Distributed Scheme for Unit Commitment Problem Using Constraint Programming and ADMM

Yuta INOUE†, Nonmember and Toshiyuki MIYAMOTO††, Senior Member

SUMMARY The unit commitment problem (UCP) is the problem of deciding up/down and generation-level patterns of energy production units. Due to the expansion of distributed energy resources and the liberalization of energy trading in recent years, solving the distributed UCP (DUCP) is attracting the attention of researchers. Once an up/down pattern is determined, the generation-level pattern can be decided distributively using the alternating direction method of multipliers (ADMM). However, ADMM does not guarantee convergence when deciding both up/down and generation-level patterns. In this paper, we propose a method to solve the DUCP using ADMM and constraint optimization programming. Numerical experiments show the efficacy of the proposed method.

key words: Distributed energy management system, Distributed unit commitment, Constraint optimization problem, ADMM

1. Introduction

In recent years, the liberalization of energy trading (e.g., deregulation of the electricity market) has progressed. In the future, further liberalization, such as energy trading between consumers, is expected. An energy management system (EMS) optimizes the energy supply plan for energy consumers. In this paper, we consider the situation where multiple EMSs exist and it is possible to exchange surplus energy with other EMSs. We call such a system a distributed energy management system (DEMS) [1], in which multiple EMSs coordinate to achieve optimization for the entire EMS group.

The unit commitment problem (UCP) is the problem of deciding the up/down and generation-level patterns of energy production units. The UCP is formulated as a problem that minimizes cost while matching the supply and demand of energy under the operational constraints of the units. The UCP is known as an NP-hard problem, and many approaches have been studied [2]. Classically, the UCP was a problem of planning the operation of generators owned by a general electricity utility. However, because of the reform of the electricity system, the UCP must be solved in a distributed manner.

In recent years, optimization in multi-agent systems has been actively studied, especially in the field of mathematical programming. Various methods have been proposed, such as distributed subgradient methods [3], dual decomposition, and the alternating direction method of multipliers (ADMM) [4]. Since these methods are for convex optimization problems, they cannot be directly applied to a distributed UCP (DUCP), which is a non-convex optimization problem that includes discrete variables. If we apply these methods directly to the DUCP, the method may not converge because the up/down pattern may not stabilize. Once an up/down pattern is determined, the DUCP becomes a convex optimization problem, which can be solved by ADMM. The method proposed by Miyamoto et al. [1] prioritizes units using the relaxed solution and gradually increases the number of up-units. Zhao et al. [5] proposed applying the subgradient method to search the up/down pattern of units. Feizollahi et al. [6] and Wang et al. [7] proposed a two-stage ADMM called release and fix (R&F). Kargarian et al. [8] approached the DUCP using analytical target cascading (ATC), but did not discuss convergence.

Constraint optimization programming (COP) [9] is another optimization paradigm that differs from mathematical programming. COP defines domains and constraints of discrete variables, and it repeats a search and constraint propagation to obtain a solution. Neither mathematical programming nor COP is superior. They both have strengths and weaknesses in the target problem. In recent years, distributed methods in COP have been studied [10], [11].

In this paper, we propose a method combining ADMM and COP to solve the DUCP. The proposed method obtains a candidate set of up/down patterns from ADMM iterations and determines an up/down pattern by COP. After fixing the up/down pattern, the method executes ADMM to obtain a generation-level pattern. As far as we know, no method exists that combines ADMM and COP. Although the proposed method does not guarantee the optimality of the solution, it is expected to efficiently obtain a suboptimal solution by finding an up/down pattern from the candidate set.

The rest of this paper is organized as follows: in Sect. 2, we outline a DEMS, formulate the optimization problem of a DEMS, and discuss optimization using ADMM. Section 3 proposes a method using ADMM and COP for DEMS optimization. Then, after evaluating the proposed method by numerical experiment in Sect. 4, the conclusion is stated in Sect. 5.

2. Distributed energy management system

2.1 Overview

We consider a DEMS in a given region, which is called a group, in which energy can be traded between EMSs. An
The set of units possessed by EMS 

\( i \) let 

\( M \)

\( U \)

\( I \)

2.2 The UCP in a DEMS

A predicted energy demand is given to each EMS. Each EMS, possessing energy conversion units such as gas turbines and boilers, can produce energy. Each EMS decides a purchase plan for the outside or local markets and an operation plan for energy conversion units to meet its demand.

The objective of an EMS is to minimize the cost of procuring energy, while constraints including energy balance, upper and lower limits, and minimum up or down time of units are satisfied.

2.3 ADMM for exchange problem

The problem involving Eq. (3) is called the exchange problem. Assuming that \( X_i, i \in I \) is a convex set and \( f_i, i \in I \) is a convex function, the optimal solution can be obtained by using ADMM [4]. Let \( k \) be the iteration number, then ADMM executes an iterative calculation as follows:

\[
\begin{align*}
    x_i^{(k+1)} & := \arg\min_{x_i \in X_i} \left( f_i(x_i) + \alpha^T x_i^M \right) \\
    & \quad - \frac{\rho}{2}\|x_i^M - x_i^{M(k)} + x(k)^2\|_2^2, \quad i \in I, \\
    \alpha^{(k+1)} & := \alpha^{(k)} + \rho x^{(k+1)}.
\end{align*}
\]

where \( x(k) = \sum_{i \in I} x_i^{M(k)}/|I| \), and \( \rho \) is a constant called the penalty parameter. This algorithm is a distributed method since each EMS executes Eq. (4). \( \alpha \in \mathbb{R}^{|M|} \) is the dual-variable vector, which corresponds to energy prices in local markets. Equation (5) means that the price is raised when there is surplus demand in local markets, and conversely, the price is lowered when there is surplus supply.

A convergence check of ADMM uses the primal residual and dual residuals of the problem. The primal residual of the exchange problem is given by the following equation:

\[
r^{(k)} = \sum_{i \in I} x_i^{M(k)}. \tag{6}
\]

Let \( x^{M(k)} \) be the vector consisting of all \( x_i^{M(k)}, i \in I \), then the dual residual is as follows:

\[
s^{(k)} = \rho (x^{M(k)} - x^{M(k-1)}). \tag{7}
\]

The tolerance of primal and dual residuals are denoted by \( \varepsilon^r \) and \( \varepsilon^s \), respectively. Then, the condition of the convergence check is as follows:

\[
\|s^{(k)}\|_2 < \varepsilon^r, \tag{8}
\]

\[
\|s^{(k)}\|_2 < \varepsilon^s. \tag{9}
\]

2.4 Motivation

Since the DUCP contains binary variables \( x^U \), \( X_i \) is not a convex set. Therefore, ADMM does not guarantee its optimality or convergence. Figure 2 and Table 1 show the results of a numerical experiment. Figure 2 shows the change of primal residuals in an electricity and a heat market during ADMM iterations. The primal residual of heat is zero. As
such, the supply and demand in the local market is balanced. On the other hand, the electricity market is not balanced, and it oscillates. Because Eq. (8) must be satisfied, the algorithm does not achieve convergence.

Table 1 shows the change in up/down patterns during ADMM iterations. As Table 1 shows, binary variables are not determined; this is the cause of non-convergence.

If the up/down pattern is fixed, the DUCP becomes a convex optimization problem. In that case, ADMM converges. Therefore, if we can find a suboptimal up/down pattern in advance, we may obtain a suboptimal solution, albeit not an optimal one. However, there are not many pattern candidates at 

\[ \sum_{t=1}^{T} c_t \] 

Therefore, in Step 8, candidate set \( C_t \) is obtained from ADMM iterations in the proposed method. However, in order to reduce the size of \( C_t \), not all up/down patterns should be included in \( C_t \). After sufficient iterations, up/down patterns are added into \( C_t \) during the subsequent \( \kappa^D \) iterations. \( \kappa^S \) represents the number of sufficient iterations. However, it is difficult to set \( \kappa^S \) in advance. Consequently, we use \( b_t \) to decide \( \kappa^S \). \( b_t \) is set to 1 when EMS \( i \) judges that ADMM has been sufficiently iterated. \( \kappa^S \) is set to \( k \) when \( b_t \) becomes 1 for all EMSs, and up/down patterns are added into \( C_t \) from \( \kappa^S + 1 \) to \( \kappa^S + \kappa^D - 1 \). From the candidate set \( C_t \), one up/down pattern is chosen by COP. From Hypothesis 1, it is expected that COP can be solved in a practical amount of time. Then, ADMM is executed again under the fixed up/down pattern.

In Step 1 of the algorithm, \( x_t \) and \( a_t \) are updated by Eqs. (4) and (5). Then, in Step 2, the convergence is checked using Eqs. (8) and (9).

In Step 4, function approximation is performed, while in Step 5, \( b_t \) is updated using the approximation function. The details are described in Sect. 3.2.1. When \( b_t = 1 \) in all EMSs, \( k \) at that time is substituted to \( \kappa^S \).

Step 1 to Step 7 are repeated until \( k > \kappa^S \). When \( k > \kappa^S \), \( x_{i}(k) \) is added into \( C_t \). However, some up/down patterns are infeasible, where we call an up/down pattern feasible if a solution exists under the up/down pattern. Therefore, in Step 8, candidate set \( C_t \) is updated so that any up/down pattern in \( C_t \) is feasible. The details of \( C_t \) are

**3. Proposed method**

3.1 Overall procedure

The flowchart of the proposed method is shown in Fig. 3.

In this algorithm, \( \kappa^S \) is a variable, \( \kappa^D \) is a positive constant, \( b_t \) is a binary variable of EMS \( i \), and \( C_t \) is the set of up/down pattern candidates at \( t \in T \), and \( K_{c_t} \) are the set of iteration numbers when pattern \( c_t \) is added into \( C_t \). \( \kappa^S \) is initialized to a large number; \( C_t \) and \( K_{c_t} \) are initialized to \( \emptyset \).

From Hypothesis 2, candidate set \( C_t \) is obtained from ADMM iterations in the proposed method. However, in order to reduce the size of \( C_t \), not all up/down patterns should be included in \( C_t \). After sufficient iterations, up/down patterns are added into \( C_t \) during the subsequent \( \kappa^D \) iterations. \( \kappa^S \) represents the number of sufficient iterations. However, it is difficult to set \( \kappa^S \) in advance. Consequently, we use \( b_t \) to decide \( \kappa^S \). \( b_t \) is set to 1 when EMS \( i \) judges that ADMM has been sufficiently iterated. \( \kappa^S \) is set to \( k \) when \( b_t \) becomes 1 for all EMSs, and up/down patterns are added into \( C_t \) from \( \kappa^S + 1 \) to \( \kappa^S + \kappa^D - 1 \). From the candidate set \( C_t \), one up/down pattern is chosen by COP. From Hypothesis 1, it is expected that COP can be solved in a practical amount of time. Then, ADMM is executed again under the fixed up/down pattern.

In Step 1 of the algorithm, \( x_t \) and \( a_t \) are updated by Eqs. (4) and (5). Then, in Step 2, the convergence is checked using Eqs. (8) and (9).

In Step 4, function approximation is performed, while in Step 5, \( b_t \) is updated using the approximation function. The details are described in Sect. 3.2.1. When \( b_t = 1 \) in all EMSs, \( k \) at that time is substituted to \( \kappa^S \).

Step 1 to Step 7 are repeated until \( k > \kappa^S \). When \( k > \kappa^S \), \( x_{i}(k) \) is added into \( C_t \). However, some up/down patterns are infeasible, where we call an up/down pattern feasible if a solution exists under the up/down pattern. Therefore, in Step 8, candidate set \( C_t \) is updated so that any up/down pattern in \( C_t \) is feasible. The details of \( C_t \) are
In Sect. 3.2.2, the proposed method starts to add up/down patterns into \( C_t \) that is the fuel cost at time \( t \) and iteration \( k \) is recorded, and the number of iterations \( k \) is added to \( K_{c_t} \). Using cost \( f_i^{FC(k)} \) and set \( K_{c_t} \), an estimated output cost is calculated, which is used in COP in Sect. 10. More details on the estimated output cost are given in Sect. 3.3.1.

When \( k = k^S + k^D \), an up/down pattern is determined by COP in Sect. 10. The COP model is described in Sect. 3.3.2.

In Sect. 11, the optimization problem in which its up/down pattern is set to that obtained in Sect. 10 is solved by ADMM. This is described in Sect. 3.4.

### 3.2 Constructing solution candidates

#### 3.2.1 Starting point to construct solution candidates

The proposed method starts to add up/down patterns into \( C_t \) after \( k^S \) iterations. However, it is difficult to set \( k^S \) in advance. Figure 4 represents a typical change in the objective value of an EMS during ADMM iterations. The objective value is large in the early stage, but then gradually decreases\(^\dagger\). After some iterations, it starts to oscillate around a final value, and the number of iterations increases. To make the Hypothesis 1 hold, a value of approximately 50 seems to be appropriate for \( k^S \) in this case. The proposed method uses the phenomenon in the change of objective values; i.e., when the objective value is close to the final value, adding up/down patterns into \( C_t \) is started. The proposed method uses the function approximation to detect the starting point because it requires only one parameter (a tolerance \( \varepsilon^b \)) to be chosen\(^\dagger\).

We propose to approximate the objective change using an exponential function because it is simple and the approximate function must have a steady value that represents the optimal objective value. Let \( K = \{1, \ldots, k\} \), and set the approximation function \( F_i^{k}(x) \) at iteration \( k \) as follows:

\[
F_i^{k}(x) = w_{i,1}^{(k)} \exp\left(-w_{i,2}^{(k)} x\right) + w_{i,3}^{(k)},
\]

where \( w_{i,1}^{(k)} \in \mathbb{R} \), and \( w_{i,2}^{(k)}, w_{i,3}^{(k)} \in \mathbb{R}^+ \) are fitting parameters.

\(^\dagger\)In another case, the objective value is small in the early stage, but then gradually increases.

\(^\dagger\dagger\)We considered using the moving average method. However, it requires two parameters: the number of points and the tolerance.

At Step 4, EMS \( i \) finds the fitting parameters \( w_{i,1}^{(k)} \), \( w_{i,2}^{(k)} \), and \( w_{i,3}^{(k)} \) from points \( (1, f_i(x_i^{(1)})), (2, f_i(x_i^{(2)})), \ldots, (k, f_i(x_i^{(k)})) \), where \( f_i(x_i^{(k)}) \) is the objective function value at \( k \)-th iteration.

In Step 5, \( b_i \) is updated as follows:

\[
b_i^{(k)} = \begin{cases} 1 & \text{if } b_i^{(k-1)} = 1 \lor \frac{\|f_i(x_i^{(k)}) - w_i^{(k)}\|_2}{w_i^{(k)}} \leq \varepsilon^b, \\ 0 & \text{otherwise} \end{cases}
\]

where \( \varepsilon^b \) represents a tolerance.

In Steps 6 and 7, when \( b_i^{(k)} = 1 \) in all EMSs, \( k^S \) is set to \( k \).

#### 3.2.2 Feasibility of up/down pattern

The proposed method selects an up/down pattern from \( C_t \) and then decides a generation-level pattern. However, it is difficult to set \( C_t \) in advance.

In another case, the objective value is small in the early stage, and then decides a generation-level pattern. Therefore, every candidate in \( C_t \) must be feasible. However, it is difficult to set \( C_t \) in advance.

Considering the above, we provide a sufficient condition for up/down patterns to be feasible. We assume that each EMS is enabled to dispose of excess energy. Under this assumption, an up/down pattern is infeasible when there is a shortage of energy supply.

Let \( L \subset G \) be the set of the energy that cannot be procured from the outside, and \( D_{g,t} \) is a demand of energy \( g \in L \) at time \( t \). The amount of energy \( g \) that EMS \( i \) produces is denoted by \( x_{i,g,t}^U \in \mathbb{R} \).

For EMS \( i \), the vector consisting of all \( x_{i,j,t}^U \), \( j \in U_t \), is denoted by \( x_{i,t}^U \in \mathbb{R}^{|U_t|} \), and the maximum production of energy \( g \) for pattern \( x_{i,t}^U \) is given by a function \( P_{i,g,t} : \mathbb{R}^{|U_t|} \mapsto \mathbb{R} \). Suppose that \( x_{i,t}^U \) satisfies local constraints in each EMS, then \( x_{i,t}^U \) is feasible if and only if

\[
\sum_{i \in \mathcal{I}} D_{i,g,t} \leq \sum_{i \in \mathcal{I}} P_{i,g,t}(x_{i,t}^U)
\]

holds. Then, the following lemma holds.

**Lemma 1.** Let \( x_{g,t}^{M(k)} \) be an element of \( \tilde{x}^{(k)} \) with regard to \( m_{g,t} \). If \( x_{g,t}^{M(k)} \leq 0 \) for all \( g \in L \), then the up/down pattern \( x_{i,t}^{U(k)} \) at \( k \)-th iteration is feasible.

**Proof.** At iteration \( k \), the following equation holds:

\[
x_{i,g,t}^{M(k)} = D_{i,g,t} - x_{i,g,t}^{P(k)}.
\]

Then, the following equation holds:

\[
x_{g,t}^{M(k)} = \frac{1}{|\mathcal{L}|} \sum_{i \in \mathcal{L}} x_{i,g,t}^{M(k)} = \frac{1}{|\mathcal{L}|} \left( \sum_{i \in \mathcal{L}} D_{i,g,t} - \sum_{i \in \mathcal{L}} x_{i,g,t}^{P(k)} \right).
\]
The following inequality holds for the energy production:
\[ x_{i,g,t}^{B(k)} \leq P_{i,g,t}(x_{i,t}^{U(k)}). \] (16)

If \( M_{g,t}(k) \leq 0 \), Eqs. (15) and (16) lead to the following inequality:
\[ \sum_{i \in I} D_{i,g,t} \leq \sum_{i \in I} x_{i,g,t}^{B(k)} \leq \sum_{i \in I} P_{i,g,t}(x_{i,t}^{U(k)}). \] (17)

Therefore, \( x_{i,t}^{U(k)} \) is feasible. \( \Box \)

From Lemma 1, we can obtain the update formula of the candidate set \( C_t \) as follows:
\[ C_t^{(k)} = \left\{ C_t^{(k-1)} \cup \left\{ x_{i,t}^{U(k)} \right\} \right\} \text{ if } \forall g \in L : x_{g,t}^{M(k)} \leq 0 \]
\[ \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \text{ otherwise} \]. (18)

3.3 COP for deciding an up/down pattern

We propose a constraint optimization model to select an up/down pattern from the candidate set \( C_t \) constructed in Step 8. The objective function in the DUCP is the sum of the start-up cost and the fuel cost. However, the fuel cost has not been determined at Step 10. Therefore, we propose to use an estimated fuel cost for the constraint optimization.

3.3.1 Estimated fuel cost

We propose to use the fuel cost obtained during ADMM iterations to obtain the estimated fuel cost for the up/down pattern.

Let \( f_{i,t}^{FC(k)} \) be the fuel cost of EMS \( i \) at time \( t \) at the \( k \)-th iteration. The average fuel cost for the entire DEMS is denoted by \( \bar{T}_t = \sum_{i \in I} f_{i,t}^{FC(k)}/|I| \). Let \( K_c \subseteq \{ k^5 + 1, \ldots, k^5 + k^D \} \), and \( K_c \) be the set of the iteration numbers when the up/down pattern is \( c_t \in C_t \), i.e.,
\[ K_c = \{ k \in K_c \mid x_{i,t}^{U(k)} = c_t \}, \quad c_t \in C_t, t \in T. \] (19)

Then, the function \( FC : \mathbb{R}^{|U|} \rightarrow \mathbb{R} \) gives an estimated fuel cost of the up/down pattern at time \( t \):
\[ FC(c_t) := \frac{|I|}{|K_c|} \sum_{k \in K_c} f_{i,t}^{FC(k)}. \] (20)

3.3.2 COP formulation

Let \( UC_j \) be the start-up cost of unit \( j \in U \), and \( FC \) be the estimated fuel cost. When a unit begins operating, it must continuously be in operation for a certain period (minimum up time). When a unit stops operating, it must be continuously suspended for a certain period (minimum down time). The minimum up and down times of unit \( j \) are denoted by \( T^u_j \) and \( T^d_j \), respectively. Let \( A_{j,t} \) be the time duration of the unit’s up/down state at time \( t \). The decision variable \( y_{i,j,t}^U \in \mathbb{R} \) represents the up/down state of unit \( j \in U \) at time \( t \). The vector consisting of \( y_{i,j,t}^U \), \( j \in U, i \in I \) is denoted by \( y_{i,t}^U \). The COP model for deciding the up/down pattern in a DEMS is formulated as follows:
\[ \min \sum_{i \in I} \left[ FC(y_{i,t}^U) + \sum_{j \in U} UC_j y_{i,j,t} \right], \] (21)
\[ \text{s.t. } y_{i,t}^U \in \mathbb{R}, \quad t \in T, \] (22)
\[ y_{i,j,t}^U - y_{i,j,t-1}^U = -1 \Rightarrow A_{j,t} \geq T^u_j, \quad j \in U, t \in T \setminus \{1\}, \] (23)
\[ y_{i,j,t}^U - y_{i,j,t-1}^U = 1 \Rightarrow A_{j,t} \geq T^d_j, \quad j \in U, t \in T \setminus \{1\}, \] (24)
\[ y_{i,j,t}^U = y_{i,j,t-1}^U \Rightarrow A_{j,t} = A_{j,t-1} + 1, \quad j \in U, t \in T \setminus \{1\}, \] (25)
\[ y_{i,j,t}^U \neq y_{i,j,t-1}^U \Rightarrow A_{j,t} = 1, \quad j \in U, t \in T \setminus \{1\}, \] (26)
\[ y_{i,j,t}^U - y_{i,j,t-1}^U = 1 \Rightarrow UC_j = UC_j, \quad j \in U, t \in T \setminus \{1\}, \] (27)
\[ y_{i,j,t}^U - y_{i,j,t-1}^U \neq 1 \Rightarrow UC_j = 0, \quad j \in U, t \in T \setminus \{1\}. \] (28)

Equation (22) prohibits the up/down patterns that are not in \( C_t \). Equations (23) and (24) impose the minimum up or down time on units. These constraints use the time duration \( A_{j,t} \) defined by Eqs. (25) and (26). The constraints regarding the start-up cost of units are represented by Eqs. (27) and (28).

Hereinafter a solution that satisfies Eqs. (22)–(28) is called a CSP\textsuperscript{D} solution. In CSP solutions, the solution that minimizes the objective function (21) is called the COP solution.

3.4 ADMM for deciding a generation-level pattern

After solving COP, ADMM is executed again for the problem where the up/down pattern is fixed to the COP solution. Let \( y_{i,t}^{U*} \) be the COP solution related to unit \( j \in U \). Then, the ADMM update formula for the problem where an up/down pattern is fixed to the COP solution is as follows:
\[ x_{i,t}^{(k+1)} := \arg\min_{x_{i,t}} \left\{ f_i(x_{i,t}) + \alpha^T x_{i,t}^M + \frac{\rho}{2} \| x_{i,t}^M - x_{i,t} \|_2^2 \right\}, \] (29)
\[ \text{s.t. } x_{i,t} = y_{i,t}^{U*} \]

The dual variables are updated by Eq. (5), and the condition of the convergence check is the same as Eqs. (8) and (9).

3.5 Termination property

The proposed method has a better termination property than \(^{1}\text{CSP is the abbreviation of constraint satisfaction programming.}^{1}\)
the normal ADMM. In fact, the proposed method terminated in success in all cases of the experiments in Sect. 4. However, there exist some conditions.

**Condition 1.** The tolerance $\varepsilon^b$ is sufficiently large.

If the tolerance is small, the condition of Eq. (12) is not satisfied and adding up/down patterns into $C_t$ is not started. Because the objective function value is finite, there exists a finite tolerance.

**Condition 2.** Each EMS is enabled to dispose of excess energy.

The condition is introduced in 3.2.2 and is necessary so that Lemma 1 holds.

**Condition 3.** At least one up/down pattern is found for each $t \in T$.

The proposed method adds up/down patterns into $C_t$ in $k^S$ iterations from the $k^S$-th iteration. If $C_t$ is empty for some $t$, COP is infeasible. Then, the proposed method terminates in failure. We could not find such a case in the experiments; however, it is difficult to prove the non-existence of such a case.

**Condition 4.** ADMM converges when the up/down pattern is fixed.

This condition is necessary so that the iterations of Steps 11 and 12 terminate. If the DUCP becomes a continuous problem when the up/down pattern is fixed, then theoretical results for convergence of ADMM [4] can be used.

Then, we can state the following theorem.

**Theorem 1.** If conditions 1–4 hold, the proposed method terminates in success.

### 3.6 Distributed execution

The proposed method is executed in a distributed manner by the agent corresponding to each EMS and the coordinator agent corresponding to the local market. Here, we describe the processes of each agent at each step shown in Fig.3.

The local market initializes $\alpha^{(0)}$ and $\overline{\alpha}^{(0)}$, and sets $k^S = \infty$. EMS $i$ initializes $x_i^{(0)}$. In addition, all agents set $k = 0$. Then, the local market sends the values of $\alpha^{(0)}$ and $\overline{\alpha}^{(0)}$ to each EMS.

At Step 1, EMS $i$ updates $x_i$ using $\alpha$ and $\overline{\alpha}$ received from the local market, and sends $x_i^M$ in return. Then, the local market updates $\alpha$ using $x_i^M$ received from all EMSs. All agents increment $k$ with each update. At Step 2, the local market checks convergence using $x_i^M$. If conditions (8) and (9) hold, the local market broadcasts the end of the algorithm. If the convergence check is false, the local market goes to Step 3. If $k > k^S$, the local market requests all EMSs to send an up/down pattern $x_i^U$ and fuel cost $f_i^{\text{FC}}$. If $k \leq k^S$, the local market sends a command to all EMSs to execute parameter fitting.

EMS $i$ fits parameters at Step 4, and updates $b_i$, sending its value to the local market at Step 5. The local market checks the condition at Step 6. When the condition holds, the local market substitutes $k$ to $k^S$ at Step 7.

If EMS $i$ receives a request to send its up/down pattern and fuel cost, EMS $i$ sends $x_i^U$ and $f_i^{\text{FC}}$ to the local market at Step 8. The local market calculates $f_i^{\text{FC}}(k)$ and records the result, and updates $C_t$ and $K_t$, at Step 8.

If $k \geq k^S + k^D$, the local market solves COP at Step 10. Then, the local market sends solution $b_i^{\text{UP}}$ to all EMSs. In this paper, COP is solved by the local market. For security reasons, however, this is not ideal. The distributed COP (DCOP) framework [10], [11] can be used by EMS agents to solve COP.

At Steps 11 and 12, the local market and EMSs run ADMM similar to Steps 1 and 2.

### 4. Computational experiments

#### 4.1 Experimental conditions

We implemented the proposed method using Java SE 11 and Python 3. The optimization of Eqs. (4) and (29) is done by IBM CPLEX 12.9, and the determination of an up/down pattern of Eqs. (21)–(28) is done by IBM CP Optimizer 12.9.

#### 4.1.1 The DEMS model

We consider a group in which electricity and heat can be traded in the local market. Each EMS purchases electricity and gas from outside, and heat and electricity from the local market. Moreover, each EMS possesses either one or both of a gas turbine (GT) and a gas boiler (BA) as an energy conversion unit. A gas turbine converts gas to electricity and heat, while a boiler converts gas to heat. Each EMS makes an energy supply and demand plan for a period $T$. In this experiment, the DEMS model is composed of two types of optimization models: a supplier model and a consumer model.

Figure 5 shows a supplier model. A supplier purchases electricity or gas from outside and produces both electricity and heat with a gas turbine and a gas boiler. Then, it sells the surplus energy to consumers in the local market.

Let $a_{\text{BE}}$, $a_{\text{BG}} \in \mathbb{R}^{|T|}$ be the constant vectors representing the price of outside electricity and gas, respectively.
The price of electricity and heat in the local market is denoted by \( a_E, a_H \in \mathbb{R}^{|T|} \), respectively. For these energy prices, each EMS determines the \(|T|\) dimensional vectors \( BE, BG \in \mathbb{R}^{|T|} \), \( SE, SH \) representing the amount of electricity and gas purchased from outside, and that of electricity and heat sold in the local market, respectively. Surplus electricity and heat that is not supplied to the local market is treated as waste \( WE \) and \( WH \in \mathbb{R}^{|T|} \). Within \( BG \), the amount of input to each unit (\( \bullet \)) is denoted by \( BG_{\bullet} \) and the element corresponding to time \( t \in T \) of \( BG_{\bullet} \) is denoted by \( BG_{\bullet t} \). The maximum and minimum inputs of each unit are denoted by \( B_{\bullet}^+ \) and \( B_{\bullet}^- \), respectively. The minimum up and down time of each unit are denoted by \( T_{\bullet u} \) and \( T_{\bullet d} \), respectively. In addition, the time duration of the unit’s state at time \( t \) is denoted by \( A_{\bullet t} \). To start a unit, it costs \( UC_{\bullet} \); let \( UC_{\bullet t} \) be the start-up cost incurred at time \( t \). The input-output characteristic of each unit is given as a piecewise linear function \( \Gamma_u(BG_{\bullet}) \), with the corresponding \( PE_{\bullet} \) and \( PH_{\bullet} \) representing the output amount of electricity and heat, respectively. The up/down state of each unit is denoted by \( x_{\bullet t} \in \{0,1\} \). Then, the optimization problem of the supplier model is as follows:

\[
\begin{align*}
\text{min} & \quad a_{BE}^T BE + a_{BG}^T BG - a_E^T SE - a_H^T WH + \sum_{t \in T} \{ UC_{GT,t} + UC_{BA,t} \}, \\
\text{s.t.} & \quad PE_{GT} = \Gamma_{GT}\left(BG_{GT}\right), \\
& \quad PH_{GT} = \Gamma_{BA}\left(BG_{BA}\right), \\
& \quad BE + PE_{GT} = DE + SE + WE, \\
& \quad PH_{GT} + PH_{BA} = DH + SH + WH, \\
& \quad BG = BG_{GT} + BG_{BA}, \\
& \quad \frac{BG_{GT} x_{BA,t}}{BG_{BA}} \leq BG_{GT,t} \leq \frac{BG_{GT} x_{BA,t}}{BG_{BA}}, \quad t \in T, \\
& \quad \frac{BG_{BA} x_{BA,t}}{BG_{BA}} \leq BG_{BA,t} \leq \frac{BG_{BA} x_{BA,t}}{BG_{BA}}, \quad t \in T, \\
& \quad x_{GT,t} - x_{GT,t-1} = 1 \Rightarrow UC_{GT,t} = UC_{GT}, \quad t \in T\backslash\{1\}, \\
& \quad x_{GT,t} - x_{GT,t-1} \neq 1 \Rightarrow UC_{GT,t} = 0, \quad t \in T\backslash\{1\}, \\
& \quad x_{BA,t} - x_{BA,t-1} = 1 \Rightarrow UC_{BA,t} = UC_{BA}, \quad t \in T\backslash\{1\}, \\
& \quad x_{BA,t} - x_{BA,t-1} \neq 1 \Rightarrow UC_{BA,t} = 0, \quad t \in T\backslash\{1\}, \\
& \quad x_{GT,t} - x_{GT,t-1} = -1 \Rightarrow A_{GT,t} \geq T_{GT}^u, \quad t \in T\backslash\{1\}, \\
& \quad x_{GT,t} - x_{GT,t-1} = 1 \Rightarrow A_{GT,t} \leq T_{GT}^d, \quad t \in T\backslash\{1\}, \\
& \quad x_{BA,t} - x_{BA,t-1} = -1 \Rightarrow A_{BA,t} \geq T_{BA}^u, \quad t \in T\backslash\{1\}, \\
& \quad x_{BA,t} - x_{BA,t-1} = 1 \Rightarrow A_{BA,t} \leq T_{BA}^d, \quad t \in T\backslash\{1\}.
\end{align*}
\]

Equations (31)–(33) are the input-output constraint of units. The energy balancing constraints are represented by Eqs. (34)–(36). Equations (37) and (38) are the constraints of the input limit, and Eqs. (39)–(42) are the constraints of the start-up cost. Equations (43)–(46) impose the minimum up or down time; they are based on time duration \( A_{\bullet t} \), which is represented by Eqs. (47)–(50).

Figure 6 shows the consumer model. The consumer possesses only a gas boiler, so it must procure electricity from the local market or the outside. It also behaves as a consumer in the local market. Let \( BE_E, BH_H \in \mathbb{R}^{|T|} \) be the amount of electricity and heat purchased from the local market, respectively. Then, the optimization problem of the consumer model is as follows:

\[
\begin{align*}
\text{min} & \quad a_{BE}^T BE + a_{BG}^T BG + a_E^T BE + a_H^T BH \\
& \quad + \sum_{t \in T} U C_{BA,t}, \\
\text{s.t.} & \quad PH_{BA} = \Gamma_{BA}(BG), \\
& \quad BE + BE_E = DE, \\
& \quad PH_{BA} + BH_H = DH + WH, \\
& \quad \frac{BG_{BA} x_{BA,t}}{BG_{BA}} \leq BG_{BA,t} \leq \frac{BG_{BA} x_{BA,t}}{BG_{BA}}, \quad t \in T, \\
& \quad x_{BA,t} - x_{BA,t-1} = 1 \Rightarrow UC_{BA,t} = UC_{BA}, \quad t \in T\backslash\{1\}, \\
& \quad x_{BA,t} - x_{BA,t-1} \neq 1 \Rightarrow UC_{BA,t} = 0, \quad t \in T\backslash\{1\}, \\
& \quad x_{BA,t} - x_{BA,t-1} = -1 \Rightarrow A_{BA,t} \geq T_{BA}^u, \quad t \in T\backslash\{1\}, \\
& \quad x_{BA,t} - x_{BA,t-1} = 1 \Rightarrow A_{BA,t} \leq T_{BA}^d, \quad t \in T\backslash\{1\}, \\
& \quad x_{BA,t} - x_{BA,t-1} = -1 \Rightarrow A_{BA,t} \geq T_{BA}^u, \quad t \in T\backslash\{1\}, \\
& \quad x_{BA,t} - x_{BA,t-1} = 1 \Rightarrow A_{BA,t} \leq T_{BA}^d, \quad t \in T\backslash\{1\}.
\end{align*}
\]
Since the types of constraints are the same as the supplier model, we omit the details.

Here, we describe the function $\Gamma_*(x)$ as an input-output characteristic. In this experiment, we use a function that linearly interpolates the plot of three points. For an input amount of 0, the output is also 0. Let $a_*, b_*, c_*, d_*$, and $p_*$ be the constraints, then $\Gamma_*(x)$ is as follows:

$$
\Gamma_*(x) = \begin{cases} 
0 & \text{if } x = 0 \\
(a_*)x + b_* & \text{if } BG_{BA} \leq x \leq p_* \\
c_*)x + d_* & \text{if } p_* \leq x \leq BG_{BA}
\end{cases}
$$

(62)

where $\Gamma_*(x)$ is continuous at the midpoint $p_*$.

4.1.2 Composition of a DEMS

In this experiment, the scheduling horizon is one day, and $|\mathcal{I}| = 24$. We use Factory 1 (F1) and Factory 2 (F2) as the supplier, and Building (B1) and Hospital (H1) as the consumer. Then, the multiple groups are constructed by combining these EMSs. Table 2 shows groups G1–G4 used in this experiment.

Figures 7 and 8 show the demand for electricity and heat. Table 3 shows the parameters of each EMS. The energy prices of outside are $\alpha_{BE} = 40.5 \text{[10}^3\text{yen/MWh]}$ and $\alpha_{BG} = 10.0 \text{[10}^3\text{yen/100m}^2\text{]}$, representing electricity and gas, respectively. In addition, we set $\kappa^D = 200$ based on preliminary experiments, the tolerances of primal and dual residuals in the local market are $\varepsilon^p = \varepsilon^d = 0.005$, and the tolerance used in parameter fitting is $\varepsilon^b = 0.01$.

4.2 Results and discussion

4.2.1 Convergence

Figure 9 shows the change in primal residuals in each group when the proposed method is not used. Figure 10 shows the change in primal residuals in the same groups after fixing the up/down pattern in the proposed method. In all the groups, primal residuals converge toward 0. Thus, the effect on convergence by fixing the up/down pattern can be seen.

4.2.2 Dependency on penalty parameter

For convex optimization problems, the optimal value obtained by ADMM does not depend on the value of penalty parameter $\rho$. However, in the DUCP, which is a non-convex problem, it is expected that the solution will depend on $\rho$. 

\begin{table}[h]
\centering
\caption{Group compositions}
\begin{tabular}{|c|c|c|c|c|}
\hline
EMS & G1 & G2 & G3 & G4 \\
\hline
F1 & X & X & X & X \\
F2 & X & X & X & X \\
B1 & X & X & X & X \\
H1 & X & X & X & X \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Parameters of each EMS}
\begin{tabular}{|c|c|c|c|c|}
\hline
 & F1 & F2 & B1 & H1 \\
\hline
$\alpha_{BA}$ [GJ/10$^3$m$^2$] & 4.34 & 4.54 & 4.44 & 4.69 \\
$\beta_{BA}$ [GJ] & -0.76 & -0.93 & -0.44 & -0.24 \\
$\gamma_{BA}$ [GJ/10$^2$m$^3$] & 3.40 & 4.09 & 3.58 & 3.94 \\
$\delta_{BA}$ [GJ] & 0.65 & -0.56 & 0.42 & 0.66 \\
$\rho_{BA}$ [10$^2$m$^3$] & 1.50 & 0.80 & 1.00 & 1.20 \\
$BG_{BA}$ [10$^2$m$^3$] & 2.75 & 1.36 & 1.84 & 2.2 \\
$BG_{BA}$ [10$^2$m$^3$] & 0.405 & 0.23 & 0.12 & 0.33 \\
$T_{BA}$ [hour] & 1 & 1 & 1 & 1 \\
$\gamma_{BA}$ [hour] & 1 & 1 & 1 & 1 \\
$UC_{BA}$ [10$^3$yen] & 6.075 & 3.45 & 1.8 & 4.95 \\
\hline
$\alpha_{GT}$ [MWh/10$^3$m$^2$] & 0.49 & 0.47 & - & - \\
$\beta_{GT}$ [MWh] & -1.84 & -2.32 & - & - \\
$\gamma_{GT}$ [MWh/10$^3$m$^2$] & 0.45 & 0.42 & - & - \\
$\delta_{GT}$ [MWh] & -0.75 & -1.62 & - & - \\
$\rho_{GT}$ [10$^2$m$^3$] & 27 & 16 & - & - \\
$BG_{GT}$ [10$^2$m$^3$] & 0.75 & 1.15 & - & - \\
$BG_{GT}$ [10$^2$m$^3$] & -3.14 & -5.38 & - & - \\
$T_{GT}$ [hour] & 0.71 & 1.0 & - & - \\
$\gamma_{GT}$ [hour] & -2.05 & -2.92 & - & - \\
$UC_{GT}$ [10$^3$yen] & 27 & 16 & - & - \\
\hline
\end{tabular}
\end{table}
In Eq. (4), \( \rho \) is multiplied by the amount of change from the previous update solution. Therefore, as \( \rho \) becomes larger, it is expected that the update solution is more likely to be trapped at a suboptimal point. Therefore, the value of \( \rho \) would affect the exploration of ADMM and the quality of the resulting solution to the DUCP.

We conducted the experiment below to verify the dependency of the parameter in group G1. By changing the value of \( \rho \) from 0.5 to 10, the candidate set is constructed in each case. COP and CSP solutions are enumerated in each case. Then, ADMM is executed by fixing the up/down pattern to one of the COP and CSP solutions. Figure 11 shows the objective values.

As shown in Fig. 11, as the value of \( \rho \) becomes smaller, the distribution of the objective values tends to be wider. In addition, the objective value of the best solution tends to decrease. Therefore, in the subsequent experiments, we set \( \rho = 0.5 \).

### 4.2.3 Appropriateness of estimated fuel cost

In the proposed method, an up/down pattern is selected using the estimated fuel cost. We proposed using the average of the obtained fuel cost during ADMM iterations in Sect. 3.3.1. In this experiment, we checked the appropriateness of using the average cost. The triangular mark in Fig. 11 is the total cost using the average cost, while the star mark is the total cost using the minimum fuel cost of units. As shown in Fig. 11, the solution of the proposed method takes the best value or close to the best value. We saw similar results in other groups. Therefore, using the average of fuel costs obtained during ADMM iterations is appropriate.

#### 4.2.4 Effectiveness of checking feasibility of up/down pattern

In Sect. 3.2.2, we proposed a method that limits the up/down patterns to be added into the candidate set \( C^t \). This experiment compares two cases of updating \( C^t \): one of using Eq. (18) and another of adding all up/down patterns.

Table 4 shows the candidate set obtained in each case in group G4. As shown in Table 4, some up/down patterns are not added into the candidate set by Eq. (18). We ran ADMM on each combination of up/down patterns and checked convergence. When the candidate set obtained using Eq. (18) was used, ADMM converged under any combination of up/down pattern. As for the candidate set obtained without using Eq. (18), ADMM did not converge in the local market for heat at time 12 when the up/down pattern that is marked by * in Table 4 was used. Figure 12 shows the change of primal residuals in local markets at time 12 during ADMM iterations, where the up/down pattern marked by * in Table 4 was used. In this case, units do not have enough heat output to supply the local market, so we verified the necessity of the limitation by Eq. (18).

### 4.2.5 Comparison with other methods

In this experiment, we discuss the solution quality by comparing the proposed method and other methods. We used the following three methods: 1) the method fixing all units to be in operation (Allup), 2) the method in [1] (Sequp), and 3) the release and fix (R&F) method in [6]. Sequp first solves the relaxed problem where the binary variables are relaxed to continuous variables, and decides an activation priority
of the units using the information from the relaxed solution. Then, it increases the number of up units until the problem is feasible. In R&F, if the value of a binary variable remains unchanged for a certain number of iterations in ADMM, it will be fixed to that value. By repeating this, the binary variables are fixed in sequence. The threshold for the number of iterations was set to 15, as in [6].

Table 5 shows the objective values for each group. All data was normalized such that the value in Allup is equal to 100%. Figure 13 shows the up/down pattern for each method in group G2, where GT_F1 represents the gas turbine of Factory 1, for example. Since the same tendency was observed in other groups, we omit them here. Allup is obvious, so it is also omitted.

Compared with Allup, other methods have lower objective function values. This is because some units are suspended at some times, as shown in Fig. 13.

The up/down pattern resulting from the proposed method and R&F are similar. The set of down states in Sequp is a subset of the proposed method and R&F. Therefore, Sequp operates excess units, which leads to an increase in the objective value. This is because the proposed method and R&F determine an up/down pattern through exploration, while Sequp determines it by rounding the relaxation solution.

While the up/down pattern resulting from the proposed method and R&F are quite similar, the proposed method performs better on the objective value. In Fig. 13, there are differences at times 4, 6, 13, 16, and 17. Table 6 shows a comparison of the fuel costs for each time; the proposed method has a lower fuel cost than R&F. Table 7 shows up/down pattern candidates at the same times. At times 16 and 17, candidate sets include the up/down pattern that is equal to the one selected by R&F. In this case, the proposed method selects the up/down pattern that leads to a lower fuel cost using the COP solution.

4.2.6 Reasonableness of the hypotheses

In Sect. 2.4, we formed two hypotheses on the candidate set $C$ of up/down patterns. Hypothesis 1 means that we can efficiently solve the COP model in terms of computation time and space. Hypothesis 2 means that the proposed method can find a suboptimal solution by choosing an up/down pattern from $C$. Here, we evaluate the hypotheses from the results of experiments.

Hypothesis 1 held for all groups. For example, group G4 has 6 units; the number of up/down patterns for each time $t$ is $2^6$. Therefore, the total number of up/down patterns is $2^{6\times 6}$. As shown in Table 4, the number of up/down patterns added into $C_t$ was at most 2; the total number of candidate patterns is $2^{10}$.

As for Hypothesis 2, because it was difficult to evaluate the proposed method using an optimal solution, we compared it with other methods in Sect. 4.2.5. The proposed method was able to find better or comparable solutions for all groups. Therefore, we believe Hypothesis 2 held for all groups.

5. Conclusion

In this paper, we described a distributed method of solving
the UCP in a DEMS. When ADMM is directly applied to the UCP of a DEMS, its convergence is not guaranteed, and in many cases ADMM does not converge due to the oscillation of up/down patterns. We proposed a method combining ADMM and constraint optimization.

We conducted numerical experiments to verify the validity of the proposed method. We verified the convergence and the feasibility of up/down pattern candidates. The obtained up/down pattern candidates depend on the ADMM penalty parameter. If the parameter is small, the cost distribution tends to be wide, and the value of the best solution among them tends to be small. The solution using the COP solution is better than other CSP solutions. This fact supports the validity of the proposed COP formulation. Finally, we compared the proposed method with other methods. The result of the comparison experiment indicates the efficacy of the proposed method.

References


Yuta Inoue received the B.E. and M.E. degrees from Osaka University, Osaka, Japan in 2019 and 2021, respectively. His research interests include multi-agent systems.

Toshiyuki Miyamoto received his B.E. and M.E. degrees in electronic engineering from Osaka University, Japan in 1992 and 1994, respectively. Moreover, he received Dr. of Eng. degree in electrical engineering from Osaka University, Japan in 1997. From 2000 to 2001, he was a visiting researcher in Department of Electrical and Computer Engineering at Carnegie Mellon University, Pittsburgh, PA. Currently, he is an Associate Professor with the Division of Electrical, Electronic and Infocommunications Engineering, Osaka University. His areas of research interests include theory and applications of concurrent systems and multi-agent systems. He is a member of IEEE, SICE, and ISCIE.