A Distributed Computing Architecture for the Large-Scale Integration of Renewable Energy and Distributed Resources in Smart Grids

Ignacio Aravena[1]*, Anthony Papavasiliou[1] and Alex Papalexopoulos[2]

[1]CORE, Université catholique de Louvain. Voie du Roman Pays 34, B-1348 Louvain-la-Neuve, Belgium.
Email: ignacio.aravena@uclouvain.be, anthony.papavasiliou}@uclouvain.be.
[2]ECCO International. 268 Bush Street, Suite 3633 San Francisco, CA 94104, USA. Email: alexp@eccointl.com.
*Corresponding author.

Abstract: We present a distributed computing architecture for smart grid management, composed of two applications at two different levels of the grid. At the high voltage level, we optimize operations using a stochastic unit commitment (SUC) model with hybrid time resolution. The SUC problem is solved with an asynchronous distributed subgradient method, for which we propose stepsize scaling and fast initialization techniques. The asynchronous algorithm is implemented in a high performance computing cluster and benchmarked against a deterministic unit commitment model with exogenous reserve targets in an industrial-scale test case of the Central Western European system (679 buses, 1.037 lines and 656 generators). At the distribution network level, we manage demand response from small clients through distributed stochastic control, which enables harnessing residential demand response while respecting the desire of consumers for control, privacy and simplicity. The distributed stochastic control scheme is successfully tested on a test case with 10.000 controllable devices. Both applications demonstrate the potential for efficiently managing flexible resources in smart grids and for systematically coping with the uncertainty and variability introduced by renewable energy.

Keywords: Smart grids, stochastic programming, asynchronous distributed algorithm, stochastic control, demand response.

1. Introduction

The progressive integration of renewable energy resources, demand response, energy storage, electric vehicles and other distributed resources in electric power grids that has been taking place worldwide in recent years is transforming power systems and resulting in numerous operational challenges, including uncertainty of supply availability, distributed storage management, real-time coordination of distributed energy resources, and changing directions of flow in distribution networks. These challenges demand a shift of the traditional centralized power system operations paradigm towards the smart grid paradigm [1], where distributed computing and control stand out as a promising technology with the potential of achieving operations with optimal performance.

The academic literature includes various applications of distributed computing in power system operations, including long- and mid-term planning, short-term scheduling, state estimation and monitoring, real-time control and simulation [2, 3, 4, 5]. Early studies pointed out several challenges related to communications and the heterogeneous characteristics of distributed computing systems, which needed to be addressed first in order to implement distributed computing applications. Nowadays, standard communication protocols are a mature technology and most current distributed computing resources can perform a broad range of operations. Such advances in distributed computing technology have paved the way for developing and implementing scalable distributed algorithms for power systems.

The prevailing industry view, as we move forward into the future smart grid, is that it will entail (i) broadcasting of dynamic prices or other information and (ii) telemetry backhaul to market participants. In the proposed model, distributed energy resource aggregators are often regarded as transaction brokers between end customers and various upstream market participants. The "failure-free market" design for a pure market-driven solution under this paradigm has been elusive, despite decades of research and development. In this chapter, we analyse the deployment of distributed computing as an enabling tool for managing the short-term operations of smart grids in two levels:

- At the level of the high-voltage grid, we centrally optimize operations using a stochastic unit commitment (SUC) model, which endogenously allocates reserve capacity by explicitly modeling uncertainty. Specifically, we present an asynchronous distributed algorithm for solving SUC, which extends the asynchronous algorithm proposed in [6] in three aspects: (i) we propose a hybrid approach for modeling quarterly dispatch decisions alongside hourly commitment decisions, (ii) we introduce a stepsize scaling on the iterative method to diminish the error due to asynchronous execution, and (iii) we propose two methods for a faster initialization of the algorithm. The asynchronous algorithm is implemented in a high performance computing (HPC) cluster and benchmarked against a deterministic unit commitment model with exogenous reserve targets (DUCR). We find that distributed computing allows solving SUC within the same time frame required for solving DUCR.
- At the level of the distribution grid, we rely on stochastic distributed control to manage consumer devices using the ColorPower architecture [7, 8, 9], which enables harnessing flexible residential demand response while respecting the desire of consumers for control, privacy and simplicity. The ColorPower control approach is inspired by the very automatic cooperative protocols that govern Internet communications. These protocols represent a distributed and federated control paradigm, in which information and decision-making authority remain local, yet global system stability is ensured.

Centralized clearing at the high-voltage grid level and distributed clearing at the distribution grid level can be integrated in a co-optimization framework, as recently proposed by Caramanis et al. [10]. These two applications of distributed computing in power system operations demonstrate the potential to fully harness the flexibility of the grid and smoothly integrate large shares of renewable and other distributed energy resources in power systems without deteriorating the quality of service delivered to consumers.

The rest of the chapter is organized as follows. Section 2 introduces the deterministic and stochastic unit commitment problems. Section 3 proposes an asynchronous algorithm for solving SUC and presents numerical experiments on a network of realistic scale. Section 4 presents the ColorPower architecture for managing demand response in the distribution grid and demonstrates its capability through a numerical experiment. Finally, section 5 concludes the chapter.

2. High-voltage power grid optimization models

Nomenclature

Sets

T_{60}	hourly periods, $T_{60} \coloneqq \{1, \dots, \lvert T_{60} \rvert\}$
T_{15}	15-minute periods, $T_{15} \coloneqq \{1, \dots, \lvert T_{15} \rvert\}$
S	scenarios, $S \coloneqq \{s_1, \dots, s_M\}$
A	reserve areas
N	buses
L	lines
G	thermal generators
$N(a)$	buses in area a
$L(n,m)$	lines between buses n and m, directed from n to m
$G(n)$	thermal generators at bus or bus set n
G_{SLOW}	slow generators, $G_{SLOW} \subseteq G$

Parameters

$\tau(t)$	corresponding hour of quarter t
π_s	probability of scenario s
$D_{n,t}$	demand at bus n in period t
$\bar{\xi}_{n,t}, \xi_{n,s,t}$	forecast renewable supply, bus n, scenario s, quarter t
$\mathcal{R}^2_a, \mathcal{R}^3_a$	secondary and tertiary reserve requirements in area a
$\Delta T_2, \Delta T_3$	delivery time of secondary and tertiary reserves, $0 < \Delta T_2 < \Delta T_3 \leq 15$
$F_l^{\pm}$	flow bounds, line l
B_l	susceptance, line l

$n(l), m(l)$ departing and arrival buses, line l
$P_g^{\pm}$ minimum stable level and maximum run capacity, generator g
$R_g^{\pm}$ maximum 15-minute ramp down/up, generator g
TL_g maximum state transition level, generator g
UT_g, DT_g minimum up/down times, generator g
K_g hourly no-load cost, generator g
S_g startup cost, generator g
$C_g(p)$ quarterly production cost function, generator g (convex, piece-wise linear)

Variables

$p_{g,t}, p_{g,s,t}$ production, generator g, scenario s, quarter t
$f_{l,t}, f_{l,s,t}$ flow through line l, scenario s, quarter t
$\theta_{n,t}, \theta_{n,s,t}$ voltage angle, bus n, scenario s, quarter t
r_g^2, r_g^3 capacity and ramp up rate reservation for secondary and tertiary reserve provision, generator g, quarter t
$u_{g,\tau}, u_{g,s,\tau}$ commitment, generator g, scenario s, hour τ
$v_{g,\tau}, v_{g,s,\tau}$ startup, generator g, scenario s, hour τ
$w_{g,\tau}, z_{g,\tau}$ non-anticipative commitment and startup, generator g, hour τ
$\mu_{g,s,\tau}, \nu_{g,s,\tau}$ dual multipliers of non-anticipativity constraints, generator g, scenario s, hour τ

2.1. Overview

Operations of the high-voltage power grid are typically scheduled in two stages: (i) day-ahead scheduling, where operations are planned based on forecast conditions for the system and the on/off status of slow generators is fixed, and (ii) real-time scheduling, where system operators balance the system for the actual conditions using the available flexibility in the system. Models for short-term scheduling are solved on a daily basis, and they occupy a central role in clearing power markets and operating power systems.

Until recently, power system operators have relied on deterministic short-term scheduling models with reserve margins to secure the system against load forecast errors and outages [11, 12, 13, 14]. The integration of renewable energy sources has placed these practices under question because they ignore the inherent uncertainty of renewable energy supply, thereby motivating system operators and researchers to look for systematic methods to address uncertainty in real-time operations. A consistent methodology for mitigating the impacts of renewable energy uncertainty – and operational uncertainty in general – is stochastic programming. Stochastic models for short-term scheduling (i.e. SUC models) were originally considered in the seminal work of Takriti *et al.* [15] and Carpentier *et al.* [16], as an approach for mitigating demand uncertainty and generator outages. Subsequently, numerous variants of the SUC model have been proposed, which differ on the number of stages, the source of uncertainty, the representation of uncertainty and the solution methods that are used. See [17] and references therein for a recent survey.

In the present work we use the deterministic and stochastic unit commitment models for day-ahead scheduling presented in sections 3.1 and 3.2. The proposed models differ from previously proposed models in the literature in that they use hybrid time resolution: hourly commitment decisions (u, v, w and z) and 15-minute dispatch decisions (p, r and f). This formulation allows modeling sub-hourly phenomena, which have been shown to be important for the operation of systems with significant levels of renewable energy integration [18].

2.2. Deterministic unit commitment with exogenous reserve targets

Using the notation provided in the beginning of the section, we model deterministic unit commitment with reserves (DUCR) as the minimization problem (1)-(9).

$$\min_{p,r,u,v,f} \quad \sum_{g\in G}\left(\sum_{\tau\in T_{60}}\left(K_g u_{g,\tau} + S_g v_{g,\tau}\right) + \sum_{t\in T_{15}} C_g\left(p_{g,t}\right)\right) \tag{1}$$

$$\text{s.t.} \quad \sum_{g\in G(n)} p_{g,t} + \sum_{l\in L(\cdot,n)} f_{l,t} + \bar{\xi}_{n,t} \geq D_{n,t} + \sum_{l\in L(n,\cdot)} f_{l,t} \quad \forall n \in N, t \in T_{15} \tag{2}$$

$$\sum_{g\in G(a)} r_{g,t}^2 \geq \mathcal{R}_a^2, \sum_{g\in G(N(a))}\left(r_{g,t}^2 + r_{g,t}^3\right) \geq \mathcal{R}_a^2 + \mathcal{R}_a^3 \quad \forall a \in A, t \in T_{15} \tag{3}$$

$$f_{l,t} = B_l\left(\theta_{n(l),t} - \theta_{m(l),t}\right), -F_l^- \leq f_{l,t} \leq F_l^+ \quad \forall l \in L, t \in T_{15} \tag{4}$$

$$P_g^- u_{g,\tau(t)} \leq p_{g,t}, p_{g,t} + r_{g,t}^2 + r_{g,t}^3 \leq P_g^+ u_{g,\tau(t)} \quad \forall g \in G_{SLOW}, t \in T_{15} \tag{5}$$

$$P_g^- u_{g,\tau(t)} \le p_{g,t}, p_{g,t} + r_{g,t}^2 \le P_g^+ u_{g,\tau(t)}, p_{g,t} + r_{g,t}^2 + r_{g,t}^3 \le P_g^+ \ \forall g \in G \backslash G_{SLOW}, t \in T_{15} \quad (6)$$

$$-TL_g + \left(TL_g - R_g^-\right) u_{g,\tau(t)} \le p_{g,t} - p_{g,t-1} \ \forall g \in G, t \in T_{15} \quad (7)$$

$$p_{g,t} + \frac{15}{\Delta T_2} r_{g,t}^2 - p_{g,t-1} \le TL_g - \left(TL_g - R_g^+\right) u_{g,\tau(t-1)},$$

$$p_{g,t} + \frac{15}{\Delta T_3}\left(r_{g,t}^2 + r_{g,t}^3\right) - p_{g,t-1} \le TL_g - \left(TL_g - R_g^+\right) u_{g,\tau(t-1)} \ \forall g \in G, t \in T_{15} \quad (8)$$

$$u_{g,\tau} \in \{0,1\}, v_{g,\tau} \in \{0,1\} \ \forall g \in G, \tau \in T_{60} \quad (9)$$

The objective function (1) corresponds to the total operating cost, composed by the no-load cost, the startup cost and the production cost. Constraints (2) enforce nodal power balance, while allowing for production shedding. Demand shedding can be included in the present formulation as having a very expensive generator connected to each bus. Equation (3) enforces the reserve margins on each area of the system, allowing for reserve cascading (secondary reserve capacity can be used to provide tertiary reserve). Equation (4) models DC power flow constraints in terms of bus angles and thermal limits of transmission lines.

The feasible production set of thermal generators is described by Eqs. (5)-(9). Production and reserve provision limits are expressed as (5) for slow generators, that can provide reserves only when they are online, and as (6) for the remaining set of generators, which can provide secondary reserves when they are online and tertiary reserves both when they are online and offline. Ramp rate constraints (7)-(8) are based on the formulation provided by Frangioni *et al.* [19]. Ramp-up rate constraints (8) enforce, in addition to the ramp-up rate limit on production, that there is enough ramping capability between periods $t-1$ and t to ramp up $r_{g,t}^2$ MW within ΔT_2 minutes (which can be used to provide secondary reserve), and to ramp up $r_{g,t}^2 + r_{g,t}^3$ MW within ΔT_3 minutes (which can be used to provide tertiary reserve). Constraints (9) enforce minimum up and down times, as proposed by Rajan and Takriti [20].

Boundary conditions of the problem are modeled by allowing the time indices to cycle within the horizon, in other words, for any commitment variable $x_{\cdot,\tau}$ with $\tau < 1$, we define $x_{\cdot,\tau} \coloneqq x_{\cdot,((\tau-1) \bmod |T_{60}|+1)}$. Similarly, for any dispatch variable $x_{\cdot,t}$ with $t < 1$ or $t > |T_{15}|$, we define $x_{\cdot,t} \coloneqq x_{\cdot,((t-1) \bmod |T_{15}|+1)}$. In this fashion, we model initial conditions ($\tau < 1$, $t < 1$) and restrain end effects of the model ($\tau = |T_{60}|$, $t = |T_{15}|$) simultaneously. In practical cases, initial conditions are given by the current operating conditions and end effects are dealt with by using an extended look-ahead horizon.

2.3. Two-stage stochastic unit commitment and scenario decomposition

Following Papavasiliou et al. [21], we formulate SUC as the two-stage stochastic program of Eqs. (10)-(17).

$$\min_{\substack{p,u,v,f \\ w,z}} \ \sum_{s\in S} \pi_s \sum_{g\in G} \left(\sum_{\tau \in T_{60}} \left(K_g u_{g,s,\tau} + S_g v_{g,s,\tau}\right) + \sum_{t \in T_{15}} C_g\left(p_{g,s,t}\right) \right) \quad (10)$$

$$\text{s.t.} \ \sum_{g\in G(n)} p_{g,s,t} + \sum_{l \in L(\cdot,n)} f_{l,s,t} + \xi_{n,s,t} \ge D_{n,t} + \sum_{l \in L(n,\cdot)} f_{l,s,t} \ \forall n \in N, s \in S, t \in T_{15} \quad (11)$$

$$f_{l,s,t} = B_l\left(\theta_{n(l),s,t} - \theta_{m(l),s,t}\right), -F_l^- \le f_{l,s,t} \le F_l^+ \ \forall l \in L, s \in S, t \in T_{15} \quad (12)$$

$$P_g^- u_{g,s,\tau(t)} \le p_{g,s,t} \le P_g^+ u_{g,s,\tau(t)} \ \forall g \in G, s \in S, t \in T_{15} \quad (13)$$

$$-TL_g + \left(TL_g - R_g^-\right) u_{g,\tau(t)} \le p_{g,s,t} - p_{g,s,t-1} \le$$
$$TL_{\mathrm{g}} - \left(TL_g - R_g^+\right) u_{g,\tau(t-1)} \ \forall g \in G, s \in S, t \in T_{15} \quad (14)$$

$$v_{g,s,\tau} \ge u_{g,s,\tau} - u_{g,s,\tau}, \sum_{\sigma=\tau-UT_g+1}^{\tau} v_{g,s,\sigma} \le u_{g,s,\tau}, \sum_{\sigma=\tau-DT_g+1}^{\tau} v_{g,s,\sigma} \le 1 - u_{g,s,\tau-DT_g},$$

$$u_{g,s,\tau} \in \{0,1\}, v_{g,s,\tau} \in \{0,1\} \ \forall g \in G, s \in S, \tau \in T_{60} \quad (15)$$

$$\pi_s u_{g,s,\tau} = \pi_s w_{g,\tau} \to \mu_{g,s,\tau}, \pi_s v_{g,s,\tau} = \pi_s z_{g,\tau} \to \nu_{g,s,\tau} \ \forall g \in G_{SLOW}, s \in S, \tau \in T_{60} \quad (16)$$

$$z_{g,\tau} \ge w_{g,\tau} - w_{g,\tau}, \sum_{\sigma=\tau-UT_g+1}^{\tau} z_{g,\sigma} \le w_{g,\tau}, \sum_{\sigma=\tau-DT_g+1}^{\tau} z_{g,\sigma} \le 1 - w_{g,\tau-DT_g}$$

$$w_{g,\tau} \in \{0,1\}, z_{g,\tau} \in \{0,1\} \ \forall g \in G, \tau \in T_{60} \quad (17)$$

The objective function (10) corresponds to the expected cost over the set of scenarios S, with associated probabilities π_s. Constraints (11)-(12) are analogous to (2) and (4). No explicit reserve requirements are enforced in the stochastic

unit commitment model, since reserves are endogenously determined by the explicit modeling of uncertainty. Consequently, generator constraints of the deterministic problem, Eqs. (5)-(10), become identical for all thermal generators and can be expressed as Eqs. (13)-(15). Non-anticipativity constraints (16) are formulated using state variables $\boldsymbol{w}$ and $\boldsymbol{z}$ for the commitment and startup decisions of slow thermal generators (first-stage decisions). We associate Lagrange multipliers μ and ν with non-anticipativity constraints. Constraints (17) enforce minimum up and down times on unit commitment variables.

3. An asynchronous distributed algorithm for stochastic unit commitment

Nomenclature

Sets

Q^D	dual queue (ordered set) of scenarios
Q^P	primal queue of pairs: ‹candidate solution, scenario›

Parameters

DP, PP	number of dual and primal processors
α_k	stepsize, asynchronous subgradient method
β_s	stepsize scaling factor, scenario s

Variables

LB, UB	lower and upper bound on objective of stochastic unit commitment
UB_s^l	upper bound of primal candidate l on scenario s

3.1. Scenario decomposition of the SUC problem

The SUC problem (10)-(17) grows linearly in size with the number of scenarios. Hence, SUC problems are in general of large scale, even for small system models. This motivated Takriti *et al.* [15] and Carpentier *et al.* [16] to rely on Lagrangian decomposition methods for solving the problem.

Recent SUC studies have focused on designing decomposition algorithms capable of solving the problem in operationally acceptable time frames. Papavasiliou *et al.* [21] propose a dual scenario decomposition scheme where the dual is solved using the subgradient method, and where the dual function is evaluated in parallel. Kim and Zavala [22] also use a dual scenario decomposition scheme, but solve the dual problem using a bundle method. Cheung *et al.* [23] present a parallel implementation of the progressive hedging algorithm of Rockafellar and Wets [24].

All previously mentioned parallel algorithms for SUC are synchronous algorithms, i.e. scenario subproblems are solved in parallel at each iteration of the decomposition method, however, it is necessary to solve all scenario subproblems before advancing to the next iteration. In cases where the solution times of subproblems differ significantly, synchronous algorithms lead to an underutilization of the parallel computing infrastructure and a loss of parallel efficiency. We have found instances where the time required to evaluate subproblems for difficult scenarios is 75 times longer than the solution time for easy scenarios.

Aiming at overcoming the difficulties faced by synchronous algorithms, we propose an asynchronous distributed algorithm for solving SUC. The algorithm is based on the scenario decomposition scheme for SUC proposed in [21], where the authors relax the non-anticipativity constraints (16) and form the following Lagrangian dual problem

$$\max_{\mu,\nu} h_0(\boldsymbol{\mu},\boldsymbol{\nu}) + \sum_{s\in S} h_s(\boldsymbol{\mu}_s,\boldsymbol{\nu}_s)\,, \tag{18}$$

where $h_0(\boldsymbol{\mu},\boldsymbol{\nu})$ and $h_s(\boldsymbol{\mu}_s,\boldsymbol{\nu}_s)$ are defined according to (19) and (20), respectively. We use boldface to denote vectors and partial indexation of dual variables with respect to scenarios, so that $\boldsymbol{\mu}_s \coloneqq \left[\mu_{g_1,s,1} \dots \mu_{g_{|G|},s,1}\right]^T$. The constraints within the infimum in (20) refer to constraints (11)-(15) for scenario s (dropping the scenario indexation of variables).

$$h_0(\boldsymbol{\mu},\boldsymbol{\nu}) \coloneqq \inf_{w,z}\left\{ \sum_{g\in G_{SLOW}} \sum_{\tau\in T_{60}} \left(-\left(\sum_{s\in S}\left(\pi_s \mu_{g,s,\tau}\right)\right) w_{g,\tau} - \left(\sum_{s\in S}\left(\pi_s \nu_{g,s,\tau}\right)\right) z_{g,\tau} \right) : (17) \right\} \tag{19}$$

$$h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s) \coloneqq \pi_s \inf_{p,u,v,f} \left\{ \begin{array}{c} \sum_{g\in G}\sum_{t\in T_{15}} C_g(p_{g,t}) + \sum_{g\in G\setminus G_{SLOW}}\sum_{\tau\in T_{60}} \left(K_g u_{g,\tau} + S_g v_{g,\tau}\right) + \\ \sum_{g\in G_{SLOW}}\sum_{\tau\in T_{60}} \left(\left(K_g + \mu_{g,s,\tau}\right)u_{g,\tau} + \left(S_g + \nu_{g,s,\tau}\right)v_{g,\tau}\right) : (11s) - (15s) \end{array} \right\} \tag{20}$$

Both $h_0(\boldsymbol{\mu}, \boldsymbol{\nu})$ and $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ for all $s \in S$ are non-differentiable convex functions. Evaluating $h_0(\boldsymbol{\mu}, \boldsymbol{\nu})$ amounts to solving a small integer programming problem, for the constraints of which we have a linear-size convex hull description [20]. Evaluating $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ amounts to solving a deterministic unit commitment problem without reserve requirements (DUC), which is a mixed-integer linear program of potentially large scale for realistic system models. In practice, the run time for evaluating $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ for any s and any dual multipliers is at least two orders of magnitude greater than the run time for evaluating $h_0(\boldsymbol{\mu}, \boldsymbol{\nu})$.

The proposed distributed algorithm exploits the characteristics of $h_0(\boldsymbol{\mu}, \boldsymbol{\nu})$ and $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ in order to maximize (18) and compute lower bounds on the optimal SUC solution, while recovering feasible non-anticipative commitment schedules with associated expected costs (upper bounds to the optimal SUC solution). The dual maximization algorithm is inspired by the work of Nedić *et al.* on asynchronous incremental subgradient methods [25].

3.2. Dual maximization and primal recovery

For simplicity, assume that we have $1 + DP + PP$ available parallel processors which can all access a shared memory space. We allocate one processor to coordinate the parallel execution and manage the shared memory space, $DP \leq |S|$ processors to solve the dual problem (18) and PP processors to recover complete solutions to the SUC problem (10)-(17). Interactions between different processors are presented in Figure 1.

We maximize the dual function (18) using a block-coordinate descent (BCD) method, in which each update is performed over a block of dual variables associated with a scenario, $(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ for certain $s \in S$, following the direction of the subgradient of the dual function in the block of variables $(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$. The BCD method is implemented in parallel and asynchronously by having each dual processor perform updates on the dual variables associated with a certain scenario, which are not being updated by any other dual processor at the same time. Scenarios whose dual variables are not currently being updated by any processor are held in the dual queue $\mathcal{Q}^D$, to be updated later.

We maintain shared memory registers of $\mathcal{Q}^D$. We denote the current multipliers as $\left(\boldsymbol{\mu}_s^{k(s)}, \boldsymbol{\nu}_s^{k(s)}\right)$ $\forall s \in S$, where $k(s)$ is the number of updates to the block of scenario s; the previous-to-current dual multipliers as $\left(\boldsymbol{\mu}_s^{k(s)-1}, \boldsymbol{\nu}_s^{k(s)-1}\right)$ and their associated lower bound on $h_s\left(\boldsymbol{\mu}_s^{k(s)-1}, \boldsymbol{\nu}_s^{k(s)-1}\right)$ as $\check{h}_s^{k(s)-1}$, $\forall s \in S$; the global update count as k; and the best lower bound found on (10)-(17) as LB. Additionally, a shared memory register of the primal queue $\mathcal{Q}^P$ is required for recovering primal solutions. Then, each dual processor performs the following operations:

1. Read and remove the first scenario s from $\mathcal{Q}^D$.
2. Read $\left(\boldsymbol{\mu}_s^{k(s)}, \boldsymbol{\nu}_s^{k(s)}\right)$ and evaluate $h_s\left(\boldsymbol{\mu}_s^{k(s)}, \boldsymbol{\nu}_s^{k(s)}\right)$ approximately.
3. Read $\left(\boldsymbol{\mu}_\omega^{k(\omega)-1}, \boldsymbol{\nu}_\omega^{k(\omega)-1}\right)$ and $\check{h}_\omega^{k(\omega)-1}$ for all $\omega \in S \setminus \{s\}$.
4. Construct the delayed multiplier vectors,

$$\bar{\boldsymbol{\mu}} \coloneqq \left(\boldsymbol{\mu}_{s_1}^{k(s_1)-1}, \dots, \boldsymbol{\mu}_{\boldsymbol{s}}^{k(s)}, \dots, \boldsymbol{\mu}_{s_M}^{k(s_M)-1}\right)$$
$$\bar{\boldsymbol{\nu}} \coloneqq \left(\boldsymbol{\nu}_{s_1}^{k(s_1)-1}, \dots, \boldsymbol{\nu}_{\boldsymbol{s}}^{k(s)}, \dots, \boldsymbol{\nu}_{s_M}^{k(s_M)-1}\right),$$

   and evaluate $h_0(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}})$ approximately.
5. Read the current global iteration count k and perform a BCD update on the dual multipliers

$$\boldsymbol{\mu}_s^{k(s)+1} \coloneqq \boldsymbol{\mu}_s^{k(s)} + \frac{\alpha_k}{\beta_s} \cdot \pi_s(\boldsymbol{u}_{SLOW}^* - \boldsymbol{w}^*)$$
$$\boldsymbol{\nu}_s^{k(s)+1} \coloneqq \boldsymbol{\nu}_s^{k(s)} + \frac{\alpha_k}{\beta_s} \cdot \pi_s(\boldsymbol{v}_{SLOW}^* - \boldsymbol{z}^*),$$

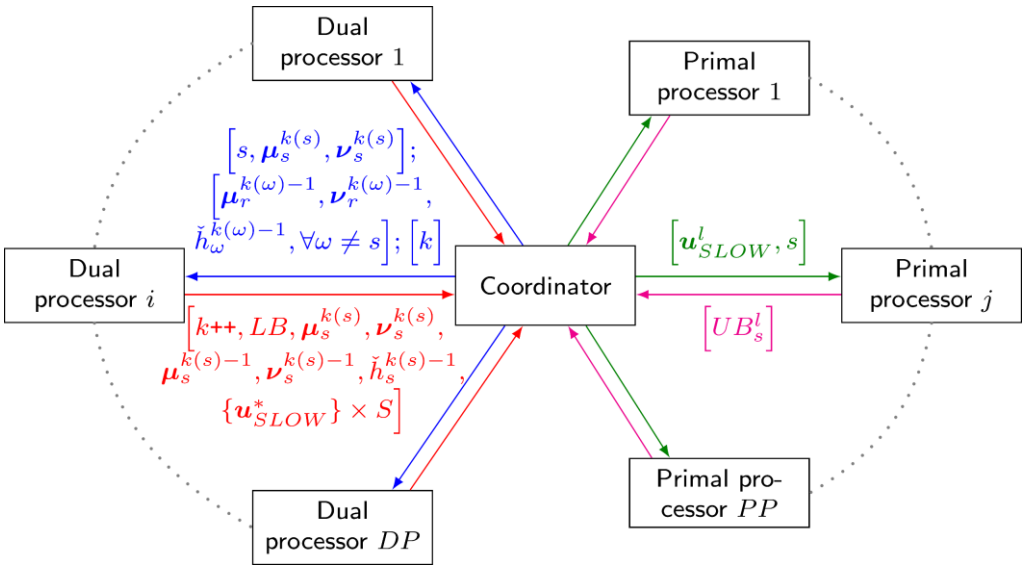

Figure 1. Asynchronous algorithm layout. Information within square brackets is read or written at a single step of the algorithm.

where $(\boldsymbol{w}^*, \boldsymbol{z}^*)$ is an approximate minimizer of (19) for $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}})$, $(\boldsymbol{p}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{f}^*)$ is an approximate minimizer of (20) for $\left(\boldsymbol{\mu}_s^{k(s)}, \boldsymbol{\nu}_s^{k(s)}\right)$ and $(\boldsymbol{u}_{SLOW}^*, \boldsymbol{v}_{SLOW}^*)$ corresponds to the commitment and startup for slow generators in $(\boldsymbol{p}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{f}^*)$.

6. Compute a new lower bound as

$$LB^{\text{new}} := \check{h}_0(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}}) + \check{h}_s\left(\boldsymbol{\mu}_s^{k(s)}, \boldsymbol{\nu}_s^{k(s)}\right) + \sum_{\omega \in S \setminus \{s\}} \check{h}_\omega^{k(\omega)-1},$$

where $\check{h}_0(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}}) \leq h_0(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}})$ and $\check{h}_s\left(\boldsymbol{\mu}_s^{k(s)}, \boldsymbol{\nu}_s^{k(s)}\right) \leq h_s\left(\boldsymbol{\mu}_s^{k(s)}, \boldsymbol{\nu}_s^{k(s)}\right)$ are the lower bounds of the MILP solution of (19) and (20).

7. Let $k(s) := k(s) + 1$ and update in memory:
   a. $k \mathrel{+}= 1$.
   b. $LB := \max\{LB, LB^{\text{new}}\}$.
   c. $\left(\boldsymbol{\mu}_s^{k(s)}, \boldsymbol{\nu}_s^{k(s)}\right)$
   d. $\left(\boldsymbol{\mu}_s^{k(s)-1}, \boldsymbol{\nu}_s^{k(s)-1}\right)$ and $\check{h}_s^{k(s)-1} := \check{h}_s\left(\boldsymbol{\mu}_s^{k(s)-1}, \boldsymbol{\nu}_s^{k(s)-1}\right)$
   e. Add $\{\boldsymbol{u}_{SLOW}^*\} \times S$ to the end of $\mathcal{Q}^P$.
8. Add s at the end of $\mathcal{Q}^D$ and return to 1.

Steps 1 to 3 of the dual processor algorithm are self-explanatory. Step 4 constructs a compound of the previous iterates which is useful for computing lower bounds.

During the execution of the algorithm, step 5 will perform updates to the blocks of dual variables associated to all scenarios. As $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ is easier to evaluate for certain scenarios than others, the blocks of dual variables associated to easier scenarios will be updated more frequently than harder scenarios. We model this process, in a simplified fashion, as if every update is performed on a randomly selected scenario from a non-uniform distribution, where the probability of selecting scenario s corresponds to

$$\beta_s := \frac{T_s^{between}}{\sum_{\omega \in S} T_\omega^{between}},$$

where $T_s^{between}$ is the average time between two updates on scenario s ($T_s^{between}$ is estimated during execution). The asynchronous BCD method can then be understood as a stochastic approximate subgradient method [26], [27]. This is an approximate method for three reasons: (i) as the objective function contains a non-separable non-differentiable function $h_0(\boldsymbol{\mu}, \boldsymbol{\nu})$, there is no guarantee that the expected update direction coincides with a subgradient of the objective of (8) at the current iterate, (ii) $h_0(\boldsymbol{\mu}, \boldsymbol{\nu})$ is evaluated for a delayed version of the multipliers $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\nu}})$ and (iii) $h_0(\boldsymbol{\mu}, \boldsymbol{\nu})$ and $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ are evaluated only approximately up to a certain MILP gap. Provided that we use a diminishing, non-summable and square-summable stepsize α_k of the type $1/k^q$, and that the error in the subgradient is bounded, the method will converge to an approximate solution of the dual problem (8) [26], [27].

In step 6, we compute a lower bound on the primal problem (10)-(17) using previous evaluations of $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ recorded in memory, as proposed in [6]. Step 7 updates the shared memory registers for future iterations and step 8 closes the internal loop of the dual processor.

We recover primal solutions by taking advantage of the fact that $(\boldsymbol{u}^*_{SLOW}, \boldsymbol{v}^*_{SLOW})$ is a feasible solution for $(\boldsymbol{w}, \boldsymbol{z})$ in (10)-(17). Therefore, in order to compute complete primal solutions and obtain upper bounds for problem (10)-(17) we can fix $\boldsymbol{w} \coloneqq \boldsymbol{u}^*_{SLOW}$ and $\boldsymbol{z} \coloneqq \boldsymbol{v}^*_{SLOW}$ and solve the remaining problem, as proposed in [28]. After fixing $(\boldsymbol{w}, \boldsymbol{z})$, the remaining problem becomes separable by scenario hence, in order to solve it, we need to solve a restricted DUC for each scenario in S. These primal evaluation jobs, i.e. solving the restricted DUC for $\{\boldsymbol{u}^*_{SLOW}\}\times S$, are appended at the end of the primal queue Q^P by dual processors after each update (step 7.e). Note that we do not require storing $\boldsymbol{v}^*_{SLOW}$ because its value is implied by $\boldsymbol{u}^*_{SLOW}$.

The primal queue is managed by the coordinator process, which assigns primal jobs to primal processors as they become available. The computation of primal solutions is therefore also asynchronous, in the sense that it runs independently of dual iterations and that the evaluation of candidate solutions $\boldsymbol{u}^*_{SLOW}$ does not require that the previous candidates have already been evaluated for all scenarios. Once a certain candidate $\boldsymbol{u}^l$ has been evaluated for all scenarios, the coordinator can compute a new upper bound to (10)-(17) as

$$UB \coloneqq \min\left\{UB, \sum_{s\in S} UB_s^l\right\},$$

where UB_s^l is the upper bound associated with $\boldsymbol{u}^l$ on the restricted DUC problem of scenario s. The coordinator process keeps track of the candidate associated with the smaller upper bound throughout the execution.

Finally, the coordinator process will terminate the algorithm when $1 - LB/UB \leq \epsilon$, where ϵ is a prescribed tolerance, or when reaching a prescribed maximum solution time. At this point, the algorithm retrieves the best-found solution and the bound on the distance of this solution from the optimal objective function value.

3.3. Dual algorithm initialization

The lower bounds computed by the algorithm presented in the previous section depend on previous evaluations of $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ for other scenarios. As the evaluation of $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ can require a substantial amount of time for certain scenarios, the computation of the first lower bound considering non-trivial values of $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ for all scenarios can be delayed significantly with respect to the advance of dual iterations and primal recovery. In other words, it might be the case that the algorithm finds a very good primal solution but it is unable to terminate because it is missing the value of $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ for a single scenario.

In order to prevent these situations and in order to obtain non-trivial bounds faster, in the first pass of the dual processors over all scenarios we can replace $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ with a surrogate $\eta_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ which is easier to compute, such that $\eta_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s) \leq h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ for any $(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$. We propose two alternatives for $\eta_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$:

i. The linear relaxation of the scenario DUC (LP):

$$\eta_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s) \coloneqq \pi_s \inf_{p,u,v,f} \left\{ \begin{array}{l} \displaystyle\sum_{g\in G}\sum_{t\in T_{15}} C_g(p_{g,t}) + \sum_{g\in G\setminus G_{SLOW}}\sum_{\tau\in T_{60}} \left(K_g u_{g,\tau} + S_g v_{g,\tau}\right) + \\ \displaystyle\sum_{g\in G_{SLOW}}\sum_{\tau\in T_{60}} \left(\left(K_g + \mu_{g,s,\tau}\right)u_{g,\tau} + \left(S_g + \nu_{g,s,\tau}\right)v_{g,\tau}\right) : \\ \text{linear relaxation of } (11s) - (15s) \end{array} \right\}$$

ii. An optimal power flow for each period (OPF):

$$\eta_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s) \coloneqq \pi_s \sum_{t \in T_{15}} \inf_{p,u,v,f} \left\{ \begin{array}{c} \sum_{g \in G} C_g(p_g) + \frac{1}{4} \sum_{g \in G \setminus G_{SLOW}} K_g u_g + \\ \frac{1}{4} \sum_{g \in G_{SLOW}} \left(\left(K_g + \mu_{g,s,\tau(t)}\right) u_g + \left(S_g + \nu_{g,s,\tau(t)}\right) v_g \right) : \\ (11st) - (13st), \boldsymbol{u} \in \{0,1\}^{|G|}, \boldsymbol{v} \in \{0,1\}^{|G_{SLOW}|} \end{array} \right\},$$

where $(11st)$ – $(13st)$ correspond to constraints (11) – (13) for scenario s and period t.

The LP approach requires solving a linear program of the same size as the original problem (20), but it has the advantage that it can be obtained as an intermediate result while evaluating $h_s(\boldsymbol{\mu}_s, \boldsymbol{\nu}_s)$ (the LP approach does not add extra computations to the algorithm). The OPF approach, on the other hand, requires solving many small MILP problems, which can be solved faster than the linear relaxation of (20). The OPF approach ignores several constraints and cost components, such as the startup cost of non-slow generators, and it adds extra computations to the algorithm.

3.4. Implementation and numerical experiments

We implement the DUCR model using Mosel and solve it directly using Xpress. We also implement the proposed asynchronous algorithm for SUC (described in the previous subsections) in Mosel, using the module *mmjobs* for handling parallel processes and communications, while solving the subproblems with Xpress [29]. We configure Xpress to solve the root node using the barrier algorithm and we set the termination gap to 1%, for both the DUCR and SUC subproblems, and the maximum solution wall time to 10 hours. Numerical experiments were run on the Sierra cluster hosted at the Lawrence Livermore National Laboratory. Each node of the Sierra cluster is equipped with two Intel XeonEP X5660 processors (12 cores per node) and 24GB of RAM memory. We use 10 nodes for the proposed distributed algorithm, assigning 5 nodes to dual processors, with 6 dual processors per node ($DP = 30$), and 5 nodes to primal recovery, with 12 primal processors per node. The coordinator is implemented on a primal node and occupies one primal processor ($PP = 59$).

We test the proposed algorithm on a detailed model of the Central Western European system, consisting of 656 thermal generators, 679 nodes and 1037 lines. The model was constructed by using the network model of Hutcheon and Bialek [30], technical generator information provided to the authors by ENGIE, and multi-area demand and renewable energy information collected from national system operators (see [31] for details). We consider 8 representative day types, one weekday and one weekend day per season, as being representative of the different conditions faced by the system throughout the year.

We consider 4 day-ahead scheduling models: the DUCR model and the SUC model with 30 (*SUC30*), 60 (*SUC60*) and 120 (*SUC120*) scenarios. The sizes of the different day-ahead scheduling models are presented in Table 1, where the size of the stochastic models refers to the size of the extensive form. While the DUCR model is of the scale of problems that fit in the memory of a single machine and can be solved by a commercial solver, the SUC models in extensive form are beyond current capabilities of commercial solvers.

Table 1. Problem sizes.

Model	Scenarios	Variables	Constraints	Integers
DUCR	1	570.432	655.784	9.552
SUC30	30	13.334.400	16.182.180	293.088
SUC60	60	26.668.800	32.364.360	579.648
SUC120	120	53.337.600	64.728.720	1.152.768

Table 2 presents the solution time statistics for all day-ahead scheduling policies. In the case of SUC, we report these results for the two dual initialization alternatives proposed in section 3.2.

Table 2. Solution time statistics over 8 day types.

Model	Nodes used	Initialization	Running time [hours] avg. (min. – max.)	Worst final gap [%]
DUCR	1	–	1,9 (0,6 – 4,2)	0,95

SUC30	10	LP	1,1 (0,7 – 2,2)	0,93
	10	OPF	0,8 (0,3 – 1,8)	1,00
SUC60	10	LP	3,2 (1,1 – 8,4)	1,00
	10	OPF	1,5 (0,6 – 4,7)	0,97
SUC120	10	LP	> 6,1 (1,6 – 10,0)	1,68
	10	OPF	> 3,0 (0,6 – 10,0)	1,07

The results of Table 2 indicate that the OPF initialization significantly outperforms the LP approach in terms of termination time. This is mainly due to the fact that the OPF approach provides non-trivial lower bounds including information for all scenarios much faster than the LP approach. On the other hand, the solution times of SUC60 and DUCR indicate that, using distributed computing, we can solve SUC at a comparable run time to that required by commercial solvers for DUCR on large-scale systems. Moreover, as shown in Table 3, for a given hard constraint on solution wall time such as 2 hours (which is common for day-ahead power system operations), the proposed distributed algorithm provides solutions to SUC with up to 60 scenarios within 2% of optimality, which is acceptable for operational purposes.

Table 3. Worst optimality gap (over 8 day types) vs. solution wall time.

Model	**Initialization**	**Worst gap [%]**			
		1 hour	2 hours	4 hours	8 hours
SUC30	LP	7,59	1,02	0,93	
	OPF	1,90	1,00		
SUC60	LP	23,00	5,32	5,22	4,50
	OPF	4,60	1,57	1,03	0,97
SUC120	LP	70,39	31,66	4,61	1,87
	OPF	46,69	27,00	1,42	1,07

4. Scalable control for distributed energy resources

Nomenclature

Parameters

$T_{DF,i}$	fixed rounds of disabled refractory time for tier i
$T_{DV,i}$	maximum random rounds disabled refractory time for tier i
$T_{EF,i}$	fixed rounds of enabled refractory time for tier i
$T_{EV,i}$	maximum random rounds enabled refractory time for tier i
f	target minimum ratio of flexible to refractory demand
α	proportion of goal discrepancy corrected each round

Variables

$s(t,a)$	state of demand for agent a at time t
$s(t)$	state of total power demand (watts) at time t
$\hat{s}(t)$	estimate of $s(t)$
$\lvert X_{i,a}\rvert$	power demand (watts) in state X for color i at agent a
$\lvert X_i\rvert$	total power demand (watts) in state X for color i
$\lvert \hat{X}_i\rvert$	estimate of $\lvert X_i\rvert$
$g(t)$	goal total enabled demand for time t
$c(t,a)$	control state for agent a at time t
$p_{\text{off},i,a}$	probability of a flexible color i device disabling at agent a
$p_{\text{on},i,a}$	probability of a flexible color i device enable at agent a
D_i	demand for ith color and above

4.1. Overview

Residential demand response has gained significant attention in recent years as an underutilized source of flexibility in power systems, and is expected to become highly valuable as a balancing resource as increasing amounts of

renewable energy are being integrated into the grid. However, the mobilization of demand response by means of real-time pricing, which represents the economists' gold standard and can be traced back to the seminal work of Schweppe *et al.* [32], has so far fallen short of expectations due to several obstacles, including regulation issues, market structure, incentives to consumers and technological limitations.

The ColorPower architecture [7, 8, 9] aims at releasing the potent power of demand response by approaching electricity as a service of differentiated quality, rather than a commodity that residential consumers are willing to trade in real time [33]. In this architecture, the coordination problem of determining which devices should consume power at what times is solved through distributed aggregation and stochastic control. The consumer designates devices or device modes using priority tiers (colors). These tiers correspond to "service level" plans, which are easy to design and implement: we can simply map the "color" designations of electrical devices into plans. A "more flexible" color means less certainty of when a device will run (e.g., time when a pool pump runs), or lower quality service delivered by a device (e.g. wider temperature ranges, slower electrical vehicle charging). These types of economic decision-making are eminently compatible with consumer desires and economic design, as evidenced by the wide range of quality-of-service contracts offered in other industries.

Furthermore, the self-identified priority tiers of the ColorPower approach enable retail power participation in wholesale energy markets, lifting the economic obstacles for demand response: since the demand for power can be differentiated into tiers with a priority order, the demand in each tier can be separately bid into the current wholesale or local (DSO level) energy markets. The price for each tier can be set according to the cost of supplying demand response from that tier, which in turn is linked to the incentives necessary for securing customer participation in the demand response program. This allows aggregated demand to send price signals in the form of a decreasing buy bid curve. Market information thus flows bi-directionally. A small amount of flexible demand can then buffer the volatility of the overall power demand by yielding power to the inflexible devices as necessary (based upon the priority chosen by the customer), while fairly distributing power to all customer devices within a demand tier.

Technological limitations to the massive deployment of demand response are dealt with by deploying field-proven stochastic control techniques across the distribution network, with the objective of subtly shifting the schedules of millions of devices in real time, based upon the conditions of the grid. These control techniques include the CSMA/CD algorithms that permit cellular phones to share narrow radio frequency bands, telephone switch control algorithms, and operating system thread scheduling, as well as examples from nature such as social insect hive behaviours and bacterial quorum sensing. Moreover, the ubiquity of Internet communications allows us to consider using the Internet platform itself for end-to-end communications between machines.

At a high level, the ColorPower algorithm operates by aggregating the demand flexibility state information of each agent into a global estimate of total consumer flexibility. This aggregate and the current demand target are then broadcast via IP multicast throughout the system, and every local controller (typically one per consumer or one per device) combines the overall model and its local state to make a stochastic control decision. With each iteration of aggregation, broadcast, and control, the overall system moves toward the target demand, set by the utility or the ISO,TSO or DSO, allowing the system as a whole to rapidly achieve any given target demand and closely track target ramps. Note that aggregation has the beneficial side-effect of preserving the privacy of individual consumers: their demand information simply becomes part of an overall statistic.

The proposed architectural approach supplements the inadequacy of pure market-based control approaches by introducing an automated, distributed, and cooperative communications feedback loop between the system and large populations of cooperative devices at the edge of the network. TSO markets and the evolving DSO local energy markets of the future will have both deep markets and distributed control architecture pushed out to the edge of the network. This smart grid architecture for demand response in the mass market is expected to be a key asset in addressing the challenges of renewable energy integration and the transition to a low-carbon economy.

4.2. The ColorPower control problem

A ColorPower system consist of a set of n agents, each owning a set of electrical devices organized into k colors, where lower-numbered colors are intended to be shut off first (e.g., 1 for "green" pool pumps, 2 for "green" HVAC, 3 for "yellow" pool pumps, etc.), and where each color has its own time constants.

Within each color, every device is either *Enabled*, meaning that it can draw power freely, or *Disabled*, meaning that has been shut off or placed in a lower power mode. In order to prevent damage to appliances and/or customer annoyance, devices must wait through a *Refractory* period after switching between *Disabled* and *Enabled*, before they return to being *Flexible* and can switch again. These combinations give four device states (e.g., *Enabled* and *Flexible*, EF), through which each device in the ColorPower system moves according to the modified Markov model of Figure 2: randomly from EF to DR and DF to ER (becoming disabled with probability p_{off} and enabled with probability p_{on}) and by randomized timeout from ER to EF and DR to DF (a fixed length of $T_{.F}$ plus a uniform random addition of up to $T_{.V}$).

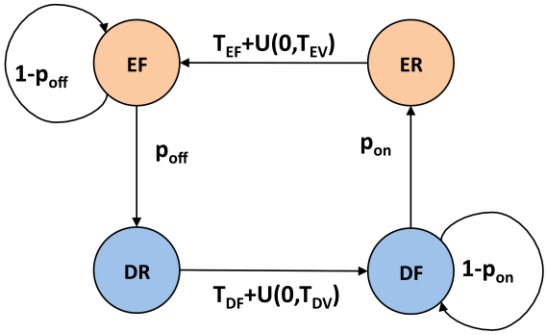

Figure 2: Markov model-based device state switching [8, 9].

The ColorPower control problem can then be stated as dynamically adjusting p_{on} and p_{off} for each agent and color tier, in a distributed manner, so that the aggregate consumption of the system follows a demand goal given by the operator of the high-voltage network.

4.3. The ColorPower architecture

The block diagram of the ColorPower control architecture is presented in Figure 3. Each ColorPower client (i.e., the controller inside a device) regulates the state transitions of the devices under its control. Each client state $s(t,a)$ is aggregated to produce a global state estimate $\hat{s}(t)$, which is broadcast along with a goal $g(t)$ (the demand target set by the utility or the ISO, TSO or DSO), allowing clients to shape demand by independently computing the control state $c(t,a)$.

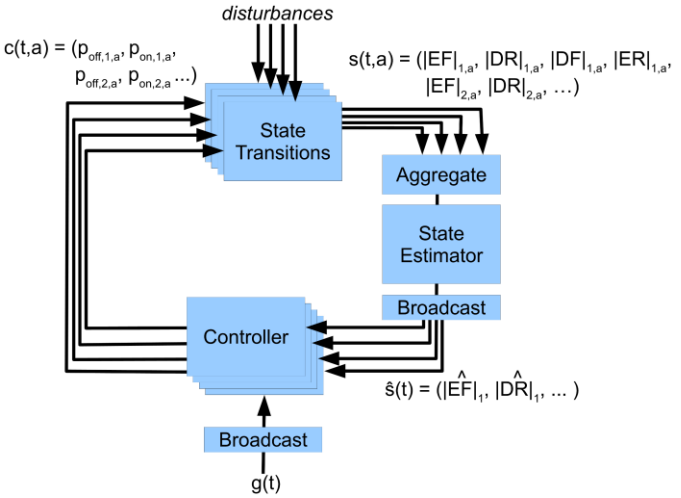

Figure 3: Block diagram of the control architecture [8, 9].

The state $s(t,a)$ of a client a at time t sums the power demands of the device(s) under its control, and these values are aggregated using a distributed algorithm (e.g. a spanning tree in [7]) and fed to a state estimator to get an overall estimate of the true state $\hat{s}(t)$ of total demand in each state for each color. This estimate is then broadcast to all clients (e.g. by gossip-like diffusion in [7]), along with the demand shaping goal $g(t)$ for the next total *Enabled* demand over all colors. The controller at each client a sets its control state $c(t,a)$, defined as the set of transition probabilities $p_{\text{on},i,a}$ and $p_{\text{off},i,a}$ for each color i. Finally, demands move through their states according to those transition probabilities, subject to exogenous disturbances such as changes in demand due to customer override, changing environmental conditions, imprecision in measurement, among others.

Note that the aggregation and broadcast algorithms must be chosen carefully in order to ensure that the communication requirements are lightweight enough to allow control rounds that last for a few seconds on low-cost hardware. The choice of algorithm depends on the network structure: for mesh networks, for example, spanning tree aggregation and gossip-based broadcast are fast and efficient (for details, see [7]).

4.4. ColorPower control algorithm

The ColorPower control algorithm, determines the control vector $c(t,a)$ by a stochastic controller formulated to satisfy four constraints:

Goal tracking: The total *Enabled* demand in $s(t)$ should track $g(t)$ as closely as possible: i.e., the sum of *Enabled* demand over all colors i should be equal to the goal. This is formalized as the equation:

$$g(t) = \sum_i (|EF_i| + |ER_i|).$$

Color priority: Devices with lower-numbered colors should be shut off before devices with higher-numbered colors. This is formalized as:

$$|EF_i| + |ER_i| = \begin{cases} D_i - D_{i+1} & \text{if } D_i \le g(t) \\ g(t) - D_{i+1} & \text{if } D_{i+1} \le g(t) < D_i \\ 0 & \text{otherwise,} \end{cases}$$

so that devices are *Enabled* from the highest color downward, where D_i is the demand for the ith color and above:

$$D_{\text{i}} = \sum_{j \ge i} \left(\left|EF_j\right| + \left|ER_j\right| + \left|DF_j\right| + |DR_j| \right).$$

Fairness: When the goal leads to some devices with a particular color being *Enabled* and other devices with that color being *Disabled*, each device has the same expected likelihood of being *Disabled*. This means that the control state is identical for every client.

Cycling: Devices within a color trade off which devices are *Enabled* and which are *Disabled* such that no device is unfairly burdened by initial bad luck. This is ensured by asserting the constraint:

$$(|EF_i| > 0) \cap (|DF_i| > 0) \Rightarrow \left(p_{\text{on},i,a} > 0\right) \cap \left(p_{\text{off},i,a} > 0\right).$$

This means that any color with a mixture of *Enabled* and *Disabled Flexible* devices will always be switching the state of some devices. For this last constraint, there is a tradeoff between how quickly devices cycle and how much flexibility is held in reserve for future goal tracking; we balance these with a target ratio f of the minimum ratio between pairs of corresponding *Flexible* and *Refractory* states.

Since the controller acts indirectly, by manipulating the p_{on} and p_{off} transition probabilities of devices, the only resources available for meeting these constraints is the demand in the flexible states EF and DF for each tier. When it is not possible to satisfy all four constraints simultaneously, the ColorPower controller prioritizes the constraints in order of their importance. Fairness and qualitative color guarantees are given highest priority, since these are part of the contract with customers: fairness by ensuring that the expected enablement fraction of each device is equivalent (though particular clients may achieve this in different ways, depending on their type and customer settings). Qualitative priority is handled by rules that prohibit flexibility from being considered by the controller outside of contractually allowable circumstances. Constraints are enforced sequentially. First comes goal tracking - the actual

shaping of demand to meet power schedules. Second is the soft color priority, which ensures that in those transient situations when goal tracking causes some devices to be in the wrong state, it is eventually corrected. Cycling is last, because it is defined only over long periods of time and thus is the least time critical to satisfy. A controller respecting the aforementioned constraints is described in [8].

4.5. Numerical experiment

We have implemented and tested the proposed demand response approach into the ColorPower software platform [8]. Simulations are executed with the following parameters: 10 trials per condition for 10.000 controllable devices, each device consumes 1 KW of power (for a total of 10 MW demand), devices are 20% green (low priority), 50% yellow (medium priority) and 30% red (high priority), the measurement error is ε = 0,1% (0,001), the rounds are 10 seconds long and all the *Refractory* time variables are 40 rounds. Error is measured by taking the ratio of the difference of a state from optimal versus the total power.

The results of the simulation test are illustrated in Figure 4. When peak control is desired, the aggregate demand remains below the quota, while individual loads are subjected stochastically to brief curtailments. Post-event rush-in, a potentially severe problem for both traditional demand response and price signal-based control systems, is also managed gracefully due to the specific design of the modified Markov model of Figure 2.

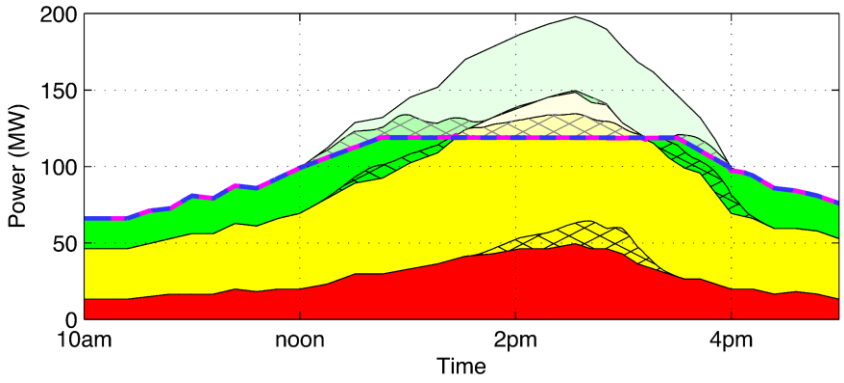

Figure 4: Simulation results with 10.000 independently fluctuating power loads. Demand is shown as a stacked graph, with Enabled demand on the bottom in saturated colors, Disabled demand at the top in pale colors, and Refractory demand cross-hatched. The goal is the dashed blue line and the current total Enabled demand is the solid magenta line. The plot illustrates a peak shaving case where a power quota, the demand response target that may be provided from an externally-generated demand forecast, is used as a guide for the demand to follow.

Taken together, these results indicate that the ColorPower approach, when coupled with an appropriate controller, should have the technological capability to flexibly and resiliently shape demand in most practical deployment scenarios.

5. Conclusions

We present two applications of distributed computing in power systems. On the one hand, we optimize high-voltage power system operations using a distributed asynchronous algorithm capable of solving stochastic unit commitment in comparable run times to those of a deterministic unit commitment model with reserve requirements, and within operationally acceptable time frames. On the other hand, we control demand response at the distribution level using stochastic distributed control, thereby enabling large-scale demand shaping during real-time operations of power systems. Together, both applications of distributed computing demonstrate the potential for efficiently managing flexible resources in smart grids and for systematically coping with the uncertainty and variability introduced by renewable energy.

Acknowledgments

The authors acknowledge the Fair Isaac Corporation FICO for providing licenses for Xpress, and the Lawrence Livermore National Laboratory for granting access and computing time at the Sierra cluster.

This research was funded by the ENGIE Chair on Energy Economics and Energy Risk Management and by the Université catholique de Louvain through an FSR grant.

References

[1] X. Fang, S. Misra, G. Xue and D. Yang, "Smart Grid — The New and Improved Power Grid: A Survey," in IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 944-980, Fourth Quarter 2012.

[2] V.C. Ramesh, "On distributed computing for on-line power system applications," International Journal of Electrical Power & Energy Systems, Volume 18, Issue 8, Pages 527-533, 1996.

[3] D. Falcão, "High performance computing in power system applications," in Vector and Parallel Processing – VECPAR'96 (J. Palma and J. Dongarra, eds.), vol. 1215 of Lecture Notes in Computer Science, pp. 1-23, Springer Berlin Heidelberg, 1997.

[4] M. Shahidehpour and Y. Wang, Communication and Control in Electric Power Systems: Appications of parallel and distributed processing, Wiley-IEEE Press, July 2003.

[5] S. Bera, S. Misra and J. J. P. C. Rodrigues, "Cloud Computing Applications for Smart Grid: A Survey," in IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 5, pp. 1477-1494, May 2015.

[6] I. Aravena and A. Papavasiliou, "A distributed asynchronous algorithm for the two-stage stochastic unit commitment problem," 2015 IEEE Power & Energy Society General Meeting, Denver, CO, 2015, pp. 1-5.

[7] Vinayak V. Ranade and Jacob Beal, "Distributed Control for Small Customer Energy Demand Management", IEEE Self-Adaptive and Self-Organization 2010, Sept 2010.

[8] Jacob Beal, Jeffrey Berliner and Kevin Hunter, "Fast Precise Distributed Control for Energy Demand Management," 2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Lyon, 2012, pp. 187-192.

[9] A. Papalexopoulos, J. Beal and S. Florek, "Precise Mass-Market Energy Demand Management Through Stochastic Distributed Computing," IEEE Transactions on Smart Grid, vol. 4, no. 4, pp. 2017-2027, Dec. 2013.

[10] M. Caramanis, E. Ntakou, W. W. Hogan, A. Chakrabortty and J. Schoene, "Co-Optimization of Power and Reserves in Dynamic T&D Power Markets with Nondispatchable Renewable Generation and Distributed Energy Resources," in Proceedings of the IEEE, vol. 104, no. 4, pp. 807-836, April 2016.

[11] APX Group, Belpex, Cegedel Net, EEX, ELIA Group, EnBw, E-On Netz, Powernext, RTE, RWE, and TenneT, "A report for the regulators of the Central West European (CWE) region on the final design of the market coupling solution in the region, by the CWE MC Project," January 2010.

[12] 50Hertz Transmission GmbH, Amprion GmbH, Elia System Operator NV, TenneT TSO B.V., TenneT TSO GmbH, and TransnetBW GmbH, "Potential cross-border balancing cooperation between the Belgian, Dutch and German electricity Transmission System Operators," October 2014.

[13] PJM, "PJM Manual 11: Energy & Ancillary Services Market Operations," December 2015.

[14] Midcontinent ISO, "Energy and Operating Reserve Markets Business Practice Manual," March 2016.

[15] 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, Aug 1996.

[16] P. Carpentier, G. Gohen, J.-C. Culioli, and A. Renaud, "Stochastic optimization of unit commitment: a new decomposition framework," IEEE Transactions on Power Systems, vol. 11, pp. 1067–1073, May 1996.

[17] M. Tahanan, W. van Ackooij, A. Frangioni, and F. Lacalandra, "Large-scale unit commitment under uncertainty," 4OR, 13(2):115–171, 2015.

[18] J.P. Deane, G. Drayton, B.P. Ó Gallachóir, "The impact of sub-hourly modelling in power systems with significant levels of renewable generation," Applied Energy, Volume 113, pp. 152-158, January 2014.

[19] A. Frangioni, C. Gentile and F. Lacalandra, "Tighter Approximated MILP Formulations for Unit Commitment Problems," in IEEE Transactions on Power Systems, vol. 24, no. 1, pp. 105-113, Feb. 2009.

[20] D. Rajan and S. Takriti. Minimum up/down polytopes of the unit commitment problem with start-up costs. IBM Research Report RC23628, Thomas J. Watson Research Center, June 2005.

[21] A. Papavasiliou, S. S. Oren and B. Rountree, "Applying High Performance Computing to Transmission-Constrained Stochastic Unit Commitment for Renewable Energy Integration," in IEEE Transactions on Power Systems, vol. 30, no. 3, pp. 1109-1120, May 2015.

[22] K. Kim and V.M. Zavala, "Algorithmic innovations and software for the dual decomposition method applied to stochastic mixed-integer programs," Optimization Online, 2015.

[23] K. Cheung, D. Gade, C. Silva-Monroy, S.M. Ryan, J.P. Watson, R.J.B. Wets, and D.L. Woodruff, "Toward scalable stochastic unit commitment. Part 2: solver configuration and performance assessment," Energy Systems, 6(3):417–438, 2015.

[24] R.T. Rockafellar and R.J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," Mathematics of Operations Research, vol. 16, no. 1, pp. 119–147, 1991.

[25] A. Nedić, D. Bertsekas, and V. Borkar, "Distributed asynchronous incremental subgradient methods," in Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications (Y. Butnariu, S. Reich and Y. Censor eds.), vol. 8 of Studies in Computational Mathematics, pp. 381–407, Amsterdam: Elsevier, 2001.

[26] Yuri Ermoliev. Stochastic quasigradient methods and their application to system optimization. Stochastics, 9(1-2):1–36, 1983.

[27] K. Kiwiel, "Convergence of approximate and incremental subgradient methods for convex optimization," SIAM Journal on Optimization, vol. 14, no. 3, pp. 807–840, 2004.

[28] Shabbir Ahmed, "A scenario decomposition algorithm for 0–1 stochastic programs," Operations Research Letters, Volume 41, Issue 6, November 2013.

[29] Y. Colombani and S. Heipcke. Multiple models and parallel solving with Mosel, February 2014. Avalaible at: http://community.fico.com/docs/DOC-1141.

[30] N. Hutcheon and J. Bialek, "Updated and validated power flow model of the main continental European transmission network," in 2013 IEEE PowerTech Grenoble, pp. 1–5, June 2013.

[31] I. Aravena and A. Papavasiliou, "Renewable Energy Integration in Zonal Markets," forthcoming in IEEE Transactions on Power Systems.

[32] F. C. Schweppe, R. D. Tabors, J. L. Kirtley, H. R. Outhred, F. H. Pickel and A. J. Cox, "Homeostatic Utility Control," IEEE Transactions on Power Apparatus and Systems, vol. PAS-99, no. 3, pp. 1151-1163, May 1980.

[33] Shmuel S. Oren, "Product Differentiation in Service Industries". Working paper presented at the First Annual Conference on Pricing, New York, NY, December 1987.