stage decision variables in order to update the multipliers and the penalties on the objective function of the PH algorithm before starting the next iteration.

7. Conclusions

This paper proposed a SMILP formulation of the STHTGS problem that was decomposed through the PH algorithm. The paper also proposed a strategy for parallelizing the solution both for shared memory and distributed memory schemes. Results showed that PH converges relatively fast for the stochastic STHTGS problem. Insights on computer simulation time were provided, and variations on the number of threads and scenarios were evaluated. Since PH decomposes the main problem by scenario (thus generating similarly sized sub-problems) our results show that parallelism and potentially the use of HPC are very good alternatives to reduce simulation times for the STHTGS problem.

This work was intended as a pilot study to explore the advantages and potential for using HPC in the stochastic STHTGS for the SIC. Despite the relatively modest computational resources used for this article, they were sufficient to explore the use of PH, to identify challenges and advantages of the different architectures, and to specify the size and characteristics of the cluster for running the problem in a more realistic setting.

A challenge for using PH in STHTGS is the existence of integer variables, as the sub-problems may be more difficult to parallelize because of the variability of sub-problem solve times. A possible line for future research is to explore further the tradeoff between number of scenarios and simulation times, as recent reports for the unit commitment problem have pointed out that the benefits of increasing the number of scenarios in the SMILP formulation beyond a certain point may be rather limited.

Convergence in the case study shown in this paper was relatively fast. Additional tests have shown us that the method may take more iterations to converge if the scenarios are more diverse and if the amount of water in the system is more limited, as more intermediate and peaking thermal units may be needed to operate in this case, causing more differences between the solutions to each sub-problem. Thus, another line for further research would be to study the convergence properties of the PH algorithm when the scenarios are more diverse and to make comparisons for both wet and dry conditions.

Acknowledgements

The authors acknowledge the support of the Chilean National Commission for Scientific and Technological Research (CONICYT) under grants Fondecyt 1151270 and Basal Project FB0008, Advanced Center for Electrical and Electronic Engineering (AC3E).

References

[1] Carrión M, Arroyo J. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Transactions on Power Systems 2006; 21:1371–1378.
[2] Gil E, Bustos J, Rudnick H. Short-term hydrothermal generation scheduling model using a genetic algorithm. IEEE Transactions on Power Systems 2003; 18:1256–1264.
[3] Farhat IA, El-Hawary ME. Optimization methods applied for solving the short-term hydrothermal coordination problem. Electric Power Systems Research 2009; 79:1308–1320.
[4] Birge J, Louveaux F. Introduction to Stochastic Programming, 2nd ed. New York: SpringerVerlag; 2011.
[5] Ruszczynski A, Shapiro A. Stochastic programming models. In: Ruszczynski A, Shapiro A., editors. Stochastic Programming, Vol. 10 of Handbooks in Operations Research and Management Science. Elsevier; 2003. p. 1-64.
[6] Ruszczynski A. Decomposition methods. In: Ruszczynski A, Shapiro A., editors. Stochastic Programming, Vol. 10 of Handbooks in Operations Research and Management Science. Elsevier; 2003. p. 141–211.
[7] Sagastizabal C. Divide to conquer: Decomposition methods for energy optimization. Mathematical Programming, Series B 2012; 134:187–222.
[8] Rockafellar R, Wets. R. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of operations research 1991; 16:119–147.
[9] Ryan S, Wets R, Woodruff D, Silva-Monroy C, Watson JP. Toward scalable, parallel progressive hedging for stochastic unit commitment. In: 2013 IEEE Power and Energy Society General Meeting. p. 1–5.
[10] Cheung K, Feng Y, Gade D, Lee Y, Silva-Monroy C, Rios I, Ryan S, Rudel F, Watson JP, Wets R, Woodruff D. Stochastic unit commitment at ISO scale: An ARPAe project. In: 2014 IEEE Power and Energy Society General Meeting. p. 1–4.
[11] Takriti S, Birge J, Long E. A stochastic model for the unit commitment problem. IEEE Transactions on Power Systems 1996; 11:1497–1508.
[12] Goncalves RE, Finardi EC, da Silva EL. Applying different decomposition schemes using the progressive hedging algorithm to the operation planning problem of a hydrothermal system. Electric Power Systems Research 2012; 83:19–27.
[13] Chivers I, Sleightholme J. Introduction to programming with Fortran, 2nd edition. Springer; 2012.
[14] Chapman SJ. Fortran 95/2003 for scientists and engineers, 3rd Edition. McGraw-Hill; 2007.
[15] Loh E. The ideal hpc programming language. Communications of the ACM 2010; 53: 42–47.
[16] IBM. IBM ILOG CPLEX v12.1: User's manual for cplex, available from www.ibm.com; 2009.
[17] OpenMP Architecture Review Board. OpenMP application program interface version 4.0, available from www.openmp.org.
[18] MPI Forum. MPI: A message-passing interface standard. version 3.0, available from www.mpi-forum.org.