

A two-level approach to large mixed-integer programs with application to cogeneration in energy-efficient buildings

Fu Lin¹ · Sven Leyffer¹ · Todd Munson¹

Received: 11 May 2015 / Published online: 12 April 2016
© Springer Science+Business Media New York (outside the USA) 2016

Abstract We study a two-stage mixed-integer linear program (MILP) with more than 1 million binary variables in the second stage. We develop a two-level approach by constructing a semi-coarse model that coarsens with respect to variables and a coarse model that coarsens with respect to both variables and constraints. We coarsen binary variables by selecting a small number of prespecified on/off profiles. We aggregate constraints by partitioning them into groups and taking convex combination over each group. With an appropriate choice of coarsened profiles, the semi-coarse model is guaranteed to find a feasible solution of the original problem and hence provides an upper bound on the optimal solution. We show that solving a sequence of coarse models converges to the same upper bound with proven finite steps. This is achieved by adding violated constraints to coarse models until all constraints in the semi-coarse model are satisfied. We demonstrate the effectiveness of our approach in cogeneration for buildings. The coarsened models allow us to obtain good approximate solutions at a fraction of the time required by solving the original problem. Extensive numerical

The submitted manuscript has been created by the UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”) under Contract No. DE-AC02-06CH11357 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

✉ Fu Lin
fulin@mcs.anl.gov
Sven Leyffer
leyffer@mcs.anl.gov
Todd Munson
tmunson@mcs.anl.gov

¹ Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Lemont, IL 60439, USA

experiments show that the two-level approach scales to large problems that are beyond the capacity of state-of-the-art commercial MILP solvers.

Keywords Coarsened models · Distributed generation · Large-scale problems · Two-level approach · Multi-period planning · Resource and cost allocation · Two-stage mixed-integer programs

Mathematics Subject Classification 91B32 · 90C06 · 90C11 · 90C90

1 Problem definition and motivation

We consider a hierarchical two-stage mixed-integer linear program (MILP) with integer variables in both the first and second stages. We are particularly interested in applications where the first-stage integer variables model design or purchasing decisions and the second-stage variables model operational decisions over a long time horizon (e.g., hourly operations over a decadal horizon). The goal is to take operational constraints into account when making a capital investment decision. Thus, our model is complicated by the fact that second-stage variables include both binary (on/off) decisions and continuous variables that model operational settings. Examples include the design of cogeneration units for commercial buildings subject to operational conditions [28–30, 33, 34] and transmission network expansion subject to unit commitment constraints [1, 23]. Models of this class can involve hundreds of thousands or even millions of binary variables and are beyond the scope of today’s state-of-the-art solvers.

We consider a two-stage MILP with m first-stage variables and three sets of N second-stage variables. The first-stage variables are $y \in \mathbb{Z}_+^m$ (non-negative integers of dimension m). We have three classes of second-stage variables: (1) on/off decisions, $x \in \{0, 1\}^N$; (2) operational settings $v \in \mathbb{R}^N$ that are switched on or off by x , that is, $Lx \leq v \leq Ux$ for finite bounds $L \leq U$; and (3) other second-stage variables $0 \leq w \in \mathbb{R}^N$. The MILP model is described by

$$\begin{aligned}
 & \underset{y,x,v,w}{\text{minimize}} && a^T y + b^T x + c^T v + d^T w \\
 & \text{subject to} && Ay + Bx + Cv + Dw \leq f \\
 & && y \in \mathbb{Z}_+^m \\
 & && x \in \{0, 1\}^N \\
 & && v \in \mathbb{R}^N, \quad Lx \leq v \leq Ux \\
 & && w \in \mathbb{R}^N, \quad w \geq 0,
 \end{aligned} \tag{1.1}$$

where $a \in \mathbb{R}^m$, $b, c, d \in \mathbb{R}^N$, $A \in \mathbb{R}^{M \times m}$, and $B, C, D \in \mathbb{R}^{M \times N}$. We assume that $m \ll N$ and $m \ll M$; that is, the number of first-stage variables is much smaller than the number of second-stage binary variables and the number of coupling constraints.

In addition to a large number of binary and continuous variables in the second stage, the challenges of model (1.1) also arise from coupling constraints. We do not assume any sparsity structure in matrices B, C , and D . This is in contrast to the arrow-type sparsity structure that arises, for example, in stochastic programs [6]. Therefore, fixing

the first-stage variable y does not decompose (1.1) into scenario-based subproblems. Moreover, the coupling constraints in (1.1) are not amenable to decomposition methods such as Lagrangian relaxation [14, 18]. The reason is that the number of coupling constraints, M , is of the same order as the number of variables, N . Dualizing all coupling constraints results in a large number of dual variables; hence, Lagrangian relaxation of (1.1) is unlikely to yield efficient decomposition methods [14].

Problem (1.1) arises in several applications. In particular, our work is motivated by the cogeneration problem with renewable energy for commercial buildings [28, 30, 33, 34]. In this case, the first-stage design involves investment decisions for cogeneration units such as fuel cells, solar panels, and battery storage. The second-stage problem aims at optimal on/off hourly operation that takes into account technology specifics such as the minimum and maximum power generation. Our goal is to include operational constraints in the design of cogeneration.

One of our objectives is to find the optimal first-stage solution of (1.1). However, the two-stage MILP with a large number of binary variables at the second stage is beyond the scope of state-of-the-art commercial MIP solvers. For example, a typical cogeneration model with a ten-year horizon results in 1.05 million binary variables ($3650 \text{ days} \times 24 \text{ h} \times 12 \text{ units}$). On the other hand, a naive approach that solves (1.1) with a short horizon at the second stage provides first-stage designs that are suboptimal for a long horizon problem. The reason is that short horizon problems do not take into account coupling constraints over a long horizon. Moreover, the problem data of short-horizon problems are not representative of long-horizon problems, resulting in suboptimal solutions.

We develop a *two-level approach* that coarsens the hourly on/off variables to daily operation profiles. Since the profile representation yields a model with many fewer variables, we refer to this step as the *primal (variable) coarsening*. The resulting *semi-coarse model* still contains the same number of constraints as in (1.1). We reduce the number of constraints by partitioning them into groups that are of the same size as the profiles and summing over each group. This aggregation of constraints results in a relaxed problem whose solutions may not be feasible for the original MILP model. We include the violated constraints, re-solve the coarsened MILP model, and repeat this process until all constraints are satisfied. We refer to the aggregation of constraints as the *dual (constraint) coarsening* and the resulting MILP as the *coarse model*.

1.1 Literature review

The idea of aggregating variables and constraints to build approximate optimization models is not new. Zipkin studies the effect of variable aggregation and row aggregation for linear programs in [40, 41]. This framework is extended to stochastic linear programs in [5, 7]. For integer programs, a large amount of work focuses on aggregating constraints into one or more *surrogate* constraints for which one can show that the solution of the original problem and that of the surrogate problem are identical. Early work includes [2, 17, 19, 20]; see also the survey paper [31].

We note that the theory of surrogate constraints focuses mainly on integer programs with non-negative integer coefficients. The seminal work in [25] results in an exponen-

tial grow of the coefficients in the surrogate constraints. Many refinements have been developed, including simultaneous and sequential aggregation schemes, to reduce the magnitude of coefficients [31]. For mixed-integer programs with real-valued coefficients, however, we are not aware of aggregation schemes that guarantee equivalence between the original problem and the surrogate problem. In our two-level approach, we employ constraint aggregation as a relaxation technique to identify a smaller set of constraints that are active at the optimal solution. This is achieved by solving a sequence of MILPs and adding the violated constraints until all constraints in the original MILP are satisfied. Furthermore, we take advantage of the LP warm-start to reduce the number of MILP re-solves and the computational time of each MILP.

A large body of literature appeared in the 1990s on bilevel or multilevel mixed-integer programs; see [10,22,26,37,38] and survey papers [35,36]. One of the main theoretical emphases is on decentralized decision-making from a game-theoretic point of view [36]. Optimality conditions for convex bilevel programs have been established in [35]. A heuristic-based branch-and-bound method is developed in [22,38], and tabu search method is introduced in [37]. Recent years have seen specific application-driven algorithms ranging from infrastructure protection planning [32] to vulnerability analysis of power grids [27].

In this context, the multilevel method is specifically connected to the problem formulation. In contrast, our two-level method concentrates on algorithmic development for mixed-integer linear programs. In particular, our two-level approach builds optimization models of different resolutions from a fine-level problem to a coarse-level problem. We develop a systematic procedure that coarsens binary and continuous variables in addition to the aggregation of constraints. Our approach allows us to solve large MILPs with more than one million binary and continuous variables and coupling constraints. Extensive numerical experiments have been conducted to verify the efficiency of the developed algorithm.

Our two-level approach is reminiscent of multigrid methods in linear algebra and solution of partial differential equations. In [16], a multilevel iterative method for generic optimization problem is discussed. In [21], a recursive trust-region method for nonlinear unconstrained problems is developed. In [39], a line search multigrid approach for nonlinear programs is proposed.

1.2 Our contributions

Our contributions are fourfold. First, we develop a systematic two-level approach for two-stage MILPs with a large number of binary and continuous variables in the second stage. The coarsening of binary variables is done by introducing profiles (vectors) with binary elements. By selecting one profile from a small number of candidate profiles, we reduce the number of binary variables by orders of magnitude. In addition, we reduce the number of continuous variables by using a convex combination of profiles with real elements. We show that the semi-coarse model with coarsening in variables results in a tightening of the original MILP and therefore provides an upper bound on the optimal solution.

Second, we propose a simple method for constraint aggregation that does not require any sparsity assumptions on the coefficient matrices. By partitioning constraints into

groups and taking a convex combination of constraints in each group, we obtain a relaxation of the semi-coarse model with many fewer constraints. We solve a sequence of coarse MILPs by adding any violated constraints until all constraints are satisfied. We show that the LP-relaxation of the coarse MILPs can be used to warm-start the MILP re-solves and significantly reduce the computational effort.

Third, we apply our two-level approach to the design of cogeneration in buildings. The resulting complex MILP model has a large number of coupling constraints over a long time horizon. We develop a moving-horizon method to generate valid profiles and show by construction that the generated profiles satisfy coupling constraints in time, further reducing the number of constraints in the coarsened models.

Fourth, we verify the efficiency of our algorithm using a variety of examples generated from simulation programs for commercial buildings. While our two-level approach requires solving a sequence of coarse MILPs, numerical results indicate that the first iterate provides a good approximate solution of the semi-coarse models. Through extensive numerical experiments, we demonstrate the scalability of the two-level approach on a rich set of large problems that are beyond the capacity of state-of-the-art commercial solvers.

Outline. The remainder of this paper is organized as follows. In Sect. 2, we describe the two-level approach for the two-stage MILPs. We show that our approach results in a tightening of the original problem and provides an upper bound on the optimal solution. Under the assumption of periodic problem data and no coupling between time periods, our two-level approach provides the optimal solution. In Sect. 3, we apply our approach to a complex MILP model from the cogeneration problem in buildings. We discuss how the profiles are generated and selected such that the coarsened models provide feasible solutions to (1.1). In Sect. 4, we demonstrate the effectiveness of our approach on a diverse set of test problems. We show that the two-level approach allows us to find good approximate solutions with a fraction of computational time compared with solving the full model. In Sect. 5, we summarize our contribution and discuss future extensions.

2 Two-level approach to MILP

In this section, we derive two models that are formulated in terms of the first-stage variables and $n \ll N$ second-stage variables. We start by presenting our ideas for variable coarsening, resulting in a semi-coarse model that has $\mathcal{O}(n)$ second-stage variables and $\mathcal{O}(N)$ second-stage constraints. Next we show how to further coarsen this model by aggregating constraints, resulting in a coarse model with both $\mathcal{O}(n)$ second-stage variables and constraints.

2.1 Primal coarsening

To coarsen the variables in the model, we introduce a coarsening factor $\delta \in \mathbb{Z}_+$ and group N second-stage variables into $n = N/\delta$ equal-sized groups of size δ (the assumption that the groups are equal-sized is not critical):

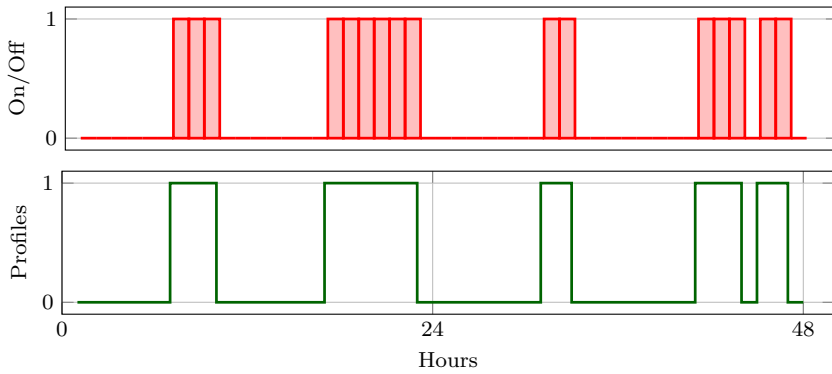


Fig. 1 The *first row* shows the hourly on/off variables $x_i \in \{0, 1\}$ for 48 h. The *second row* shows the profile representation with two daily profiles $\bar{X}_k \in \{0, 1\}^{24}$

$$\mathbf{x}_i = \begin{pmatrix} x_{\delta(i-1)+1} \\ \vdots \\ x_{\delta i} \end{pmatrix} \quad \text{for } i = 1, \dots, n. \quad (2.1)$$

We define groups for v and w analogously. These groups correspond to a partition of the second-stage variables x , v , and w :

$$x = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}, \quad v = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{pmatrix}, \quad w = \begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_n \end{pmatrix}. \quad (2.2)$$

Next, we introduce δ -profiles, which are fixed parameter values for a group of variables:

$$\bar{X}_k \in \{0, 1\}^\delta \quad \text{for } k = 1, \dots, K, \quad (2.3)$$

where K is a positive integer. We show in Sect. 3 how these profiles are generated and selected. Figure 1 illustrates that $N = 48$ hourly on/off variables can be represented by $n = 2$ daily profiles with length $\delta = 24$. Now for every \bar{X}_k , we collect a set of I_k operational δ -profiles for \bar{V} ,

$$\bar{V}_{jk} \in \mathbb{R}^\delta, \quad \text{with } L\bar{X}_k \leq \bar{V}_{jk} \leq U\bar{X}_k \quad \text{for } j = 1, \dots, I_k, \quad k = 1, \dots, K, \quad (2.4)$$

and a set of J operational δ -profiles for \bar{W} ,

$$0 \leq \bar{W}_j \in \mathbb{R}^\delta \quad \text{for } j = 1, \dots, J, \quad (2.5)$$

where J is a positive integer.

Given these sets of δ -profiles, we perform a change of variables that replaces the second-stage variables (x, v, w) by a reduced set of variables $(\bar{x}, \bar{v}, \bar{w})$ for the δ -profiles, resulting in a coarsening of the second-stage variables. In particular, we set

$$\mathbf{x}_i = \sum_{k=1}^K \bar{x}_{ik} \bar{X}_k, \quad \sum_{k=1}^K \bar{x}_{ik} \leq 1, \quad \bar{x}_{ik} \in \{0, 1\} \quad \text{for } i = 1, \dots, n, \quad k = 1, \dots, K. \tag{2.6}$$

Thus, we have that $\bar{x} \in \{0, 1\}^{n \times K}$, and our goal is to create models where $nK \ll N$. Similarly, we write v and w as

$$\mathbf{v}_i = \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \bar{V}_{jk}, \quad \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \leq 1, \quad \sum_{j=1}^{I_k} \bar{v}_{ijk} = \bar{x}_{ik}, \quad 0 \leq \bar{v}_{ijk} \leq 1, \tag{2.7}$$

for $i = 1, \dots, n,$

and

$$\mathbf{w}_i = \sum_{j=1}^J \bar{w}_{ij} \bar{W}_j, \quad \sum_{j=1}^J \bar{w}_{ij} \leq 1, \quad 0 \leq \bar{w}_{ij} \leq 1, \quad \text{for } i = 1, \dots, n. \tag{2.8}$$

We note that we do not enforce $\bar{w}_{ij} \in \{0, 1\}$ or $\bar{v}_{ijk} \in \{0, 1\}$. The reason is that we wish to allow more freedom for choosing operational profiles in the second stage as long as they remain feasible. This choice also simplifies the coarsened second-stage problem and provides a valid approach if we assume that convex combinations of operational δ -profiles \bar{V}_{jk} and \bar{W}_j are feasible, which is often the case in practice.

The partition of variables (x, v, w) implies a partition of the problem matrices $B, C,$ and D of (1.1) as

$$B = [B_1 : \dots : B_n], \quad C = [C_1 : \dots : C_n], \quad D = [D_1 : \dots : D_n],$$

where $B_i, C_i,$ and $D_i \in \mathbb{R}^{M \times \delta}$ for $i = 1, \dots, n$. Next, we define δ -profile matrices as

$$\begin{aligned} \bar{X} &= [\bar{X}_1 : \dots : \bar{X}_K], \quad \bar{V} = [\bar{V}_{11} : \dots : \bar{V}_{1I_1} : \dots : \bar{V}_{K1} : \dots : \bar{V}_{KI_K}], \\ \bar{W} &= [\bar{W}_1 : \dots : \bar{W}_J], \end{aligned} \tag{2.9}$$

where $\bar{X}_k, \bar{V}_{jk},$ and \bar{W}_j are vectors of length δ . We define aggregated (coarse) matrices $\bar{B}, \bar{C},$ and \bar{D} as

$$\bar{B} = [B_1 \bar{X} : \dots : B_n \bar{X}], \quad \bar{C} = [C_1 \bar{V} : \dots : C_n \bar{V}], \quad \bar{D} = [D_1 \bar{W} : \dots : D_n \bar{W}], \tag{2.10}$$

where $\bar{B} \in \mathbb{R}^{M \times nK}, \bar{C} \in \mathbb{R}^{M \times n\mathcal{I}},$ and $\bar{D} \in \mathbb{R}^{M \times nJ},$ where $\mathcal{I} = \sum_{k=1}^K I_k$. Similarly, we aggregate the coefficient vectors in the objective function to obtain cost vectors $\bar{b}, \bar{c},$ and \bar{d}

$$\bar{b} = [b_1^T \bar{X} : \cdots : b_n^T \bar{X}]^T, \quad \bar{c} = [c_1^T \bar{V} : \cdots : c_n^T \bar{V}]^T, \quad \bar{d} = [d_1^T \bar{W} : \cdots : d_n^T \bar{W}]^T, \quad (2.11)$$

where $\bar{b} \in \mathbb{R}^{nK}$, $\bar{c} \in \mathbb{R}^{nI}$, and $\bar{d} \in \mathbb{R}^{nJ}$.

We obtain the following aggregated MILP, which we refer to as the *semi-coarse MILP*, because it has been coarsened in the primal variables only:

$$\begin{aligned} & \underset{y, \bar{x}, \bar{v}, \bar{w}}{\text{minimize}} && a^T y + \bar{b}^T \bar{x} + \bar{c}^T \bar{v} + \bar{d}^T \bar{w} \\ & \text{subject to} && Ay + \bar{B}\bar{x} + \bar{C}\bar{v} + \bar{D}\bar{w} \leq f \\ & && \sum_{k=1}^K \bar{x}_{ik} \leq 1, \quad \bar{x}_{ik} \in \{0, 1\} \\ & && \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \leq 1, \quad \sum_{j=1}^{I_k} \bar{v}_{ijk} = \bar{x}_{ik}, \quad 0 \leq \bar{v}_{ijk} \leq 1 \\ & && \sum_{j=1}^J \bar{w}_{ij} \leq 1, \quad 0 \leq \bar{w}_{ij} \leq 1. \end{aligned} \quad (2.12)$$

The MILP (2.12) has potentially many fewer variables than (1.1) but contains as many constraints as the original problem. Before we show how to aggregate constraints in the next section, we finish this section by summarizing some of the properties of (2.12).

Proposition 1 *Let $(\bar{x}, \bar{v}, \bar{w})$ be a feasible point of the semi-coarse model (2.12). Then it follows that the corresponding fine-scale variables*

$$x = \begin{pmatrix} \sum_{k=1}^K \bar{x}_{1k} \bar{X}_k \\ \vdots \\ \sum_{k=1}^K \bar{x}_{nk} \bar{X}_k \end{pmatrix}, \quad v = \begin{pmatrix} \sum_{k,j=1}^{K, I_k} \bar{v}_{1jk} \bar{V}_{jk} \\ \vdots \\ \sum_{k,j=1}^{K, I_k} \bar{v}_{njk} \bar{V}_{jk} \end{pmatrix}, \quad w = \begin{pmatrix} \sum_{j=1}^J \bar{w}_{1j} \bar{W}_j \\ \vdots \\ \sum_{j=1}^J \bar{w}_{nj} \bar{W}_j \end{pmatrix} \quad (2.13)$$

are feasible in the original MILP (1.1).

Proof The binary constraint $x \in \{0, 1\}^N$ follows from the representation (2.6) and the definition of binary vectors \bar{X}_k . Similarly, the non-negativity of w is a direct consequence of non-negative profiles \bar{W}_j and non-negative coefficients \bar{w}_{ij} . To show that v is feasible, we start with the definition of \bar{V}_{jk} in (2.4):

$$L \cdot \bar{X}_k \leq \bar{V}_{jk} \leq U \cdot \bar{X}_k.$$

Since $\bar{v}_{ijk} \geq 0$, it follows that

$$L \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \bar{X}_k \leq \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \bar{V}_{jk} \leq U \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \bar{X}_k.$$

Using $\bar{x}_{ik} = \sum_{j=1}^{I_k} \bar{v}_{ijk}$, we have

$$L \sum_{k=1}^K \bar{x}_{ik} \bar{X}_k \leq v_i \leq U \sum_{k=1}^K \bar{x}_{ik} \bar{X}_k, \quad \text{for } i = 1, \dots, n.$$

From the definition of x_i in (2.13), it follows that $Lx_i \leq v_i \leq Ux_i$, for $i = 1, \dots, n$, and thus $Lx \leq v \leq Ux$. The proof is complete by noting that

$$f \geq Ay + \bar{B}\bar{x} + \bar{C}\bar{v} + \bar{D}\bar{w} = Ay + Bx + Cv + Dw,$$

where the equality follows from the definition of aggregated matrices (2.10) and the representation of the fine-scale variables (2.13). □

Next, we show that the semi-coarse model provides an upper bound.

Proposition 2 *The semi-coarse model (2.12) is a tightening of the original MILP (1.1), and its solution provides an upper bound on (1.1). The two problems are equivalent if all optimal profiles from the solution of (1.1) are included in (2.12).*

Proof Let z_{MILP}^* and z_{semi}^* be the optimal value of the original MILP (1.1) and the semi-coarse model (2.12), respectively. Since any feasible point of the semi-coarse model (2.12) is a feasible point of the original MILP (1.1), it follows that (2.12) is a tightening of (1.1) and thus

$$z_{\text{MILP}}^* \leq z_{\text{semi}}^* .$$

Let (x^*, v^*, w^*) be the optimal solution of MILP (1.1). We now extract optimal profiles $(\bar{X}^*, \bar{V}^*, \bar{W}^*)$ corresponding to (x^*, v^*, w^*) . Then it follows that there exists a solution of (2.12) such that

$$x_i^* = \sum_{k=1}^K \bar{x}_{ik}^* \bar{X}_k^*, \quad v_i^* = \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk}^* \bar{V}_{jk}^*, \quad w_i^* = \sum_{j=1}^J \bar{w}_{ij}^* \bar{W}_j^*, \quad \text{for } i = 1, \dots, n.$$

Moreover, the solution $(\bar{x}^*, \bar{v}^*, \bar{w}^*)$ is feasible in (2.12), and the objective value is the same as the objective value of (x^*, v^*, w^*) . Because (2.12) is a tightening of (1.1), there cannot be a better solution than $(\bar{x}^*, \bar{v}^*, \bar{w}^*)$ that we constructed. Hence, $(\bar{x}^*, \bar{v}^*, \bar{w}^*)$ is the optimal solution, and both problems (1.1) and (2.12) are equivalent. □

2.2 Dual coarsening

We next coarsen the constraints by partitioning them into n groups of size δ and summing over each group. We define the rows of A as A_i^T , and we define a set of aggregated constraints by taking convex combination over the rows of A within each group

$$\hat{A} := \begin{pmatrix} \sum_{i=1}^{\delta} A_i^T \lambda_i \\ \sum_{i=1}^{\delta} A_{\delta+i}^T \lambda_{\delta+i} \\ \vdots \\ \sum_{i=1}^{\delta} A_{\delta(n-1)+i}^T \lambda_{\delta(n-1)+i} \end{pmatrix},$$

where the non-negative weights λ 's satisfy

$$\lambda_{\delta(k-1)+i} \in [0, 1], \quad \sum_{i=1}^{\delta} \lambda_{\delta(k-1)+i} = 1, \quad \text{for } k = 1, \dots, n.$$

We aggregate the constraint matrices \bar{B} , \bar{C} , and \bar{D} in a similar way:

$$\hat{\bar{B}} := \begin{pmatrix} \sum_{i=1}^{\delta} \bar{B}_i^T \lambda_i \\ \sum_{i=1}^{\delta} \bar{B}_{\delta+i}^T \lambda_{\delta+i} \\ \vdots \\ \sum_{i=1}^{\delta} \bar{B}_{\delta(n-1)+i}^T \lambda_{\delta(n-1)+i} \end{pmatrix}, \quad \hat{\bar{C}} := \begin{pmatrix} \sum_{i=1}^{\delta} \bar{C}_i^T \lambda_i \\ \sum_{i=1}^{\delta} \bar{C}_{\delta+i}^T \lambda_{\delta+i} \\ \vdots \\ \sum_{i=1}^{\delta} \bar{C}_{\delta(n-1)+i}^T \lambda_{\delta(n-1)+i} \end{pmatrix},$$

$$\hat{\bar{D}} := \begin{pmatrix} \sum_{i=1}^{\delta} \bar{D}_i^T \lambda_i \\ \sum_{i=1}^{\delta} \bar{D}_{\delta+i}^T \lambda_{\delta+i} \\ \vdots \\ \sum_{i=1}^{\delta} \bar{D}_{\delta(n-1)+i}^T \lambda_{\delta(n-1)+i} \end{pmatrix}.$$

Note that we use “bar” notation to denote coarsening in variables and “hat” notation to denote coarsening in constraints. We coarsen the right-hand side f analogously and obtain the *coarse MILP*:

$$\begin{aligned}
 & \underset{y, \bar{x}, \bar{v}, \bar{w}}{\text{minimize}} && a^T y + \bar{b}^T \bar{x} + \bar{c}^T \bar{v} + \bar{d}^T \bar{w} \\
 & \text{subject to} && \hat{A}y + \hat{B}\bar{x} + \hat{C}\bar{v} + \hat{D}\bar{w} \leq \hat{f} \\
 & && \sum_{k=1}^K \bar{x}_{ik} \leq 1, \quad \bar{x}_{ik} \in \{0, 1\} \\
 & && \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \leq 1, \quad \sum_{j=1}^{I_k} \bar{v}_{ijk} = \bar{x}_{ik}, \quad 0 \leq \bar{v}_{ijk} \leq 1 \\
 & && \sum_{j=1}^J \bar{w}_{ij} \leq 1, \quad 0 \leq \bar{w}_{ij} \leq 1.
 \end{aligned} \tag{2.14}$$

It follows easily that (2.14) is a relaxation of (2.12), because we have simply aggregated the constraints. We can use this fact to develop a simple algorithm that solves (2.12) by solving a sequence of tighter relaxations. The main idea is that after solving a relaxation (2.14) we can check whether all constraints in (2.12) are satisfied and add any violated constraints to (2.14). It follows easily that this algorithm is finite, because after finitely many constraints have been added to (2.14), it is equivalent to (2.12).

In practice, however, solving a sequence of MILPs (2.14) may not be efficient, because MILPs do not warm-start. Instead, we solve the LP-relaxation of the coarse model (2.14) and add violated constraints until all constraints in the LP-relaxation of the semi-coarse model (2.12) are satisfied. We use the identified constraints as the initial set of constraints for the MILP coarse model iterations. We summarize this procedure in Algorithm 1.

Proposition 3 *When Algorithm 1 terminates, the solution $(y^k, \bar{x}^k, \bar{v}^k, \bar{w}^k)$ of the coarse model (2.14) with the set of added constraints $\mathcal{A}_{\text{MILP}}^k$ is the solution of the semi-coarse model (2.12).*

Proof Since $(y^k, \bar{x}^k, \bar{v}^k, \bar{w}^k)$ minimizes the objective function of the semi-coarse model (2.12) and satisfies all constraints (2.12), it is the optimal solution of (2.12). □

We show in Sect. 4 that our approach in Algorithm 1 is advantageous; in particular, it significantly reduces the number of MILP re-solves, as opposed to the case without LP-relaxation as warm-start.

In Appendix 1, we show that if the problem data is periodic with period δ and if there exists no coupling between periods, then our proposed approach converges in a single iteration. This observation motivates the use of our approach on problems that are nearly periodic.

3 Application to cogeneration for buildings

In this section, we apply our two-level approach to the cogeneration problem for buildings. Our MILP model (7.1) is adapted from models for cogeneration in commercial

Phase I: LP Warm-start

Set counter $l \leftarrow 0$ and the set of additional constraints $\mathcal{A}_{\text{LP}}^l \leftarrow \emptyset$;
 Solve the LP-relaxation of the coarse model (2.14) to get solution $(y_{\text{LP}}^l, \bar{x}_{\text{LP}}^l, \bar{v}^l, \bar{w}^l)$;
while $(y_{\text{LP}}^l, \bar{x}_{\text{LP}}^l, \bar{v}^l, \bar{w}^l)$ is not feasible for the LP-relaxation of the semi-coarse MILP (2.12) **do**
 Find the set of constraints $\mathcal{C}_{\text{LP}}^l$ that are violated in the LP-relaxation of (2.12) and add them to
 $\mathcal{A}_{\text{LP}}^{l+1} \leftarrow \mathcal{A}_{\text{LP}}^l \cup \mathcal{C}_{\text{LP}}^l$;
 Solve the LP-relaxation of (2.14) with the set of added constraints $\mathcal{A}_{\text{LP}}^{l+1}$ to get
 $(y_{\text{LP}}^{l+1}, \bar{x}_{\text{LP}}^{l+1}, \bar{v}^{l+1}, \bar{w}^{l+1})$;
 Set $l \leftarrow l + 1$;

Phase II: MILP Iterates

Set counter $k \leftarrow 0$ and $\mathcal{A}_{\text{MILP}}^k \leftarrow \mathcal{A}_{\text{LP}}^l$;
 Solve the coarse MILP (2.14) with additional constraints $\mathcal{A}_{\text{MILP}}^k$ identified in **Phase I** to get solution
 $(y^k, \bar{x}^k, \bar{v}^k, \bar{w}^k)$;
while $(y^k, \bar{x}^k, \bar{v}^k, \bar{w}^k)$ is not feasible for (2.12) **do**
 Find the set of constraints $\mathcal{C}_{\text{MILP}}^k$ that are violated in (2.12) and add them to
 $\mathcal{A}_{\text{MILP}}^{k+1} \leftarrow \mathcal{A}_{\text{MILP}}^k \cup \mathcal{C}_{\text{MILP}}^k$;
 Solve the coarse MILP (2.14) with the added constraints $\mathcal{A}_{\text{MILP}}^{k+1}$ to get
 $(y^{k+1}, \bar{x}^{k+1}, \bar{v}^{k+1}, \bar{w}^{k+1})$;
 Set $k \leftarrow k + 1$;

Algorithm 1: Solve the semi-coarse MILP (2.12) via a sequence of coarse MILPs (2.14) using the LP-relaxation as warm-start.

buildings [28,30]. In particular, we take linearized models for fuel cells and water tank storage from [28], and we penalize on/off operations using switching cost as done in [30]. While the two-level framework described in Sect. 2 applies to generic MILPs (1.1), the MILP model (7.1) for cogeneration entails several complex constraints as discussed in Sect. 3.2. As a result, additional work is required to construct appropriate semi-coarse and coarse models.

We note that our MILP model (7.1) for the cogeneration problem has the following features that are not included in existing models. First, our model has a large number of binary variables, on the order of $\mathcal{O}(10^6)$, in the second stage. This is orders of magnitude larger than the number of binary variables in [28,30,33,34]. This modeling feature allows considerably more degrees of freedom for on/off operation during the life time of new technologies. Second, our model contains three sets of coupling constraints that (i) couple first- and second-stage binary variables, (ii) couple second-stage variables for different technologies, and (iii) couple second-stage variables over a long time horizon (e.g., 10–20 years). The number of coupling constraints is on the same order of variables, namely, $\mathcal{O}(10^6)$. This makes the problem significantly harder because it does not lend itself to decomposition techniques such as Lagrangian relaxation.

In what follows, we describe the main characteristics of the MILP model in Sect. 3.1, derive the semi-coarse and coarse models in Sects. 3.2 and 3.3, and discuss profile generation and selection in Sect. 3.4.

3.1 MILP model

The cogeneration problem consists of two components: the investment decision and the operation planning. The investment decision concerns what new technologies to purchase, while the operation planning concerns how to dispatch units over a long-term period (e.g., 10–20 years). We formulate a two-stage MILP, where the first-stage variables model investment decisions and the second-stage variables model equipment operations. Note that both the first and second stages contain integer variables. In particular, on/off operations for new technologies in second-stage are made on an hourly basis.

Following [28,30,33,34], we consider five technologies: batteries, boilers, solid-oxide fuel cells (SOFCs), combined heat and power (CHP) SOFCs, and water tank storage. The selection criterion is based on the installation, operation, maintenance, fuel consumption, and carbon emission costs, while meeting electricity and heating demands. The detailed MILP model is given in (7.1).

3.2 Semi-coarse model

We begin by introducing the profiles for variables. We denote profiles by upper-case letters with the bar notation on top: $\bar{X}_{jk} \in \mathcal{P}_o$ are the on/off profiles for technology j , $\bar{P}_{jkl} \in \mathcal{P}_p$ are the production profiles associated with \bar{X}_{jk} , $\bar{U}_k \in \mathcal{P}_u$ are the profiles for power purchased from the utility, $\bar{Q}_k \in \mathcal{P}_q$ are the heat generation profiles from boilers, $\bar{B}_k, \bar{B}_k^{IO} \in \mathcal{P}_b$ are the power storage and input/output profiles for batteries, and $\bar{S}_k, \bar{S}_k^{out} \in \mathcal{P}_s$ are the heat storage and output profiles for water tanks, where $\mathcal{P}_o, \mathcal{P}_p, \mathcal{P}_u, \mathcal{P}_q, \mathcal{P}_b,$ and \mathcal{P}_s are the corresponding set of profiles. We denote the h th element of a profile by $\bar{X}_{jk}(h)$ for $h \in \{1, \dots, \delta\}$. Time-dependent parameters D_t^P and D_t^Q and discount factors Y_t are concatenated into profiles $\bar{D}_d^P, \bar{D}_d^Q,$ and $\bar{Y}_d \in \mathbb{R}^\delta$.

As described in Sect. 2, the variables in the original MILP model are represented by using profiles in the coarsened models. In particular, binary variables x_{ijt} are coarsened to profiles \bar{X}_{jk} , and continuous variables $p_{ijt}, u_t, b_t, s_t, s_t^{out},$ and q_t are coarsened to profiles $\bar{P}_{jkl}, \bar{U}_k, \bar{B}_k, \bar{S}_k,$ and $\bar{S}_k^{out}, \bar{Q}_k \geq 0,$ respectively. The coefficients for profiles are denoted by lower-case letters with a bar on top; for example, $\bar{x}_{ijk} \in \{0, 1\}$ indicates whether profile k is selected on day d for unit i of technology j . As modeled in (9.1d), no more than one on/off profile can be selected for fixed $i, j,$ and d . This, in conjunction with binary elements of \bar{X}_{jk} , results in binary-valued x_{ijt} . Similarly, non-negative profiles $\bar{P}_{jkl}, \bar{U}_k, \bar{B}_k, \bar{S}_k, \bar{S}_k^{out},$ and $\bar{Q}_k \geq 0$ and their non-negative coefficients in (9.1g), (9.1q), and (9.1r) result in non-negative $p_{ijt}, u_t, b_t, s_t, s_t^{out},$ and q_t .

We next turn to constraints that *do not* couple variables in time, namely, (7.1c), (7.1e), (7.1f), (7.1h), (7.1k), (7.1l), (7.1o), and (7.1p). The profile representation for these constraints is straightforward; one simply substitutes the summation of profiles as shown in (9.1c), (9.1e), (9.1f), (9.1h), (9.1k), (9.1l), (9.1o), and (9.1p), respectively. Note that all inequalities in (9.1) are elementwise inequalities. We do not include the production constraint (7.1d) in the semi-coarse model. Instead, we require that the production profile \bar{P}_{jkl} associated with the on/off profile \bar{X}_{jk} satisfy

$$R_j^{\min} \bar{X}_{jk} \leq \bar{P}_{jkl} \leq R_j^{\max} \bar{X}_{jk}. \tag{3.1}$$

This approach is justified because (7.1d) follows by construction due to the convex combination of coefficients for production profiles in (9.1g). By an analogous argument, $s_t^{\text{out}} \leq s_t$ in (7.1o) is a direct consequence of the following condition imposed on profiles:

$$\bar{S}_k^{\text{out}} \leq \bar{S}_k. \tag{3.2}$$

We next discuss how to coarsen variables that are coupled over time periods. The boundary conditions (7.1j) and (7.1n) can be expressed as (9.1j) and (9.1n). The maximum power purchased constraint (7.1h) can be rewritten as (9.1h), where $\mathbf{1}$ denotes the vector of all ones with length δ . We now turn to switching and storage constraints (7.1g), (7.1i), and (7.1m), whose profile representation needs additional notation. Given an on/off profile $\bar{X}_{jk} \in \{0, 1\}^\delta$, we can construct the associated switching profile

$$\bar{W}_{jk}(h) = |\bar{X}_{jk}(h + 1) - \bar{X}_{jk}(h)|. \tag{3.3}$$

Then the switching cost can be included in the objective function in (9.1). To deal with the battery storage constraint (7.1i) that couples b_t and b_t^{IO} over the horizon T , we require that the pair of profiles $(\bar{B}_k, \bar{B}_k^{\text{IO}})$ satisfy

$$\bar{B}_k(h + 1) = (1 - L^P) \bar{B}_k(h) + \bar{B}_k^{\text{IO}}(h), \quad h = 1, \dots, \delta - 1. \tag{3.4}$$

We assign the same coefficient \bar{b}_{dk} to both sets of profiles \bar{B}_k and \bar{B}_k^{IO} throughout (9.1). It follows that (7.1i) is satisfied except for hours between profiles, namely, $t = d\delta$ for $d < |\mathcal{D}|$. Therefore, we introduce constraint (9.1i) to guarantee that (7.1i) holds for $t = d\delta$ for $d < |\mathcal{D}|$. We will show in Sect. 3.4 how to generate profiles that satisfy (3.1), (3.2), and (3.4). The heat storage constraint (7.1m) for $j = \text{chp}$ can be expressed as

$$\begin{aligned} & \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \left\{ [S_\delta - (1 - L^Q)I_\delta] \bar{S}_k + \bar{S}_k^{\text{out}} \right\} \\ & + \sum_{k \in \mathcal{P}_s} \bar{s}_{(d+1)k} E_\delta \bar{S}_k \leq (E_j^Q / E_j^P) \sum_{i,k,l} \bar{p}_{ijkl} \bar{P}_{jkl}, \quad d < |\mathcal{D}|, \end{aligned}$$

where $I_\delta \in \mathbb{R}^{\delta \times \delta}$ is the identity matrix, $E_{\delta 1} \in \mathbb{R}^{\delta \times \delta}$ is a matrix with 1 in the $(\delta, 1)$ entry, and $S_\delta \in \mathbb{R}^{\delta \times \delta}$ is a Toeplitz matrix with only nonzero elements being 1 at the first upper-subdiagonal.

3.3 Coarse model

We next turn to the coarse model. Note that constraints in the semi-coarse model (9.1) are elementwise equalities or inequalities involving daily profiles. We aggregate

hourly constraints by taking a weighted sum of the elements of daily profiles. Using the hat notation on the top to denote the elementwise summation of profiles, we coarsen δ hourly constraints into a single constraint. For example, the hourly demand constraint (9.1c) is replaced by the daily demand constraint (10.1c). Similarly, the maximum hourly power demand (9.1h) is aggregated into the maximum daily demand (10.1h). The symmetric breaking constraint (10.1f) has the interpretation that the number of times unit $i + 1$ is turned on is no greater than the number of times unit i is turned on in day d . The capacity constraints for technologies (10.1e), (10.1k), (10.1o), and (10.1p) imply that the daily power/heat output is bounded by the daily capacity of the technology units purchased at the first stage. Since (9.1i), (9.1j), and (9.1n) are scalar constraints themselves, no aggregation is applied to (10.1i), (10.1j), and (10.1n) in the coarse model. Also, since the coefficients of profiles are the same for both the semi-coarse and coarse models, (10.1d), (10.1g), and (10.1q) stay the same as (9.1d), (9.1g), and (9.1r) in the semi-coarse model.

3.4 Profile generation and selection

Recall that the on/off profiles \bar{X}_{jk} must have binary elements and the profiles \bar{P}_{jkl} , \bar{Q}_k , \bar{U}_k , \bar{B}_k , \bar{S}_k , and \bar{S}_k^{out} must have non-negative elements. In addition, we impose constraints (3.1), (3.2), and (3.4) in formulating the semi-coarse model in Sect. 3.2. In this section, we discuss how to generate valid profiles, and we provide suggestions for profile selections.

One approach to generating profiles that satisfy the above constraints is as follows. We solve a number of small instances of the original MILP (7.1), take snapshots of the second-stage solutions, and extract profiles from these snapshots. For example, consider a four-day MILP (7.1); that is, the time horizon in the second-stage problem has only four days. Solving such a four-day model, we have four snapshots of daily operation and production; in particular, we have four sets of on/off profiles for $x_{ijt} \in \{0, 1\}$ and four sets of profiles p_{ijt} , u_t , b_t , s_t , s_t^{out} , and $q_{jt} \geq 0$. We will show in Proposition 1 that these are valid profiles for the semi-coarse model (9.1).

The remaining question is how to choose the short-horizon MILPs. Our objective is to generate a rich set of profiles that are representative of the optimal solutions for a long-horizon MILP (7.1). In what follows, we develop a *moving-horizon approach*. Given a long-horizon MILP, the idea is to solve MILPs over a short window, roll the window forward, and re-solve the new MILP until the window reaches the end of the horizon; see Fig. 2 for an illustration. We summarize this approach and provide additional details in Algorithm 2.

Lemma 1 *The moving-horizon method described in Algorithm 2 generates non-negative profiles $\{\bar{P}_{jkl}, \bar{Q}_k, \bar{U}_k, \bar{B}_k, \bar{S}_k, \bar{S}_k^{\text{out}}\} \in \mathbb{R}_+^\delta$ and binary profiles $\{\bar{X}_{jk}, \bar{W}_{jk}\} \in \{0, 1\}^\delta$. Moreover, the profile pairs $\{\bar{X}_{jk}, \bar{P}_{jkl}\}$, $\{\bar{S}_k^{\text{out}}, \bar{S}_k\}$, and $\{\bar{B}_k, \bar{B}_k^{\text{IO}}\}$ satisfy (3.1), (3.2), and (3.4), respectively.*

Proof Non-negativity of the profiles $\{\bar{P}_{jkl}, \bar{Q}_k, \bar{U}_k, \bar{B}_k, \bar{S}_k, \bar{S}_k^{\text{out}}\}$ follows from the fact that the second-stage solutions $\{p_{ijt}, q_t, u_t, b_t, s_t, s_t^{\text{out}}\}$ of the original MILP model (7.1) are non-negative. Similarly, $\bar{X}_{jk} \in \{0, 1\}^\delta$ follows from $x_{ijt} \in \{0, 1\}$; and

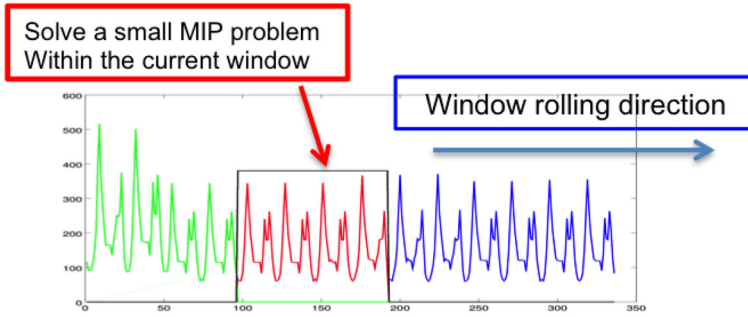


Fig. 2 Illustration of the moving horizon method in Algorithm 2

Data: The parameters for MILP (7.1) with a horizon of D days, and a window of w days with $w \ll D$.
Result: Profiles $\{\bar{P}_{jkl}, \bar{Q}_k, \bar{U}_k, \bar{B}_k, \bar{S}_k, \bar{S}_k^{\text{out}}\} \in \mathbb{R}_+^\delta$, $\bar{B}_k^{\text{IO}} \in \mathbb{R}^\delta$, and $\{\bar{X}_{jk}, \bar{W}_{jk}\} \in \{0, 1\}^\delta$.
 Set $k \leftarrow 0$, $\mathcal{W} = \{1, \dots, \delta w\}$, $\mathcal{R} = \{1, \dots, \delta\}$;
while $k + w \leq D$ **do**
 Solve MILP (7.1) in the current window $t \in \mathcal{W}$;
 Take snapshots of solutions $\{x_{ijt}, p_{ijt}, q_t, u_t, b_t, b_t^{\text{IO}}, s_t, s_t^{\text{out}}\}$ in $t \in \mathcal{R}$;
 Extract profiles $\{\bar{X}_{jk}, \bar{P}_{jkl}, \bar{Q}_k, \bar{U}_k, \bar{B}_k, \bar{B}_k^{\text{IO}}, \bar{S}_k, \bar{S}_k^{\text{out}}\}$, and $\bar{W}_{jk} \in \{0, 1\}^\delta$ using (3.3);
 Roll the window forward by setting $k \leftarrow k + 1$;
 if $k + w < D$ **then** Set $\mathcal{W} = \{\delta k + 1, \dots, \delta(k + w)\}$, $\mathcal{R} = \{\delta k + 1, \dots, \delta(k + 1)\}$;
 if $k + w = D$ **then** Set $\mathcal{W} = \mathcal{R} = \{\delta k + 1, \dots, \delta H\}$.

Algorithm 2: Moving horizon method for profile generation.

\bar{W}_{jk} , constructed from (3.3), is elementwise binary. Since the production variable p_{ijt} and the on/off variable x_{ijt} satisfy (7.1d), it follows that $\{\bar{X}_{jk}, \bar{P}_{jkl}\}$ satisfies (3.1). Since the power storage b_t and power input/output b_t^{IO} satisfy (7.1i) and since the heat storage s_t and heat output s_t^{out} satisfy (7.1m), we conclude that (3.2) and (3.4) follow by construction. \square

Algorithm 2 potentially generates a huge number of profiles; hence, solving (9.1) and (10.1) with all generated profiles may be prohibitive. Instead, we select on/off profiles \bar{X}_{jk} that appear most frequently in the generated profile pool. Since the aim of the two-level approach is to reduce the problem size, it is desired that the number of on/off profiles k is much smaller than the length of profiles δ (e.g., $k \approx \delta/10$). For the production profiles \bar{P}_{jkl} associated with each \bar{X}_{jk} , we choose production profiles that have the minimum or maximum total production $\sum_{h=1}^\delta \bar{P}_{jkl}(h)$. These extreme profiles provide an envelope of other profiles; thus, their convex combination (9.1g) provides a good range of profiles for selection. Similarly, profiles with the minimum and maximum sum of absolute values are chosen for the battery storage \bar{B}_k , the heat storage \bar{S}_k , the battery input/output \bar{B}_k^{IO} , and the heat output \bar{S}_k^{out} . Further, profiles for the power purchased from the utility \bar{U}_k and the heat output from the boiler \bar{Q}_k are uniformly sampled over periods of the entire horizon.

Alternatively, we also employ a *k-means clustering algorithm* in order to cluster profiles. Given a prespecified k number of clusters, this algorithm assigns profiles to one of k clusters defined by the centroids [24]. Since demand and pricing data for buildings tend to differ significantly in winters and summers, we set $k = 2$ and choose one profile that has the minimum least-squares distance to the centroids. In our numerical experiments in Sect. 4, we apply a k -means algorithm to the boiler output \bar{Q}_k , power purchased from the utility \bar{U}_k , heat storage output \bar{S}_k^{out} , and battery power output \bar{B}_k^{IO} profiles. The clustering approach achieves better performance in the objective function than does simple sampling heuristics for profile selection.

4 Numerical results and case studies

In this section, we illustrate the effectiveness of our two-level approach using five different building examples. We demonstrate that both the semi-coarse and the coarse models allow us to find good approximate solutions in a fraction of the time compared with the full MILP model. The two-level approach also scales to large problems that are beyond the scope of state-of-the-art commercial MILP solvers.

4.1 Generation of problem instances

We use *EnergyPlus 8.4* [8, 9] to generate yearly electricity and heating demands for five types of buildings, namely, a secondary school, a supermarket, a hospital, a stand-alone retail, and a full-service restaurant, located in Chicago, Illinois. Figure 3 shows the electricity demand of the secondary school and the restaurant in a year and in November. Note that the demand data shows cyclic structure in days, weeks, and months, which is a desired problem characteristic for our two-level approach as discussed in Appendix 1. To generate multiyear demands, we take 1-year demand and perturb with a zero mean unit variance Gaussian noise whose magnitude is 2 % proportional to the magnitude of demands. We follow the pricing structure in [34]: The electricity price is \$0.12 per kWh and the gas price is \$0.049 per kWh. The peak demand charge is \$14.2 per kW for summer months (from June to September) and \$11.36 per kW for winter months (from October to May). For multiyear pricing, we follow the history data from U.S. Energy Information Administration [12] and increase the electricity price 3 % annually.

4.2 Numerical experiment setup

Numerical experiments are performed on a workstation with 32 GB memory and two Intel E5430 Xeon 4-core 2.66 GHz CPUs. We implement our algorithms in AMPL [15] to take advantage of AMPL's compact modeling syntax in profile generation, storage, and management. We use CPLEX version 12.6.1.0 as the MIP solver in AMPL. We set a 3-h time limit and 1 % relative gap as the stopping criteria for CPLEX. Our AMPL codes are made publicly available at <http://www.mcs.anl.gov/~fulin/codes/DistrGen.zip>.

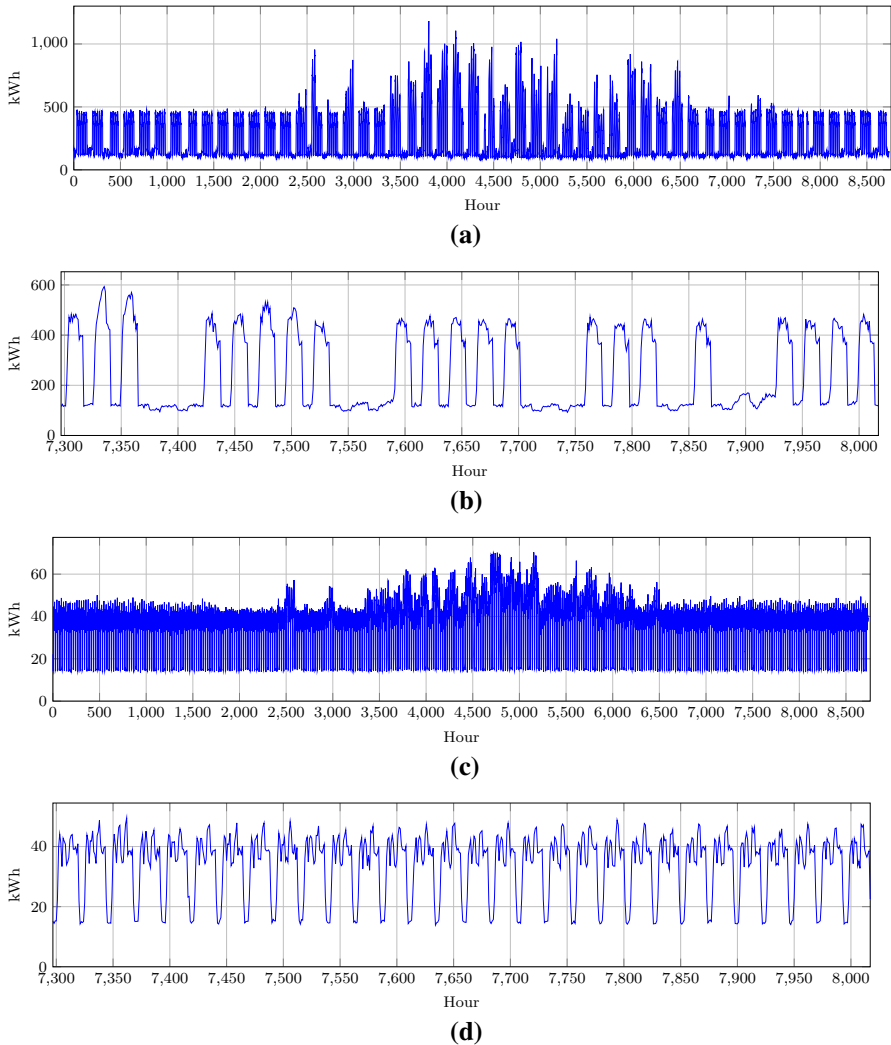


Fig. 3 Electricity demand generated by EnergyPlus for a secondary school and a full-service restaurant located in Chicago. **a** Electricity demand for a secondary school in Chicago in a year. **b** Electricity demand for a secondary school in Chicago in November. The school is closed on Thanksgiving Day. **c** Electricity demand for a full-service restaurant in Chicago in a year. **d** Electricity demand for a full-service restaurant in Chicago in November

4.3 Solutions of the full model with short second-stage horizons

Table 1 shows the problem size of the full MILP model (7.1) as a function of the number of days in the second stage. The number of binary variables, continuous variables, and constraints grows linearly with the number of days in the second-stage problem. The one-year model has 1.05×10^5 binary variables, 2.71×10^5 continuous variables, and 6.11×10^5 constraints.

Table 1 Problem size of the MILP model (7.1). The number of binary variables, continuous variables, and constraints grows linearly with the number of days

Days	Binary	Continuous	Constraint
4	1152	2994	6698
7	2016	5226	11,738
14	4032	10,434	23,498
28	8064	20,850	47,018
84	24,192	62,514	1.41×10^5
364	1.05×10^5	2.71×10^5	6.11×10^5

Table 2 Solutions of the MILP model (7.1) for Restaurant using CPLEX

Days	Time (s)	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor
4	5.81e+01	1050	1.30e+05	2	1	1	1	1
7	1.28e+02	2307	2.41e+05	2	1	1	1	1
14	3.21e+02	3883	4.92e+05	2	1	1	1	2
28	3.50e+03	35836	2.22e+06	2	1	1	1	1
84 ^a	1.06e+04	13252	3.99e+06	3	1	1	1	1

^a The 84-day model reaches the 3-h limit

Table 2 shows the solutions of small problems for Restaurant using CPLEX. The solution time in seconds, the number of nodes, and the total number of simplex iterations for the branch-and-cut algorithm grow exponentially with the number of days in the second stage. The first-stage solutions for batteries (Bat), boilers (Boil), CHP-SOFC (Chp), Power SOFC (Pow), and water tank storage (Stor) vary with the problem size. Solving the 84-day example is beyond the capabilities of CPLEX on the designated workstation. After reaching the 3-h time limit, the relative MIP gap for the 84-day model is still greater than 2.49%. Removing the time limit does not help because then the branch-and-bound tree generated by CPLEX will consume all allowable disk space of 100 GB on the workstation. Our experience indicates that the full MILP model (7.1) probably can not be solved over a ten-year time horizon. Similar observations can be made for other building examples. Figure 4 shows that for all five buildings, both the number of simplex iterations and the amount of solution time increase exponentially with the number of days in the second stage.

We point out that different buildings show different levels of difficulty for the MILP model (7.1). As we see in Fig. 4, the five buildings differ by orders of magnitude in solution time and the number of simplex iterations. Similarly, the number of nodes in the branch-and-bound trees varies significantly over different buildings. For example, the 28-day model requires 35,836 nodes for Restaurant as opposed to 360 nodes for School. Additional numerical results are summarized in Table 7 in Appendix 3.

4.4 Solutions of semi-coarse model with long second-stage horizons

Figure 5 compares the problem size for the full model (7.1), the semi-coarse model (9.1), and the coarse model (10.1) over a 10-year horizon in the second stage.

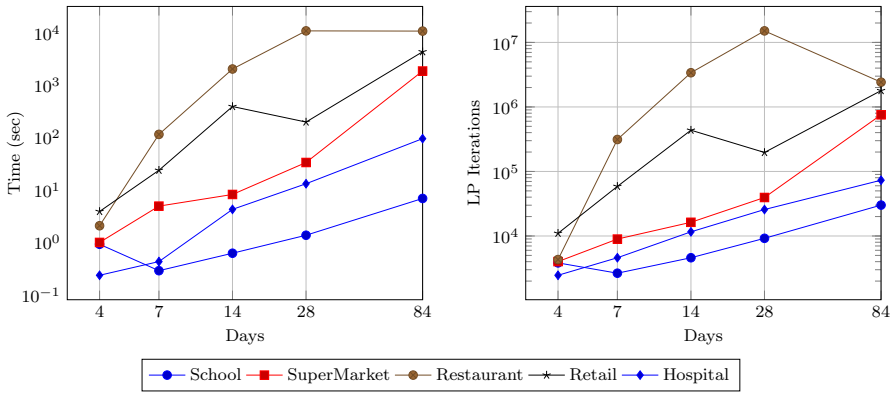


Fig. 4 Exponential increase of the amount of time and the number of simplex iterations with the number of days in the second stage of the full MILP model (7.1). The number of simplex iterations drops for the 84-day *restaurant* example because CPLEX reaches the 3-h time limit

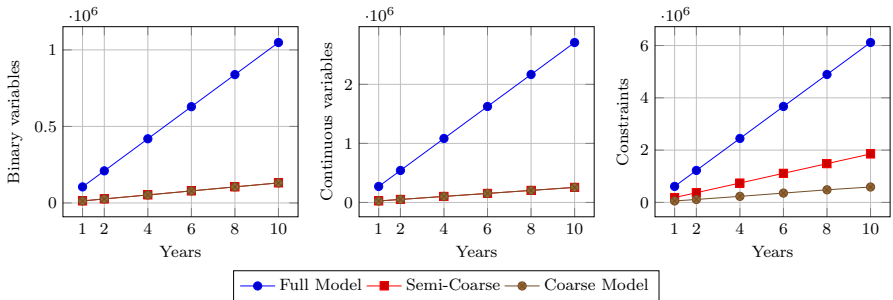


Fig. 5 Problem size for the full MILP model (7.1), the semi-coarse model (9.1), and the coarse model (10.1)

We see a roughly 8 times reduction in the number of binary variables and 12 times reduction in the number of continuous variables. The reason is that for each day with 24 h, we pick the three most frequent on/off profiles for binary variables and two cluster centers from the *k*-means clustering algorithm for continuous variables. While increasing the number of profiles improves the quality of the coarsened models (9.1) and (10.1), the resulting computational effort increases significantly. For 10-year horizon problems, we find that the two-level approach strikes a good balance between solution quality and computational time with a small number of profiles (typically 2 - 3). Note the large reduction in the number of constraints from the full model to the semi-coarse model. This is mainly because we have embedded the min/max production constraints (7.1d) and the switching constraints (7.1g) in the profile generation; see Sect. 3.2. In particular, for a 10-year model with 12 fuel cell units, there is a reduction of $12 \times 87600 \times 4 \approx 4.2 \times 10^6$ constraints (i.e., a 67 % reduction in the number of constraints.)

Table 3 shows the solution information for the semi-coarse model over a 10-year horizon. As opposed to the exponential increase for the full model, the solution time, the number of nodes, and the number of simplex iterations for the semi-coarse model grow more slowly.

Table 3 Solution of the semi-coarse model (9.1) for restaurant

Days	Time (s)	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor
364	134	487	62,200	0	1	1	1	1
728	404	390	1.16×10^5	0	1	1	1	1
1456	1560	508	1.66×10^5	0	1	1	2	1
2184	9510	3015	1.29×10^5	0	1	3	1	1
2912	8090	727	5.71×10^5	0	1	1	1	1
3640	9950	515	4.6×10^5	0	1	0	0	1

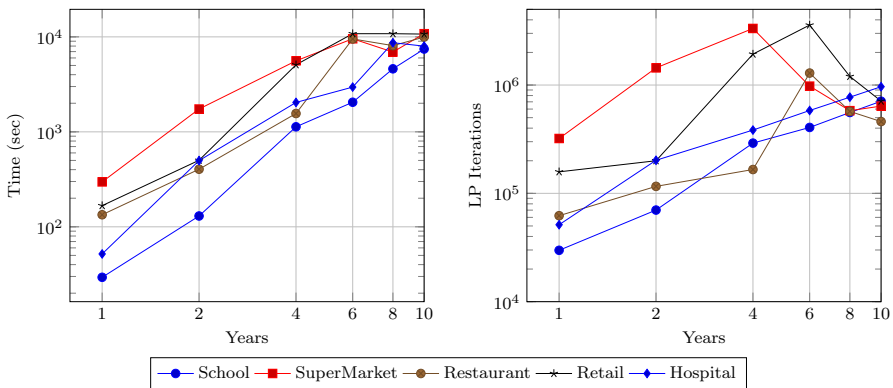


Fig. 6 Solution time and number of simplex iterations for the semi-coarse model over a 10-year horizon

Figure 6 shows that both the number of simplex iterations and the amount of solution time are more scalable for the semi-coarse models. For example, the number of simplex iterations for 10-year semi-coarse models is on the same order as that of the 84-day full model. The growth of computational effort is considerably slower than that of the full model shown in Fig. 4. The number of simplex iterations drops for 6-, 8-, and 10-year problems for SuperMarket and 8- and 10-year problems for Retail and Restaurant because CPLEX reaches the 3-h time limit. The numerical results for the semi-coarse model are summarized in Table 8 in Appendix 3.

4.5 Solutions of the coarse model with long second-stage horizons

As shown in Fig. 5, the semi-coarse and coarse models have the same number of binary and continuous variables, and the semi-coarse model has about twice as many constraints as the coarse model. Therefore, one expects that both models can be solved with a similar amount of computational effort in terms of time and simplex iterations. Since we solve a sequence of coarse models, the total computational effort often exceeds that of solving the semi-coarse model. Because of the LP warm-start phase in Algorithm 1, however, we find that the first iterate of the coarse MILPs provides remarkably good approximate solutions of the semi-coarse model.

Table 4 Problem size of MILP iterates for a 6-year model of Restaurant. The first iterate accounts for 91.5 % of computational time, and the relative gap between the objective value defined in (4.1) is less than 0.72 %

Iter	Binary	Continuous	Constraint	Objective value				
1	78,624	179,171	314,999	7,498,287.39				
2	78,624	179,171	315,563	7,552,429.85				
3	78,624	179,171	315,654	7,552,430.06				

Iter	Time (s)	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor
1	6,112.33	7,492	1.06×10^6	0	1	1	1	1
2	283.44	0	1.12×10^5	0	1	1	1	1
3	284.89	0	1.16×10^5	0	1	1	1	1

To illustrate this point, we show in Table 4 the solution history of the MILP phase in Algorithm 1. The MILP phase converges in three coarse MILP iterates, and the first iterate requires much more computational effort than other iterates; in particular, it accounts for 91.5 % of total computational time. The relative gap of the objective value

$$\mu = (\text{ObjVal}_{\text{semi}} - \text{ObjVal}_{\text{coar}}) / \text{ObjVal}_{\text{semi}} \quad (4.1)$$

between the semi-coarse model and the coarse model at the first iterate is less than 0.72 %. Numerical results for the first iterate of coarse MILPs are summarized in Table 9 in Appendix 3.

4.6 Effect of the LP warm-start

Figure 7 shows the number of constraints and the computational time during the LP warm-start phase. A large number of constraints are added in the first few (3–5) iterates, resulting in more computation time at the beginning of the algorithm. As the LP warm-start phase progresses, fewer constraints are violated, and each LP-relaxation becomes easier to solve because of the warm-start. The amount of time in the LP warm-start phase is a fraction of that with one MILP re-solve in the MILP phase.

We next compare the MILP phase of Algorithm 1 with and without the LP warm-start. Figure 8 shows the number of MILP re-solves and the computation time for both cases. With the LP warm-start, Algorithm 1 takes 5 MILP re-solves, as opposed to 15 MILP re-solves without the LP warm-start. Moreover, the MILP re-solves with LP warm-start is up to 20 times faster than MILPs without the LP warm-start. We note that the improvement of the LP warm-start phase is observed in all building examples.

4.7 Solution quality of the semi-coarse model

The solution quality of the semi-coarse model is determined by the number of profiles for binary and continuous variables in the primal coarsening step (2.13). As

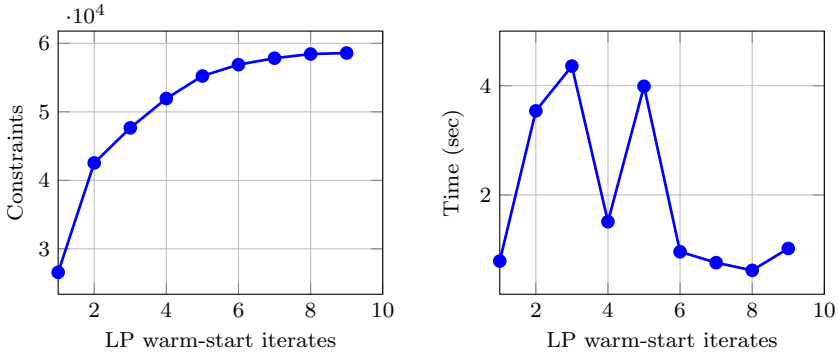


Fig. 7 Number of constraints (*left panel*) and the solution time (*right panel*) in the LP warm-start phase for the one-year retail example

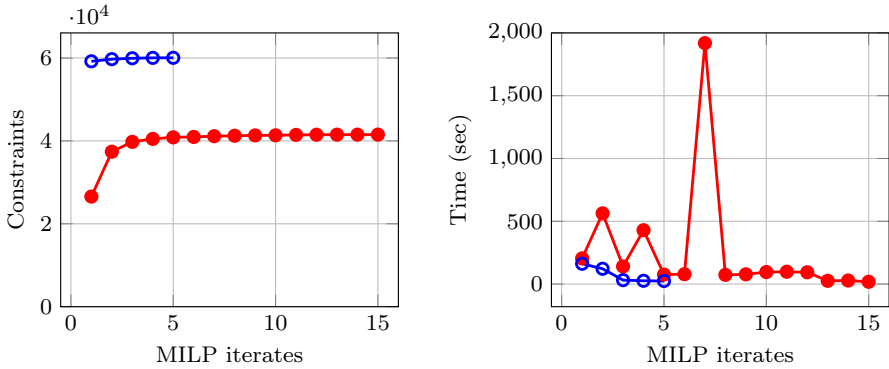


Fig. 8 Effect of LP warm-start for the one-year retail example. Algorithm 1 with LP warm-start phase (*open circle*) converges with fewer MILP iterations than without LP warm-start (*filled circle*). While more constraints are added to MILPs with LP warm-start (*left panel*), each MILP iterate is up to 20 times faster than MILPs without LP warm-start (*right panel*)

discussed in Sect. 2.1, a larger number of profiles imply a less tightening semi-coarse model. Table 5 shows the problem size and the objective value of the semi-coarse model for School with the 84-day horizon. As the number of on/off profiles, K , grows, the objective value decreases monotonically. Here, the number of profiles for the continuous variables is set to be four for each on/off profile; that is, $I_k = 4$ for $k = 1, \dots, K$ in (2.4), and $J = 4K$ in (2.5). When $K = 20$, the relative difference between the objective value of the semi-coarse model and the full model is $(\text{ObjVal}_{\text{semi}} - \text{ObjVal}_{\text{full}})/\text{ObjVal}_{\text{full}} = 18\%$. Table 10 in Appendix 3 shows a more detailed comparison for five buildings between the full model and the semi-coarse model with $K = 5$ for 28, 56, and 84 days.¹

We next examine the correlation between the solution quality of the semi-coarse model and the periodicity of the demand data. We perform the discrete Fourier trans-

¹ For the full model, the longest horizon that can be solved is 84 days.

Table 5 Solution quality and problem size of the semi-coarse model versus the number of on/off profiles K for School with 84-day horizon

K	Binary	Continuous	Constraints	ObjVal	Ratio
3	3024	9258	42,758	3.85×10^7	1.77
5	5040	15,978	44,774	3.71×10^7	1.71
10	8820	30,258	48,554	3.57×10^7	1.64
15	11,760	42,858	51,494	3.48×10^7	1.6
20	12,936	51,930	52,670	2.57×10^7	1.18
Full	24,192	62,514	1.41×10^5	2.18×10^7	1

The ratio between the objective value of the semi-coarse model and the full model is shown in the last column. The last row shows the result for the full model

form of the yearly electricity demand, $d_{\omega+1} = \sum_{t=0}^{N-1} e^{-\frac{2\pi\mathbf{j}t\omega}{N}} D_{t+1}$, where $\mathbf{j} := \sqrt{-1}$ and $\omega = 0, \dots, N - 1$ with $N = 24 \times 364 = 8736$. Since the semi-coarse model employs daily profiles with $\delta = 24$, we are interested in those frequencies that correspond to periods with integer multiples of δ in time. Let $\Omega := \{\omega \in \{0, \dots, N - 1\} \mid \text{mod}(\bar{\omega}, \omega) = 0\}$, where $\bar{\omega} = 364$ is the frequency corresponding to daily profiles with $\delta = 24$, or specifically $\Omega = \{0, 1, 2, \dots, 182, 364\}$. We compute the spectral density at desired frequencies Ω divided by the total spectral density:

$$\phi := \left(\sum_{\omega \in \Omega} |d_{\omega}|^2 \right) \left(\sum_{\omega=0}^{N-1} |d_{\omega}|^2 \right)^{-1}. \tag{4.2}$$

Thus, a higher density ratio ϕ implies a higher degree of data periodicity with respect to daily profiles.

Table 6 shows the spectral density ratio of buildings in a descending order. The Hospital has the highest density ratio, 0.99011, trailed by SuperMarket, Restaurant, School, and Retail with the lowest ratio, 0.96787. Table 6 also shows the objective value ratio between the full model and the semi-coarse model for 364 days. Here, we solve the full model with fixed first-stage solutions obtained from the semi-coarse model. The ranking of the density ratio correlates well with the solution quality of the semi-coarse model; in particular, a higher density ratio corresponds to a larger objective value ratio between the full model and the semi-coarse model. While we do not take into account coupling between time periods, numerical results in Table 6 suggest that periodic demand data with periods being integer multiple of δ are a desired problem characteristic for our two-level approach.

4.8 First-stage solutions

Since the first-stage solutions are of primary importance, we next compare them from the full MILP, the semi-coarse, and the coarse models. In Table 7, we see that the investment decisions at the first stage vary significantly with respect to the horizon

Table 6 Correlation between spectral density ratio ϕ in (4.2) of yearly demands and the solution quality of semi-coarse models

Building	Density ratio	ObjVal _{full} /ObjVal _{semi}
Hospital	0.99011	0.90334
SuperMarket	0.98985	0.88315
Restaurant	0.97579	0.65701
School	0.97108	0.63695
Retail	0.96787	0.56018

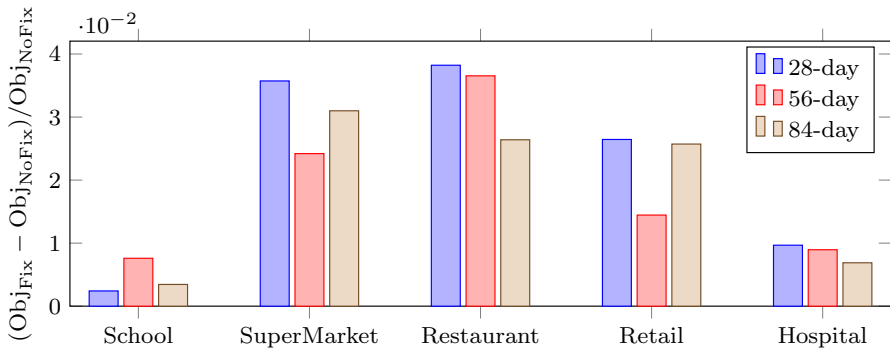


Fig. 9 Relative difference between the objective value of the full model with and without fixing the first-stage solutions from the semi-coarse model

length of the second-stage problem. (For the School example, the number of boilers jumps from 1 unit in the 4-day model to 8 units in the 28-day model, and the number of batteries drops from 4 units in the 28-day model to 1 unit in the 4-day model.) Thus, the first-stage solutions from a short-horizon problem are suboptimal for a long-horizon problem. This implies that one must solve long-horizon MILPs (7.1) as done for the semi-coarse and coarse models. In Table 8, the first-stage solutions are less sensitive to the horizon length at the second stage. Similarly, the first-stage solutions at the first iterate of the coarse MILPs are less sensitive to the horizon length; see Table 9.

To evaluate the solution quality, we re-solve the full model by fixing the first-stage solutions from the semi-coarse model. Figure 9 shows the relative difference of the objective value (i.e., the total cost from both the first and second stages) between the full model with and without fixing the first-stage solutions for five buildings with 28, 56, and 84 days in the second stage.² Note that the quality of first-stage solutions from the semi-coarse model varies with the building type. For SuperMarket and Restaurant, the relative difference of the objective value, $\nu = (\text{ObjVal}_{\text{Fix}} - \text{ObjVal}_{\text{NoFix}}) / \text{ObjVal}_{\text{NoFix}}$, is $\nu \leq 3.8\%$, while $\nu \leq 2.7\%$ for Retail and $\nu \leq 1\%$ for School and Hospital. In other words, the first-stage solutions from the semi-coarse model is within 1% suboptimal for School and Hospital and is within 3.8% suboptimal for SuperMarket and Restaurant.

The difference of the solution quality between buildings can be understood by examining the number of cogeneration units and other numerical results summarized

² The longest horizon that the full model can be solved for without fixing the first-stage solution is 84 days.

in Table 11. Note that the number of cogeneration units for Restaurant is much smaller than that for School; in particular, 3 battery units, 1 boiler unit, 1 CHP-SOFC unit, 1 Power-SOFC unit, and 1 storage unit for Restaurant, as opposed to 3 battery units, 8 boiler units, 5 CHP-SOFC units, 7 Power-SOFC units, and 8 storage units for School. The reason is that the electricity and heating demands of Restaurant, on average, are 13 % of those of School; see Fig. 3. Therefore, a small deviation in the first-stage solution for Restaurant has a bigger impact, and hence a larger ν , than for School.

5 Conclusions

We study two-stage MILPs with more than 1 million binary variables in the second-stage problem. We develop a two-level approach by constructing semi-coarse models (coarsened with respect to variables) and coarse models (coarsened in both variables and constraints). We show that the semi-coarse model is guaranteed to provide a feasible solution of the original MILP and hence results in an upper bound on the optimal solution. We solve a sequence of coarse MILPs with aggregated constraints that converges to the same upper bound with a finite number of steps. Furthermore, we take advantage of the LP warm-start to reduce the number of MILP re-solves. Under the assumption of periodic problem data with no coupling between time periods, we show that the semi-coarse model provides the optimal solution to the original fine model. Under the additional assumption of a full-rank non-negative weight matrix in the constraint aggregation, we show that the solution of the coarse model is feasible to the semi-coarse model.

We apply our approach to the cogeneration problem for commercial buildings. We demonstrate the effectiveness of the two-level approach using building examples with simulation data. In particular, the two-level approach allows us to obtain good approximate solutions at a fraction of the time required for solving the original MILPs. Furthermore, we show that the two-level approach scales to large problems that are beyond the capacity of state-of-the-art commercial solvers.

A number of extensions are of interest in our future work. First, how should one add new profiles in the coarsened models? Currently, we select a number of prespecified profiles and fix them throughout the MILP re-solves. Since the solution quality of the coarsened models is determined by the number of profiles, it is of interest to add new profiles dynamically that are most promising in reducing the objective value. It seems that the reduced cost associated with profiles can be obtained by solving appropriate pricing problems like those in the column generation approach [4]. Further, it is also of interest to consider dynamic aggregations of constraints as new profiles are included. For set-partitioning constraints, a similar approach has been developed in the column generation framework [11]. We plan to explore this line of research.

Second, how can one obtain lower bounds on the optimal solution in the two-level framework? Since the coarse model is a relaxation of the semi-coarse model, which itself is a tightening of the full model, the coarse model is not a relaxation of the original MILP. While relaxing binary variables in the original MILP results in a lower bound, our numerical results indicate a sizable gap between this lower bound and the optimal

solution. It is an open problem how to generate lower bounds using appropriate coarse MILPs.

Furthermore, advanced multilevel approaches for PDEs typically require several sweeps of fine/coarse levels in order to achieve accurate solutions [13]. It may be beneficial to iterate between solving coarse models to optimality and solving “partially” the fine model until a prespecified accuracy for the solution is achieved. We intend to extend our two-level framework in this direction.

Acknowledgments This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract number DE-AC02-06CH11357. We thank the reviewers for their helpful comments. Fu Lin thanks Dr. Ralph Muehleisen for useful discussions on EnergyPlus.

Appendix 1: Periodic solutions

In this appendix, we show that the semi-coarse model is equivalent to the fine model under the idealistic assumption that the problem data is periodic and there is no coupling between time periods. Moreover, we show that the coarse model is equivalent to the semi-coarse model if the non-negative weights in the convex combination of constraints form a full rank matrix. This equivalence motivates the use of our algorithm on problems that have “nearly” periodic data.

Let us consider periodic coefficients in the objective function and constraints; that is, we can divide the second-stage data into $n = N/\delta$ identical parts,

$$b = \begin{pmatrix} \mathbf{b} \\ \vdots \\ \mathbf{b} \end{pmatrix}, \quad c = \begin{pmatrix} \mathbf{c} \\ \vdots \\ \mathbf{c} \end{pmatrix}, \quad d = \begin{pmatrix} \mathbf{d} \\ \vdots \\ \mathbf{d} \end{pmatrix}, \quad f = \begin{pmatrix} \mathbf{f} \\ \vdots \\ \mathbf{f} \end{pmatrix}, \quad A = \begin{pmatrix} \mathbf{A} \\ \vdots \\ \mathbf{A} \end{pmatrix}, \quad (6.1)$$

where $\mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^\delta$, $\mathbf{f} \in \mathbb{R}^\gamma$, and $\mathbf{A} \in \mathbb{R}^{\gamma \times m}$. Similarly, we assume (without loss of generality) that the first-stage decisions impact all n time periods identically. We also assume that the second-stage matrices B, C , and D are block diagonal:

$$B = \begin{bmatrix} \mathbf{B} & & \\ & \ddots & \\ & & \mathbf{B} \end{bmatrix}, \quad C = \begin{bmatrix} \mathbf{C} & & \\ & \ddots & \\ & & \mathbf{C} \end{bmatrix}, \quad D = \begin{bmatrix} \mathbf{D} & & \\ & \ddots & \\ & & \mathbf{D} \end{bmatrix}, \quad (6.2)$$

where $\mathbf{B}, \mathbf{C}, \mathbf{D} \in \mathbb{R}^{\gamma \times \delta}$ with $N = n\delta$ and $M = n\gamma$.

Lemma 2 *Under the periodic data assumption (6.1)–(6.2), there exists an optimal solution to (1.1) that satisfies*

$$\mathbf{x}^* = \mathbf{x}_i^*, \quad \mathbf{v}^* = \mathbf{v}_i^*, \quad \mathbf{w}^* = \mathbf{w}_i^*, \quad \text{for } i = 1, \dots, n. \quad (6.3)$$

Proof Suppose that (x^*, v^*, w^*) is an optimal solution that does not satisfy (6.3). That is, there exist $(\mathbf{x}_q^*, \mathbf{v}_q^*, \mathbf{w}_q^*) \neq (\mathbf{x}_p^*, \mathbf{v}_p^*, \mathbf{w}_p^*)$ such that $\mathbf{b}^T \mathbf{x}_q^* + \mathbf{c}^T \mathbf{v}_q^* + \mathbf{d}^T \mathbf{w}_q^* \leq$

$\mathbf{b}^T \mathbf{x}_p^* + \mathbf{c}^T \mathbf{v}_p^* + \mathbf{d}^T \mathbf{w}_p^*$ and $\mathbf{A}y + \mathbf{B}\mathbf{x}_i + \mathbf{C}\mathbf{v}_i + \mathbf{D}\mathbf{w}_i \leq \mathbf{f}$ for $i = p, q$. Then one can construct another solution by replacing the set of variables $(\mathbf{x}_p^*, \mathbf{v}_p^*, \mathbf{w}_p^*)$ with $(\mathbf{x}_q^*, \mathbf{v}_q^*, \mathbf{w}_q^*)$, and repeat this process until the periodic solution (6.3) is obtained. \square

From Lemma 2, it follows that the periodic version of (1.1) can be equivalently written as

$$\begin{aligned}
 & \underset{\mathbf{v}, \mathbf{w}, \mathbf{x}, y}{\text{minimize}} && a^T y + n(\mathbf{b}^T \mathbf{x} + \mathbf{c}^T \mathbf{v} + \mathbf{d}^T \mathbf{w}) \\
 & \text{subject to} && \mathbf{A}y + \mathbf{B}\mathbf{x} + \mathbf{C}\mathbf{v} + \mathbf{D}\mathbf{w} \leq \mathbf{f} \\
 & && y \in \mathbb{Z}_+^m, \mathbf{x} \in \{0, 1\}^\delta \\
 & && \mathbf{v} \in \mathbb{R}^\delta, L\mathbf{x} \leq \mathbf{v} \leq U\mathbf{x} \\
 & && \mathbf{w} \in \mathbb{R}^\delta, \mathbf{w} \geq 0.
 \end{aligned} \tag{6.4}$$

In other words, it suffices to solve the periodic version of (1.1) with the single period problem (6.4) under condition (6.1)–(6.2), that is, if the data is periodic and if there is no coupling between time periods.

In our algorithm, we obtain the δ -profiles in (2.9) for $\bar{X}, \bar{V}, \bar{W}$ by solving the following short-horizon problem:

$$\begin{aligned}
 & \underset{\mathbf{v}_i, \mathbf{w}_i, \mathbf{x}_i, y}{\text{minimize}} && a^T y + \frac{n}{h} \left(\sum_{i \in \mathcal{D}} (\mathbf{b}^T \mathbf{x}_i + \mathbf{c}^T \mathbf{v}_i + \mathbf{d}^T \mathbf{w}_i) \right) \\
 & \text{subject to} && \mathbf{A}y + \mathbf{B}\mathbf{x}_i + \mathbf{C}\mathbf{v}_i + \mathbf{D}\mathbf{w}_i \leq \mathbf{f}, \quad i \in \mathcal{D} \\
 & && y \in \mathbb{Z}_+^m, \mathbf{x}_i \in \{0, 1\}^\delta, \quad i \in \mathcal{D} \\
 & && \mathbf{v}_i \in \mathbb{R}^\delta, L\mathbf{x}_i \leq \mathbf{v}_i \leq U\mathbf{x}_i, \quad i \in \mathcal{D} \\
 & && \mathbf{w}_i \in \mathbb{R}^\delta, \mathbf{w}_i \geq 0, \quad i \in \mathcal{D},
 \end{aligned} \tag{6.5}$$

where \mathcal{D} is a small set of periods and h is the number of periods in \mathcal{D} . (A more detailed description of the moving-horizon approach is provided in Sect. 3.4.) Similar to Lemma 2, we can show that the solution of (6.5) is equivalent to (6.4), independent of the choice of \mathcal{D} .

We do not assume uniqueness of the periodic solution. However, since (6.4) and (6.5) are equivalent, we can assume that any solver returns the same solution no matter how we choose \mathcal{D} . Hence, we will assume without loss of generality that a single periodic profile, $(\bar{X}, \bar{V}, \bar{W})$, is used for the semi-coarse model. By construction, we have

$$\bar{X} \in \{0, 1\}^\delta, \quad L\bar{X} \leq \bar{V} \leq U\bar{X}, \quad \bar{W} \geq 0, \quad \text{and} \quad \mathbf{A}y + \mathbf{B}\bar{X} + \mathbf{C}\bar{V} + \mathbf{D}\bar{W} \leq \mathbf{f}. \tag{6.6}$$

Substituting $\mathbf{x}_i = x_i \bar{X}$, $\mathbf{v}_i = v_i \bar{V}$, and $\mathbf{w}_i = w_i \bar{W}$ into the periodic version of (1.1) yields the semi-coarse model

$$\begin{aligned} & \underset{v, w, x, y}{\text{minimize}} && a^T y + \sum_{i=1}^n \left(x_i \mathbf{b}^T \bar{X} + v_i \mathbf{c}^T \bar{V} + w_i \mathbf{d}^T \bar{W} \right) \\ & \text{subject to} && \mathbf{A}y + (x_i \mathbf{B} \bar{X} + v_i \mathbf{C} \bar{V} + w_i \mathbf{D} \bar{W}) \leq \mathbf{f}, \quad i = 1, \dots, n \\ & && y \in \mathbb{Z}_+^m, \quad x_i \in \{0, 1\}, \quad v_i = x_i, \quad w_i \in [0, 1], \quad i = 1, \dots, n. \end{aligned} \tag{6.7}$$

Proposition 4 *The solution of the semi-coarse model (6.7) is the optimal solution of (6.4) under the periodic data assumption (6.1)–(6.2).*

Proof Since $(\bar{X}, \bar{V}, \bar{W})$ is an optimal profile, that is, the periodic solution of the short-horizon problem (6.5), then setting $x_i = v_i = w_i = 1$ in (6.7) in conjunction with (6.6) reduces the semi-coarse model to the short-horizon problem (6.5) with the optimal profile $(\bar{X}, \bar{V}, \bar{W})$, which is equivalent to (6.4).

We next consider a convex combination of (row) constraints

$$\mathbf{A}y + x_i \mathbf{B} \bar{X} + v_i \mathbf{C} \bar{V} + w_i \mathbf{D} \bar{W} \leq \mathbf{f}, \quad i = 1, \dots, n \tag{6.8}$$

to define the coarse model constraints

$$\lambda_i^T \mathbf{A}y + x_i \lambda_i^T \mathbf{B} \bar{X} + v_i \lambda_i^T \mathbf{C} \bar{V} + w_i \lambda_i^T \mathbf{D} \bar{W} \leq \lambda_i^T \mathbf{f}, \quad i = 1, \dots, n \tag{6.9}$$

where the non-negative weights $\lambda_i \in \mathbb{R}^\delta$ sum up to 1 for $i = 1, \dots, n$. We can easily show that if λ_i in (6.9) is such that $\Lambda = [\lambda_1 \cdots \lambda_n] \in \mathbb{R}_+^{\delta \times n}$ is a full row-rank matrix, then the set of aggregated constraints (6.9) is equivalent to (6.8). This observation leads to the following result.

Proposition 5 *Let the non-negative weights λ 's be such that $[\lambda_1, \dots, \lambda_n]$ has full row rank. Then the solution of the coarse model is feasible (hence optimal) to the solution of the semi-coarse model (6.7):*

$$\begin{aligned} & \underset{v, w, x, y}{\text{minimize}} && a^T y + \sum_{i=1}^n \left(x_i \mathbf{b}^T \bar{X} + v_i \mathbf{c}^T \bar{V} + w_i \mathbf{d}^T \bar{W} \right) \\ & \text{subject to} && \mathbf{A}y + x_i \lambda_i^T \mathbf{B} \bar{X} + v_i \lambda_i^T \mathbf{C} \bar{V} + w_i \lambda_i^T \mathbf{D} \bar{W} \leq \lambda_i^T \mathbf{f}, \quad i = 1, \dots, n \\ & && y \in \mathbb{Z}_+^m, \quad x_i \in \{0, 1\}, \quad v_i = x_i, \quad w_i \in [0, 1], \quad i = 1, \dots, n. \end{aligned} \tag{6.10}$$

Thus, we have shown that if the data is periodic as in (6.1)–(6.2), then the full, semi-coarse, and coarse models are equivalent. Hence, our approach converges in one iteration. Our argument has assumed that there is no coupling between successive periods, resulting in a block-diagonal structure of B , C , and D . In the application of the cogeneration problem in Sect. 3, there exists coupling in time, for example, in the form of storage in the water tanks. In practice, the commercial buildings we consider indeed have periodic diurnal patterns, and hence coupling between successive days is

relatively weak. We believe that our approach can be useful in cases where interdaily coupling is small.

Appendix 2: Full model for the cogeneration problem

For the sake of completeness, we describe here our cogeneration model.

Index sets. We use calligraphic upper-case letters for all sets. The set of technologies: batteries, boilers, CHP-SOFC, power SOFC, and water tank storage is denoted by \mathcal{J} . We use *batt*, *boil*, *pow*, *chp*, and *stor* for shortcuts. The subset $\mathcal{J}_g = \{\text{power SOFC, CHP-SOFC}\}$ denotes SOFC generation technologies that require on/off operations at the second stage. Each technology $j \in \mathcal{J}_g$ has a number of identical units, and the index set is denoted by \mathcal{U}_j . \mathcal{T} is the index set of time, \mathcal{M} is the index set of months, and $\mathcal{T}_m \subset \mathcal{T}$ is the index set of time for each month $m \in \mathcal{M}$.

Variables. We use lower-case letters to denote variables. The first-stage variables in our model are the number of units of technology $j \in \mathcal{J}$, which we denote by $y_j \in \mathbb{Z}_+$. All other variables model operation of the installed units, and they are our second-stage variables. We use the subscript t to indicate the time period: b_t and b_t^{IO} are the power storage and input/output for batteries, respectively; s_t , and s_t^{out} are the heat storage and output for water tanks, respectively; p_{jt} and $q_{jt} \geq 0$ are the power and heat output of technology j , respectively; $u_t \geq 0$ is the power purchased from the utility and $u_m^{\text{max}} \geq 0$ is the maximum power purchased in month m ; $x_{ijt} \in \{0, 1\}$ indicates whether unit i of technology j operates in time period t ; and $w_{ijt} \in [0, 1]$ is an auxiliary variable to indicate whether unit i of technology j switches from t to $t + 1$. Note that we do not impose a binary constraint on w_{ijt} because it takes a binary value at the solution due to the switching constraint (7.1g) and the binary constraint $x_{ijt} \in \{0, 1\}$. We use the convention $t \in \mathcal{T}$, $j \in \mathcal{J}$, $i \in \mathcal{U}_j$, $m \in \mathcal{M}$ unless explicitly mentioned.

Parameters. We use upper-case letters to denote parameters and constants: C_j is the capital and installation cost of technology j ; H is the number of hours in the lifetime of technology (e.g., $H = 87600$ for a ten-year model); T is the number of hours in the problem horizon; Y is the hourly discount factor with 3% annual interest rate (i.e., $Y = 1 - 0.03/8760$); I is the hourly increase factor with 10% annual increase rate (i.e., $I^{8760} = 1.1$); M_j is the operation and maintenance cost of technology j ; P_t and G_t are the price for electricity and natural gas, respectively; P_m^{max} is the peak demand price for power from the utility in month m ; W_j is the cost of switching a unit of technology j on or off; S_j^c is the thermal capacity per unit for technology j ; B^c is the power capacity per unit for batteries; D_t^P and D_t^Q are the power and heat demand in period t , respectively; R_j^{min} and R_j^{max} are the minimum and maximum power output when a unit of technology j is turned on, respectively; E_j^P and E_j^Q are the power

and thermal efficiency of technology j , respectively; and L^P and L^Q are the average power loss in battery and heat loss in water tank storage, respectively.

Cost function. We now formulate the complete MILP model for the cogeneration problem in (7.1). The objective function consists of two parts: the capital and installation cost in the first stage and the operation cost in the second stage. The operation cost includes the peak power usage, operation and maintenance, switching units, gas, and purchased power costs. We discount the second-stage costs with an *hourly* discount factor based on a 3% *annual* interest rate; hence $Y = 1 - 0.03/8760$. Therefore, the cost incurred in hour t is multiplied by the discount factor Y^t .³ The peak demand charge at the end of month m is discounted with Y^{tm} . We take into account a 10% annually increase of the maintenance and switching costs over time. This is done by multiplying I^t to cost terms in (7.1a) where $I = 1 + 1.088 \times 10^{-5}$. To compare cost over different time horizons T , we compute the average hourly cost and multiply the result with the lifetime of technologies, H , resulting in the factor H/T in the second-stage cost in (7.1).

Constraints. The constraints in the MILP model (7.1) can be divided into three groups. The first group of constraints couples first- and second-stage variables; in particular, the number of on-units for fuel cells and the capacity of battery, storage, and boiler are bounded by the number of units purchased in the first stage, as described in (7.1e), (7.1k), (7.1o), and (7.1p), respectively. The second group of constraints couples variables across technologies; in particular, technologies are coordinated to meet the power demand (7.1c) and the heat demand (7.1l). We note that dualizing these constraints results in decoupled variables for each technology; see, for example, [3, 14]. However, MILP (7.1) does not lend itself to this Lagrangian relaxation approach because of coupling constraints in the third group. This group of constraints couples variables over time periods in the second stage. In particular, the battery and storage constraints (7.1i) and (7.1m) link variables over the entire horizon. The heat storage constraint (7.1m) is a combination of two constraints, namely, the storage equation $s_{t+1} = (1 - L^Q)s_t - s_t^{\text{out}} + E_{\text{chp}}^Q s_t^{\text{in}}$ and the upper bound on the heat generated by CHP-SOFC, $s_t^{\text{in}} \leq p_{\text{chp},t}/E_{\text{chp}}^P$. Eliminating the heat input s_t^{in} yields (7.1m). In addition, constraint (7.1h), which models the maximum power purchased from the utility in each months, links variables over time instances in each month. Moreover, constraint (7.1g), which models on/off switches of the generation units, couples binary variables x_{ijt} over two *immediate* time instances. Boundary conditions (7.1j) and (7.1n) for battery and storage couple variables at both ends of the entire horizon.

We conclude this subsection by explaining the rest of the constraints in (7.1). The number of purchased units is upper bounded in (7.1b). Since all units of the same

³ For comparison, the annual discount factor is $Y^{8760} \approx 97.044\%$, which is consistent with the 3% interest rate; and a ten-year discount factor is $Y^{87600} \approx 74.082\%$.

technology are identical, we introduce a symmetry breaking constraint (7.1f) that states that if unit $i + 1$ is on, then unit i must be on as well. When unit i of technology j is turned on, the power generated p_{ijt} is bounded by the minimum and maximum capacity in (7.1d). The non-negativity constraint for variables is given by (7.1q).

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in \mathcal{J}} C_j y_j + \frac{H}{T} \left\{ \sum_{m \in \mathcal{M}} Y^{t_m} P_m^{\max} u_m^{\max} / \right. \\ & \left. + \sum_{t \in \mathcal{T}} Y^t \left[\sum_{j \in \mathcal{J}_g} (I^t \cdot M_j p_{jt} + \sum_{i \in \mathcal{U}_j} I^t \cdot W_j w_{ijt}) + G_t q_t + P_t u_t \right] \right\} \end{aligned} \tag{7.1a}$$

$$\text{subject to} \quad y_j \in \mathbb{Z}, \quad 0 \leq y_j \leq |\mathcal{U}_j| \tag{7.1b}$$

$$\sum_{j \in \mathcal{J}_g} p_{jt} + u_t - b_t^{\text{IO}} \geq D_t^P \tag{7.1c}$$

$$R_j^{\min} x_{ijt} \leq p_{ijt} \leq R_j^{\max} x_{ijt}, \quad p_{jt} = \sum_{i \in \mathcal{U}_j} p_{ijt} \quad j \in \mathcal{J}_g \tag{7.1d}$$

$$x_{ijt} \in \{0, 1\}, \quad \sum_{i \in \mathcal{U}_j} x_{ijt} \leq y_j \quad j \in \mathcal{J}_g \tag{7.1e}$$

$$x_{(i+1)jt} \leq x_{ijt} \quad j \in \mathcal{J}_g, i < |\mathcal{U}_j| \tag{7.1f}$$

$$x_{ij(t+1)} - x_{ijt} \leq w_{ijt}, \quad x_{ijt} - x_{ij(t+1)} \leq w_{ijt}, \quad 0 \leq w_{ijt} \leq 1, \\ j \in \mathcal{J}_g, t < |\mathcal{T}| \tag{7.1g}$$

$$0 \leq u_t \leq u_m^{\max} \quad t \in \mathcal{T}_m \tag{7.1h}$$

$$b_{t+1} = (1 - L^P) b_t + b_t^{\text{IO}} \quad t < |\mathcal{T}| \tag{7.1i}$$

$$b_1 = b_{|\mathcal{T}|} \tag{7.1j}$$

$$0 \leq b_t \leq B^c y_{\text{batt}} \tag{7.1k}$$

$$s_t^{\text{out}} + q_t \geq D_t^Q \tag{7.1l}$$

$$s_{t+1} - (1 - L^Q) s_t + s_t^{\text{out}} \leq (E_{\text{chp}}^Q / E_{\text{chp}}^P) p_{\text{chp},t} \quad t < |\mathcal{T}| \tag{7.1m}$$

$$s_1 = s_{|\mathcal{T}|} \tag{7.1n}$$

$$s_t^{\text{out}} \leq s_t \leq S_{\text{stor}}^c y_{\text{stor}} \tag{7.1o}$$

$$q_t \leq S_{\text{boil}}^c y_{\text{boil}} \tag{7.1p}$$

$$b_t, p_{jt}, q_t, s_t, s_t^{\text{out}}, u_t \geq 0 \tag{7.1q}$$

Appendix 3: Tables of numerical results

Table 7 Problem size and AMPL/CPLEX solution information for the cogeneration problem (7.1) for five buildings

Building	RelGap	Days	Binary	Continuous	Constraint	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	MSG
School	9.5×10^{-3}	4	1152	2994	6698	0	6940	1	1	2	2	1	2.86	5.94×10^6	0
School	9.43×10^{-3}	7	2016	5226	11,738	10	12,500	2	2	5	7	7	7.67	1.7×10^7	0
School	6.27×10^{-4}	14	4032	10,434	23,498	141	23,600	2	8	5	7	8	1.95	2.44×10^7	0
School	9.03×10^{-3}	28	8064	20,850	47,018	360	55,500	4	8	5	7	8	69.5	2.44×10^7	0
School	3.43×10^{-3}	84	24,192	62,514	1.41×10^5	532	1.92×10^5	3	8	5	7	8	683	2.15×10^7	0
School	2.2×10^{-2}	364	1.05×10^5	2.71×10^5	6.11×10^5	200	7.39×10^5	4	8	5	7	8	10,800	2.35×10^7	1
SuperMarket	5.66×10^{-3}	4	1152	2994	6698	20	13,000	1	1	3	2	4	4.87	1.06×10^7	0
SuperMarket	8.89×10^{-3}	7	2016	5226	11,738	30	32,600	1	1	3	3	3	14.9	1.11×10^7	0
SuperMarket	6.18×10^{-3}	14	4032	10,434	23,498	140	62,300	1	1	3	3	3	42.3	1.12×10^7	0
SuperMarket	5.37×10^{-3}	28	8064	20,850	47,018	360	97,100	2	1	3	4	4	115	1.11×10^7	0
SuperMarket	9.36×10^{-3}	84	24,192	62,514	1.41×10^5	498	2.91×10^5	1	1	3	3	4	1380	1.07×10^7	0
SuperMarket	0.83	364	1.0×10^5	2.71×10^5	6.11×10^5	0	6.04×10^5	6	3	5	7	8	10,800	2.98×10^7	1
Restaurant	7.34×10^{-3}	4	1152	2994	6698	1050	1.3×10^5	2	1	1	1	1	58.1	2.58×10^6	0
Restaurant	9.99×10^{-3}	7	2016	5226	11,738	2307	2.41×10^5	2	1	1	1	1	128	2.32×10^6	0
Restaurant	1×10^{-2}	14	4032	10,434	23,498	3883	4.92×10^5	2	1	1	1	2	321	2.6×10^6	0
Restaurant	9.99×10^{-3}	28	8064	20,850	47,018	35,836	2.22×10^6	2	1	1	1	1	3500	2.6×10^6	0
Restaurant	2.49×10^{-2}	84	24,192	62,514	1.41×10^5	13,252	3.99×10^6	3	1	1	1	1	10,600	2.45×10^6	1
Restaurant	0.38	364	1.05×10^5	2.71×10^5	6.11×10^5	0	5.87×10^5	0	1	1	1	1	10,800	4.15×10^6	1
Retail	8.07×10^{-3}	4	1152	2994	6698	140	28,400	2	1	1	1	1	6.46	3.01×10^6	0
Retail	8.78×10^{-3}	7	2016	5226	11,738	1557	1.8×10^5	2	1	1	1	2	109	2.44×10^6	0

Table 7 continued

Building	RelGap	Days	Binary	Continuous	Constraint	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	MSG
Retail	9.59×10^{-3}	14	4032	10,434	23,498	7967	5.23×10^5	2	1	1	1	2	403	2.77×10^6	0
Retail	9.96×10^{-3}	28	8064	20,850	47,018	1902	3.69×10^5	2	1	1	1	2	1040	2.8×10^6	0
Retail	1.96×10^{-2}	84	24,192	62,514	1.41×10^5	21,523	4.57×10^6	3	1	1	1	1	10,600	2.69×10^6	1
Retail	0.36	364	1.05×10^5	2.71×10^5	6.11×10^5	0	6×10^5	0	1	1	1	1	10,800	4.45×10^6	1
Hospital	3.3×10^{-3}	4	1152	2994	6698	0	7750	3	3	5	6	2	2.56	3.35×10^7	0
Hospital	9.82×10^{-3}	7	2016	5226	11,738	0	6360	6	3	5	7	8	1.42	4.32×10^7	0
Hospital	7.82×10^{-3}	14	4032	10,434	23,498	0	13,600	6	3	5	7	8	2.35	4.3×10^7	0
Hospital	7.04×10^{-3}	28	8064	20,850	47,018	130	33,500	6	3	5	7	8	23.7	4.22×10^7	0
Hospital	8.95×10^{-3}	84	24,192	62,514	1.41×10^5	0	1×10^5	2	3	5	7	8	228	4.09×10^7	0
Hospital	5.61×10^{-3}	364	1.05×10^5	2.71×10^5	6.11×10^5	0	4.6×10^5	2	3	5	7	8	8810	4.18×10^7	0

Model instances that reach 3-h time limit are indicated by MSG = 1

Table 8 Problem size and AMPL/CPLEX solution information for the semi-coarse model (9.1) for five buildings

Building	RelGap	Days	Binary	Continuous	Constraint	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	MSG
School	9.18×10^{-3}	364	13,104	25,498	1.85×10^5	250	29,800	1	10	5	7	6	2.94	4.43×10^7	0
School	6.59×10^{-3}	728	26,208	50,991	3.71×10^5	570	70,100	2	10	5	7	6	130	5.41×10^7	0
School	1.47×10^{-3}	1456	52,416	1.02×10^5	7.41×10^5	3722	2.91×10^5	0	10	5	7	6	1130	6.12×10^7	0
School	9.37×10^{-3}	2184	78,624	1.53×10^5	1.11×10^6	2742	4.06×10^5	4	10	5	7	6	2050	7.48×10^7	0
School	3.16×10^{-3}	2912	1.05×10^5	2.04×10^5	1.48×10^6	6729	5.58×10^5	4	10	5	7	6	4610	9×10^7	0
School	3.89×10^{-3}	3640	1.31×10^5	2.55×10^5	1.85×10^6	8510	7.09×10^5	1	10	5	7	6	7450	1.08×10^8	0
SuperMarket	8.3×10^{-3}	364	13,104	25,498	1.85×10^5	1847	3.21×10^5	0	2	4	4	2	297	1.88×10^7	0
SuperMarket	6.98×10^{-3}	728	26,208	50,991	3.71×10^5	3122	1.44×10^6	0	2	5	2	2	1740	2.06×10^7	0
SuperMarket	9.72×10^{-3}	1456	52,416	1.02×10^5	7.41×10^5	5298	3.33×10^6	0	2	5	2	2	5590	2.52×10^7	0
SuperMarket	9.98×10^{-3}	2184	78,624	1.53×10^5	1.11×10^6	9057	9.77×10^5	0	2	5	6	2	9550	3×10^7	0
SuperMarket	8.16×10^{-2}	2912	1.05×10^5	2.04×10^5	1.48×10^6	500	5.8×10^5	0	3	5	6	2	6920	3.84×10^7	0
SuperMarket	0.1	3640	1.31×10^5	2.55×10^5	1.85×10^6	486	6.39×10^5	0	3	5	3	2	10,800	4.64×10^7	1
Restaurant	4.01×10^{-3}	364	13,104	29,866	1.85×10^5	487	62,200	0	1	1	1	1	134	4.7×10^6	0
Restaurant	5.1×10^{-3}	728	26,208	59,727	3.71×10^5	390	1.16×10^5	0	1	1	1	1	404	5.25×10^6	0
Restaurant	9.28×10^{-3}	1456	52,416	1.19×10^5	7.41×10^5	508	1.66×10^5	0	1	1	2	1	1560	6.34×10^6	0
Restaurant	1.83×10^{-2}	2184	78,624	1.79×10^5	1.11×10^6	3015	1.29×10^6	0	1	3	1	1	9510	7.63×10^6	0
Restaurant	0.31	2912	1.05×10^5	2.39×10^5	1.48×10^6	727	5.71×10^5	0	1	1	1	1	8090	1.3×10^7	1
Restaurant	0.43	3640	1.31×10^5	2.99×10^5	1.85×10^6	515	4.6×10^5	0	1	0	0	1	9950	1.85×10^7	1
Retail	9.84×10^{-3}	364	13,104	27,318	1.85×10^5	482	1.58×10^5	0	1	2	1	1	167	6.23×10^6	0
Retail	6.37×10^{-3}	728	26,208	53,903	3.71×10^5	511	2×10^5	0	1	2	1	1	502	6.89×10^6	0
Retail	4.81×10^{-3}	1456	52,416	1.08×10^5	7.41×10^5	4586	1.93×10^6	0	1	2	1	1	5080	8.37×10^6	0
Retail	0.17	2184	78,624	1.64×10^5	1.11×10^6	2077	3.58×10^6	0	2	5	7	1	10,800	1.21×10^7	1

Table 8 continued

Building	RelGap	Days	Binary	Continuous	Constraint	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	MSG
Retail	0.31	2912	1.05×10^5	2.19×10^5	1.48×10^6	779	1.2×10^6	0	2	5	6	1	10,800	1.68×10^7	1
Retail	0.43	3640	1.31×10^5	2.73×10^5	1.85×10^6	461	7.11×10^5	0	2	2	1	1	10,700	2.5×10^7	1
Hospital	9.94×10^{-3}	364	13,104	27,318	1.85×10^5	500	51,300	0	5	5	7	2	51.8	5.62×10^7	0
Hospital	9.47×10^{-3}	728	26,208	54,631	3.71×10^5	1423	2.02×10^5	0	5	5	7	2	498	6.19×10^7	0
Hospital	9.31×10^{-3}	1456	52,416	1.09×10^5	7.41×10^5	2542	3.84×10^5	0	5	5	7	2	2040	7.41×10^7	0
Hospital	9.87×10^{-3}	2184	78,624	1.64×10^5	1.11×10^6	2624	5.82×10^5	0	5	5	7	2	2960	8.73×10^7	0
Hospital	6.48×10^{-3}	2912	1.05×10^5	2.19×10^5	1.48×10^6	6426	7.73×10^5	0	4	5	7	2	8680	1.03×10^8	0
Hospital	1.21×10^{-2}	3640	1.31×10^5	2.73×10^5	1.85×10^6	3043	9.65×10^5	0	5	5	7	2	7990	1.22×10^8	0

Model instances that reach 3-h time limit are indicated by MSG = 1

Table 9 Problem size and AMPL/CPLEX solution information for the first iteration of the coarse model (10.1) for five buildings

Building	Year	Binary	Continuous	Constraint	Nodes	LP-iter	Time	Bat	Boil	Chp	Pow	Stor	ObjVal	MSG
School	1	13,104	25,498	53,177	330	34,230	32.7	1	10	5	7	6	4.82×10^7	0
School	2	26,208	50,991	1.11×10^5	760	60,021	104.03	4	10	5	7	6	4.94×10^7	0
School	4	52,416	1.02×10^5	2.21×10^5	2404	2.92×10^5	1025.75	4	10	5	7	6	6.74×10^7	0
School	6	78,624	1.53×10^5	3.41×10^5	6139	4.5×10^5	2966.99	4	10	5	7	6	7.41×10^7	0
School	8	1.05×10^5	2.04×10^5	4.34×10^5	498	3.74×10^5	1073.26	3	10	5	7	8	9.16×10^7	0
School	10	1.31×10^5	2.55×10^5	5.49×10^5	5305	5.77×10^5	4439.90	4	10	5	7	6	1.2×10^8	0
SuperMarket	1	13,104	25,498	75,556	6575	4.48×10^5	392.99	0	2	5	2	2	1.91×10^7	0
SuperMarket	2	26,208	50,991	1.34×10^5	525	1.41×10^5	334.80	0	3	5	4	3	2.28×10^7	0
SuperMarket	4	52,416	1.02×10^5	3.07×10^5	37,585	3.47×10^6	10,805.73	0	2	5	7	2	2.53×10^7	1
SuperMarket	6	78,624	1.53×10^5	4.59×10^5	7887	1.17×10^6	10,494.83	0	3	5	7	2	3.01×10^7	0
SuperMarket	8	1.05×10^5	2.04×10^5	6.27×10^5	6177	6.48×10^5	8583.82	0	3	5	5	2	1.64×10^7	0
SuperMarket	10	1.31×10^5	2.55×10^5	7.69×10^5	1569	7.59×10^5	10,814.87	0	3	5	7	2	1.6×10^7	1
Restaurant	1	13,104	29,866	50,995	485	42,610	71.41	0	1	1	2	1	4.73×10^6	0
Restaurant	2	26,208	59,727	1.08×10^5	512	75,718	215.27	0	1	1	2	1	5.23×10^6	0
Restaurant	4	52,416	1.19×10^5	1.97×10^5	3450	3.82×10^5	3215.74	0	1	1	1	1	6.29×10^6	0
Restaurant	6	78,624	1.79×10^5	3.15×10^5	7492	1.06×10^6	6112.33	0	1	1	1	1	7.5×10^6	0
Restaurant	8	1.05×10^5	2.39×10^5	4.15×10^5	4902	1.28×10^6	10,428.87	0	1	5	1	1	9.24×10^6	0
Restaurant	10	1.31×10^5	2.95×10^5	5.15×10^5	662	8.4×10^5	10,714.63	0	1	5	1	1	1.08×10^7	1
Retail	1	13,104	26,954	58,550	4456	3.54×10^5	719.02	0	1	2	2	1	6.23×10^6	0
Retail	2	26,208	53,903	1.2×10^5	7734	7.58×10^5	1864.11	0	1	2	1	1	6.91×10^6	0
Retail	4	52,416	1.09×10^5	2.34×10^5	18,193	2.48×10^6	10,807.85	0	1	5	7	1	8.79×10^6	1
Retail	6	78,624	1.62×10^5	3.56×10^5	1536	1.14×10^6	10,808.54	0	2	5	7	1	1.06×10^7	1

Table 9 continued

Building	Year	Binary	Continuous	Constraint	Nodes	LP-iter	Time	Bat	Boil	Chp	Pow	Stor	ObjVal	MSG
Retail	8	1.05×10^5	2.16×10^5	5.15×10^5	1612	1.42×10^6	10,336.09	0	1	5	7	1	1.32×10^7	0
Retail	10	1.31×10^5	2.73×10^5	6.08×10^5	2610	1.24×10^6	10,333.87	0	1	5	7	1	1.55×10^7	1
Hospital	1	13,104	27,318	56,514	518	50,675	38.92	0	5	5	7	2	5.59×10^7	0
Hospital	2	26,208	54,631	1.11×10^5	512	1.65×10^5	132.42	0	5	5	7	2	6.16×10^7	0
Hospital	4	52,416	1.09×10^5	2.24×10^5	2010	3.57×10^5	939.48	0	5	5	7	2	7.37×10^7	0
Hospital	6	78,624	1.64×10^5	3.34×10^5	7220	59,811	4308.58	0	5	5	7	2	8.75×10^7	0
Hospital	8	1.05×10^5	2.19×10^5	4.46×10^5	8777	8.14×10^5	8113.05	0	5	5	7	2	1.04×10^8	0
Hospital	10	1.31×10^5	2.73×10^5	5.53×10^5	8608	9.04×10^5	10,736.90	0	5	5	7	2	1.21×10^8	1

Model instances that reach 3-h time limit are indicated by MSG = 1

Table 10 Comparison between the full model (7.1) and the semi-coarse model (9.1) for five buildings

Building	Days	Model	Binary	Continuous	Constraint	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	FirstStageCost
School	28	Full	8064	20,850	47,018	360	55,500	4	8	5	7	8	69.5	2.44×10^7	9.93×10^5
School	28	Semi	1008	4442	14,254	0	2470	5	11	5	7	7	1.95	3.84×10^7	1.04×10^6
School	56	Full	16,128	41,682	94,058	340	1.17×10^5	2	8	5	7	8	271	2.12×10^7	9.33×10^5
School	56	Semi	2016	8866	28,506	0	4130	6	9	5	7	6	2.65	3.56×10^7	1.01×10^6
School	84	Full	24,192	62,514	1.41×10^5	532	1.92×10^5	3	8	5	7	8	683	2.15×10^7	9.63×10^5
School	84	Semi	3024	13,290	42,758	0	7030	6	10	5	7	7	8.67	3.73×10^7	1.06×10^6
SuperMarket	28	Full	8064	20,850	47,018	360	97,100	2	1	3	4	4	115	1.11×10^7	5.09×10^5
SuperMarket	28	Semi	1008	4442	14,254	111	6110	1	2	4	2	2	6.57	1.51×10^7	4.37×10^5
SuperMarket	56	Full	16,128	41,682	94,058	650	3.36×10^5	1	1	3	5	4	1180	1.07×10^7	5.19×10^5
SuperMarket	56	Semi	2016	8866	28,506	221	9850	1	2	4	2	3	13.8	1.43×10^7	4.57×10^5
SuperMarket	84	Full	24,192	62,514	1.41×10^5	498	2.91×10^5	1	1	3	3	4	1380	1.07×10^7	4.39×10^5
SuperMarket	84	Semi	3024	13,290	42,758	250	16,900	1	2	4	2	3	24	1.42×10^7	4.57×10^5
Restaurant	28	Full	8064	20,850	47,018	35,836	2.22×10^6	2	1	1	1	1	3500	2.6×10^6	1.99×10^5
Restaurant	28	Semi	1008	4778	14,254	145	9090	1	1	1	1	1	7.32	4.01×10^6	1.69×10^5
Restaurant	56	Full	16,128	41,682	94,058	13,805	2.7×10^6	3	1	1	1	1	9890	2.46×10^6	2.29×10^5
Restaurant	56	Semi	2016	9538	28,506	505	55,500	1	1	1	1	1	49	4.04×10^6	1.69×10^5
Restaurant	84	Full	24,192	62,514	1.41×10^5	13,252	3.99×10^6	3	1	1	1	1	10,600	2.45×10^6	2.29×10^5
Restaurant	84	Semi	3024	14,298	42,758	495	72,100	1	1	1	1	1	103	4.11×10^6	1.69×10^5
Retail	28	Full	8064	20,850	47,018	1902	3.69×10^5	2	1	1	1	2	1040	2.8×10^6	2.19×10^5
Retail	28	Semi	1008	4582	14,254	40	8620	2	2	2	1	2	6.91	5.91×10^6	2.97×10^5
Retail	56	Full	16,128	41,682	94,058	47,259	5.79×10^6	3	1	1	1	2	9950	2.79×10^6	2.49×10^5
Retail	56	Semi	2016	9146	28,506	284	21,400	2	1	2	1	2	18.6	5.58×10^6	2.84×10^5

Table 10 continued

Building	Days	Model	Binary	Continuous	Constraint	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	FirstStageCost
Retail	84	Full	24,192	62,514	1.41×10^5	21,523	4.57×10^6	3	1	1	1	1	10,600	2.69×10^6	2.29×10^5
Retail	84	Semi	3024	13,710	42,758	490	31,600	2	1	2	1	2	76.3	5.07×10^6	2.84×10^5
Hospital	28	Full	8064	20,850	47,018	130	33,500	6	3	5	7	8	23.7	4.22×10^7	9.86×10^5
Hospital	28	Semi	1008	4582	14,254	0	4690	6	4	5	7	2	2.36	5.05×10^7	8.79×10^5
Hospital	56	Full	16,128	41,682	94,058	0	59,300	6	3	5	7	8	78.5	4.15×10^7	9.86×10^5
Hospital	56	Semi	2016	9146	28,506	0	10,700	6	4	5	7	2	7.31	4.93×10^7	8.79×10^5
Hospital	84	Full	24,192	62,514	1.41×10^5	0	1×10^5	2	3	5	7	8	228	4.09×10^7	9.66×10^5
Hospital	84	Semi	3024	13,710	42,758	0	16,200	6	4	5	7	2	12.5	4.85×10^7	8.79×10^5

Table 11 Comparison between the full model (7.1) with and without fixing the first-stage solutions from the semi-coarse model (9.1) for five buildings

Building	Days	Fix1stStage	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	FirstStageCost
School	28	No	360	55,500	4	8	5	7	8	69.5	2.43×10^7	9.93×10^5
School	28	Yes	200	39,000	5	11	5	7	7	24.9	2.45×10^7	1.04×10^6
School	56	No	340	1.17×10^5	2	8	5	7	8	271	2.12×10^7	9.33×10^5
School	56	Yes	400	82,600	6	9	5	7	5	72.6	2.14×10^7	1.01×10^6
School	84	No	532	1.92×10^5	3	8	5	7	8	683	2.15×10^7	9.63×10^5
School	84	Yes	1532	2.52×10^5	6	10	5	7	7	548	2.16×10^7	1.06×10^6
SuperMarket	28	No	360	97,100	2	1	3	4	4	115	1.11×10^7	5.09×10^5
SuperMarket	28	Yes	0	15,700	1	2	4	2	2	4.32	1.15×10^7	4.37×10^5
SuperMarket	56	No	650	3.36×10^5	1	1	3	5	4	1.180	1.07×10^7	5.19×10^5
SuperMarket	56	Yes	0	32,800	1	2	4	2	3	10.7	1.1×10^7	4.57×10^5
SuperMarket	84	No	498	2.91×10^5	1	1	3	3	4	1380	1.07×10^7	4.39×10^5
SuperMarket	84	Yes	0	92,700	1	2	4	2	3	33.4	1.1×10^7	4.57×10^5
Restaurant	28	No	35,836	2.22×10^6	2	1	1	1	1	3.500	2.6×10^6	1.99×10^5
Restaurant	28	Yes	8.15×10^5	2.18×10^7	1	1	1	1	1	10,200	2.7×10^6	1.69×10^5
Restaurant	56	No	13,805	2.7×10^6	3	1	1	1	1	9890	2.46×10^6	2.29×10^5
Restaurant	56	Yes	3.09×10^5	1.37×10^7	1	1	1	1	1	10,300	2.55×10^6	1.69×10^5
Restaurant	84	No	13,252	3.99×10^6	3	1	1	1	1	10,600	2.45×10^6	2.29×10^5
Restaurant	84	Yes	1.75×10^5	9.59×10^6	1	1	1	1	1	10,300	2.51×10^6	1.69×10^5
Retail	28	No	1902	3.69×10^5	2	1	1	1	2	1040	2.8×10^6	2.19×10^5
Retail	28	Yes	460	27,900	2	2	2	1	2	14	2.87×10^6	2.97×10^5
Retail	56	No	47,259	5.79×10^6	3	1	1	1	2	9950	2.79×10^6	2.49×10^5
Retail	56	Yes	1688	94,100	2	1	2	1	2	156	2.83×10^6	2.84×10^5

Table 11 continued

Building	Days	Fix 1stStage	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	FirstStageCost
Retail	84	No	21,523	4.57×10^6	3	1	1	1	1	10,600	2.69×10^6	2.29×10^5
Retail	84	Yes	2625	1.35×10^5	2	1	2	1	2	296	2.73×10^6	2.84×10^5
Hospital	28	No	130	33,500	6	3	5	7	8	23.7	4.22×10^7	9.86×10^5
Hospital	28	Yes	0	28,500	6	4	5	7	2	21.9	4.26×10^7	8.79×10^5
Hospital	56	No	0	59,300	6	3	5	7	8	78.5	4.15×10^7	9.86×10^5
Hospital	56	Yes	0	58,600	6	4	5	7	2	62.8	4.19×10^7	8.79×10^5
Hospital	84	No	0	1×10^5	2	3	5	7	8	228	4.09×10^7	9.66×10^5
Hospital	84	Yes	0	91,800	6	4	5	7	2	117	4.12×10^7	8.79×10^5

Appendix 4: Semi-coarse model for the cogeneration problem

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in \mathcal{J}} C_j y_j + \frac{H}{T} \left\{ \sum_{m \in \mathcal{M}} Y^m P_m^{\max} u_m^{\max} + \sum_d \bar{Y}_d^T \left[\sum_{j,i,k,l} \bar{p}_{ijkl} M_j \bar{P}_{jkl} \right. \right. \\ & \left. \left. + \sum_{d,j,i,k} \bar{x}_{ijk} W_j \bar{W}_{jk} + \sum_{d,k \in \mathcal{P}_q} \bar{q}_{dk} \bar{Q}_k + \sum_{d,k \in \mathcal{P}_u} \bar{u}_{dk} \bar{U}_k \right] \right\} \end{aligned} \tag{9.1a}$$

$$\text{subject to} \quad y_j \in \mathbb{Z}, \quad 0 \leq y_j \leq |\mathcal{U}_j| \quad j \in \mathcal{J} \tag{9.1b}$$

$$\sum_{j,i,k,l} \bar{p}_{ijkl} \bar{P}_{jkl} + \sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \bar{U}_k - \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \bar{B}_k^{\text{IO}} \geq \bar{D}_d^P \tag{9.1c}$$

$$\bar{x}_{ijk} \in \{0, 1\}, \quad \sum_{k \in \mathcal{P}_o} \bar{x}_{ijk} \leq 1 \tag{9.1d}$$

$$\sum_{i,k} \bar{x}_{ijk} \bar{X}_{jk} \leq y_j \mathbf{1} \tag{9.1e}$$

$$\sum_{k \in \mathcal{P}_o} \bar{x}_{(i+1)jdk} \bar{X}_{jk} \leq \sum_{k \in \mathcal{P}_o} \bar{x}_{ijk} \bar{X}_{jk} \quad i < |\mathcal{U}_j| \tag{9.1f}$$

$$\bar{p}_{ijk} \in [0, 1], \quad \sum_{l \in \mathcal{P}_p} \bar{p}_{ijkl} = \bar{x}_{ijk}, \quad \sum_{k \in \mathcal{P}_o} \sum_{l \in \mathcal{P}_p} \bar{p}_{ijkl} \leq 1 \tag{9.1g}$$

$$0 \leq \sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \bar{U}_k \leq u_m^{\max} \mathbf{1} \quad d \in \mathcal{D}_m \tag{9.1h}$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{(d+1)k} \bar{B}_k(1) = \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} [(1 - L^P) \bar{B}_k(\delta) + \bar{B}_k^{\text{IO}}(\delta)] \quad d < |\mathcal{D}| \tag{9.1i}$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{1k} \bar{B}_k(1) = \sum_{k \in \mathcal{P}_b} \bar{b}_{|\mathcal{D}|k} \bar{B}_k(\delta) \tag{9.1j}$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \bar{B}_k \leq B^c y_{\text{batt}} \mathbf{1} \tag{9.1k}$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \bar{S}_k^{\text{out}} + \sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \bar{Q}_k \geq \bar{D}_d^Q \tag{9.1l}$$

$$\begin{aligned} & \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} [S_\delta - (1 - L^Q) I_\delta] \bar{S}_k + \sum_{k \in \mathcal{P}_s} \bar{s}_{(d+1)k} E_\delta \bar{S}_k \quad d < |\mathcal{D}| \\ & + \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \bar{S}_k^{\text{out}} \leq (E_j^Q / E_j^P) \sum_{i,k,l} \bar{p}_{ijkl} \bar{P}_{jkl} \quad j = \text{chp} \end{aligned} \tag{9.1m}$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{1k} \bar{S}_k(1) = \sum_{k \in \mathcal{P}_s} \bar{s}_{|\mathcal{D}|k} \bar{S}_k(\delta) \tag{9.1n}$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \bar{S}_k^{\text{out}} \leq S_{\text{stor}}^c y_{\text{stor}} \mathbf{1} \tag{9.1o}$$

$$\sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \bar{Q}_k \leq S_{\text{boil}}^c y_{\text{boil}} \mathbf{1} \tag{9.1p}$$

$$\bar{u}_{dk}, \bar{b}_{dk}, \bar{s}_{dk}, \bar{q}_{dk} \in [0, 1] \tag{9.1q}$$

$$\sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \leq 1, \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \leq 1, \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \leq 1, \sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \leq 1. \tag{9.1r}$$

Appendix 5: Coarse model for the cogeneration problem

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in \mathcal{J}} C_j y_j + \frac{H}{T} \left\{ \sum_{m \in \mathcal{M}} Y^{tm} P_m^{\max} u_m^{\max} + \sum_d \bar{Y}_d^T \left[\sum_{j,i,k,l} \bar{p}_{ijkl} M_j \bar{P}_{jkl} \right. \right. \\ & \left. \left. + \sum_{d,j,i,k} \bar{x}_{ijk} W_j \bar{W}_{jk} + \sum_{d,k \in \mathcal{P}_u} \bar{u}_{dk} \bar{U}_k + \sum_{d,k \in \mathcal{P}_q} \bar{q}_{dk} \bar{Q}_k \right] \right\} \end{aligned} \tag{10.1a}$$

$$\text{subject to} \quad y_j \in \mathbb{Z}, 0 \leq y_j \leq |\mathcal{U}_j| \quad j \in \mathcal{J} \tag{10.1b}$$

$$\sum_{j,i,k,l} \bar{p}_{ijkl} \hat{P}_{jkl} + \sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \hat{U}_k - \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \hat{B}_k^{\text{IO}} \geq \hat{D}_d^P \tag{10.1c}$$

$$\bar{x}_{ijk} \in \{0, 1\}, \sum_{k \in \mathcal{P}_o} \bar{x}_{ijk} \leq 1 \tag{10.1d}$$

$$\sum_{i,k} \bar{x}_{ijk} \hat{X}_{jk} \leq \delta \cdot y_j \tag{10.1e}$$

$$\sum_{k \in \mathcal{P}_o} \bar{x}_{ijk} \hat{X}_{jk} \geq \sum_{k \in \mathcal{P}_o} \bar{x}_{(i+1)jdk} \hat{X}_{jk} \quad i < |\mathcal{U}_j| \tag{10.1f}$$

$$\bar{p}_{ijkl} \in [0, 1], \sum_{l \in \mathcal{P}_p} \bar{p}_{ijkl} = \bar{x}_{ijk}, \sum_{k \in \mathcal{P}_o} \sum_{l \in \mathcal{P}_p} \bar{p}_{ijkl} \leq 1 \tag{10.1g}$$

$$0 \leq \sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \hat{U}_k \leq \delta \cdot u_m^{\max} \quad d \in \mathcal{D}_m \tag{10.1h}$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{(d+1)k} \bar{B}_k(1) = \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \left((1 - L^P) \bar{B}_k(\delta) + \bar{B}_k^{\text{IO}}(\delta) \right) \quad d < |\mathcal{D}| \tag{10.1i}$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{1k} \bar{B}_k(1) = \sum_{k \in \mathcal{P}_b} \bar{b}_{|\mathcal{D}|k} \bar{B}_k(\delta) \tag{10.1j}$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \hat{B}_k \leq \delta \cdot B^c y_{\text{batt}} \tag{10.1k}$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \hat{S}_k^{\text{out}} + \sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \hat{Q}_k \geq \hat{D}_d^Q \tag{10.1l}$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} (L^Q \hat{S}_k - \bar{S}_k(1)) + \sum_{k \in \mathcal{P}_s} \bar{s}_{(d+1)k} \bar{S}_k(1) \quad d < |\mathcal{D}|$$

$$+ \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \hat{S}_k^{\text{out}} \leq (E_j^Q / E_j^P) \sum_{i,k,l} \bar{p}_{ijkl} \hat{P}_{jkl} \quad j = \text{chp} \quad (10.1m)$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{1k} \bar{S}_k(1) = \sum_{k \in \mathcal{P}_s} \bar{s}_{|\mathcal{D}|k} \bar{S}_k(\delta) \quad (10.1n)$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \hat{S}_k^{\text{out}} \leq \delta \cdot S_{\text{stor}}^c y_{\text{stor}} \quad (10.1o)$$

$$\sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \hat{Q}_k \leq \delta \cdot S_{\text{boil}}^c y_{\text{boil}} \quad (10.1p)$$

$$\bar{u}_{dk}, \bar{b}_{dk}, \bar{s}_{dk}, \bar{q}_{dk} \in [0, 1]$$

$$\sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \leq 1, \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \leq 1, \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \leq 1, \sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \leq 1. \quad (10.1q)$$

References

1. Alguacil, N., Motto, A.L., Conejo, A.J.: Transmission expansion planning: a mixed-integer LP approach. *IEEE Trans. Power Syst.* **18**(3), 1070–1077 (2003)
2. Balas, E.: An additive algorithm for solving linear programs with zero-one variables. *Oper. Res.* **13**(4), 517–546 (1965)
3. Bard, J.F.: Short-term scheduling of thermal-electric generators using Lagrangian relaxation. *Oper. Res.* **36**(5), 756–766 (1988)
4. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W., Vance, P.H.: Branch-and-price: column generation for solving huge integer programs. *Oper. Res.* **46**(3), 316–329 (1998)
5. Birge, J.R.: Aggregation bounds in stochastic linear programming. *Math. Program.* **31**(1), 25–41 (1985)
6. Birge, J.R., Louveaux, F.: *Introduction to Stochastic Programming*. Springer, Berlin (2011)
7. Clay, R.L., Grossmann, I.E.: A disaggregation algorithm for the optimization of stochastic planning models. *Comput. Chem. Eng.* **21**(7), 751–774 (1997)
8. Crawley, D.B., Lawrie, L.K., Winkelmann, F.C., et al.: EnergyPlus: creating a new-generation building energy simulation program. *Energy Build.* **33**(4), 319–331 (2001)
9. Crawley, D.B., Pedersen, C.O., Lawrie, L.K., Winkelmann, F.C.: EnergyPlus: energy simulation program. *ASHRAE J.* **42**, 49–56 (2000)
10. Edmunds, T.A., Bard, J.F.: An algorithm for the mixed-integer nonlinear bilevel programming problem. *Ann. Oper. Res.* **34**(1), 149–162 (1992)
11. Elhallaoui, I., Villeneuve, D., Soumis, F., Desaulniers, G.: Dynamic aggregation of set-partitioning constraints in column generation. *Oper. Res.* **53**(4), 632–645 (2005)
12. Energy Information Administration: Average retail price of electricity, monthly. <http://www.eia.gov/electricity/data/browser/#/topic/?agg=0,1&geo=vvvvvvvvvvvo&endsec=vg&linechart=~ELEC.PRICE.IL-ALL.M&columnchart=ELEC.PRICE.US-ALL.M&map=ELEC.PRICE.US-ALL.M&freq=M&start=200101&end=201402&ctype=linechart<ype=pin&rtype=s&maptype=0&rse=0&pin=> (2015)
13. Falgout, R.D.: An introduction to algebraic multigrid. *Comput. Sci. Eng.* **8**(6), 24–33 (2006)
14. Fisher, M.L.: The Lagrangian relaxation method for solving integer programming problems. *Manag. Sci.* **27**(1), 1–18 (1981)
15. Fourer, R., Gay, D.M., Kernighan, B.W.: *AMPL: A Modeling Language for Mathematical Programming*. Cengage Learning, Boston (2012)

16. Gelman, E., Mandel, J.: On multilevel iterative methods for optimization problems. *Math. Program.* **48**(1–3), 1–17 (1990)
17. Geoffrion, A.M.: An improved implicit enumeration approach for integer programming. *Oper. Res.* **17**(3), 437–454 (1969)
18. Geoffrion, A.M.: *Lagrangian Relaxation for Integer Programming*. Springer, Berlin (1974)
19. Glover, F.: Surrogate constraints. *Oper. Res.* **16**(4), 741–749 (1968)
20. Glover, F.: Heuristics for integer programming using surrogate constraints. *Decis. Sci.* **8**(1), 156–166 (1977)
21. Gratton, S., Sartenaer, A., Toint, P.L.: Recursive trust-region methods for multiscale nonlinear optimization. *SIAM J. Optim.* **19**(1), 414–444 (2008)
22. Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. *SIAM J. Sci. Stat. Comput.* **13**(5), 1194–1217 (1992)
23. Hedman, K.W., Ferris, M.C., O’Neill, R.P., Fisher, E.B., Oren, S.S.: Co-optimization of generation unit commitment and transmission switching with N-1 reliability. *IEEE Trans. Power Syst.* **25**(2), 1052–1063 (2010)
24. Jain, A.K.: Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **31**(8), 651–666 (2010)
25. Mathews, G.B.: On the partition of numbers. *Proc. Lond. Math. Soc.* **1**(1), 486–490 (1896)
26. Moore, J.T., Bard, J.F.: The mixed integer linear bilevel programming problem. *Oper. Res.* **38**(5), 911–921 (1990)
27. Pinar, A., Meza, J., Donde, V., Lesieutre, B.: Optimization strategies for the vulnerability analysis of the electric power grid. *SIAM J. Optim.* **20**(4), 1786–1810 (2010)
28. Pruitt, K.A., Braun, R.J., Newman, A.M.: Evaluating shortfalls in mixed-integer programming approaches for the optimal design and dispatch of distributed generation systems. *Appl. Energy* **102**, 386–398 (2013)
29. Pruitt, K.A., Leyffer, S., Newman, A.M., Braun, R.J.: A mixed-integer nonlinear program for the optimal design and dispatch of distributed generation systems. *Optim. Eng.* **15**, 167–197 (2014)
30. Ren, H., Gao, W.: A MILP model for integrated plan and evaluation of distributed energy systems. *Appl. Energy* **87**(3), 1001–1014 (2010)
31. Rogers, D.F., Plante, R.D., Wong, R.T., Evans, J.R.: Aggregation and disaggregation techniques and methodology in optimization. *Oper. Res.* **39**(4), 553–582 (1991)
32. Scaparra, M.P., Church, R.L.: A bilevel mixed-integer program for critical infrastructure protection planning. *Comput. Oper. Res.* **35**(6), 1905–1923 (2008)
33. Siddiqui, A.S., Marnay, C., Bailey, O., Hamachi LaCommare, K.: Optimal selection of on-site generation with combined heat and power applications. *Int. J. Distrib. Energy Res.* **1**(1), 33–62 (2005)
34. Stadler, M., Marnay, C., Siddiqui, A., Lai, J., Coffey, B., Aki, H.: Effect of heat and electricity storage and reliability on microgrid viability: A study of commercial buildings in California and New York states. Lawrence Berkeley National Laboratory Technical Report LBNL-1334E (2009)
35. Vicente, L.N., Calamai, P.H.: Bilevel and multilevel programming: a bibliography review. *J. Glob. Optim.* **5**(3), 291–306 (1994)
36. Wen, U.P., Hsu, S.T.: Linear bi-level programming problems—a review. *J. Oper. Res. Soc.* **42**(2), 125–133 (1991)
37. Wen, U.P., Huang, A.D.: A simple tabu search method to solve the mixed-integer linear bilevel programming problem. *Eur. J. Oper. Res.* **88**(3), 563–571 (1996)
38. Wen, U.P., Yang, Y.H.: Algorithms for solving the mixed integer two-level linear programming problem. *Comput. Oper. Res.* **17**(2), 133–142 (1990)
39. Wen, Z., Goldfarb, D.: A line search multigrid method for large-scale nonlinear optimization. *SIAM J. Optim.* **20**(3), 1478–1503 (2009)
40. Zipkin, P.H.: Bounds for row-aggregation in linear programming. *Oper. Res.* **28**(4), 903–916 (1980)
41. Zipkin, P.H.: Bounds on the effect of aggregating variables in linear programs. *Oper. Res.* **28**(2), 403–418 (1980)