

Article

Comparative Study on Optimization Solvers for Implementation of a Two-Stage Economic Dispatch Strategy in a Microgrid Energy Management System

Gi-Ho Lee, Jae-Young Park, Seung-Jun Ham and Young-Jin Kim *

Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Gyungbuk 37673, Korea; ghlee0111@postech.ac.kr (G.-H.L.); luckypark@postech.ac.kr (J.-Y.P.); juniel@postech.ac.kr (S.-J.H.)

* Correspondence: powersys@postech.ac.kr; Tel.: +82-54-279-2368

Received: 23 January 2020; Accepted: 26 February 2020; Published: 2 March 2020

Abstract: A microgrid energy management system (MEMS) optimally schedules the operation of dispatchable distributed energy resources to minimize the operation costs of microgrids (MGs) via an economic dispatch (ED). Actual ED implementation in the MEMS relies on an optimization software package called an optimization solver. This paper presents a comparative study of optimization solvers to investigate their suitability for ED implementation in the MEMS. Four optimization solvers, including commercial as well as open-source-based ones, were compared in terms of their computational capability and optimization results for ED. Two-stage scheduling was applied for the ED strategy, whereby a mixed-integer programming problem was solved to yield the optimal operation schedule of battery-based energy storage systems. In the first stage, the optimal schedule is identified one day before the operating day; in the second stage, the optimal schedule is updated every 5 min during actual operation to compensate for operational uncertainties. A modularized programming strategy was also introduced to allow for a comparison between the optimization solvers and efficient writing of codes. Comparative simulation case studies were conducted on three test-bed MGs to evaluate the optimization results and computation times of the compared optimization solvers.

Keywords: microgrids (MGs); microgrid energy management system (MEMS); economic dispatch (ED); optimization solver; modeling package; energy storage systems (ESSs)

1. Introduction

Microgrids (MGs) have gained much attention as an important building block for future power systems. MGs provide an effective means to accommodate the high penetration of renewable generators and minimize power transmission losses by supplying local loads using distributed energy resources (DERs). A microgrid energy management system (MEMS) is a hierarchical supervisory control system responsible for the reliable, secure, and economical operation of MGs [1,2]. For the economical operation, the MEMS optimally coordinates the operation schedules of dispatchable DERs (e.g., micro turbines, batteries, and controllable loads), such that the MG's daily operating cost is minimized while considering the forecasted load demand and renewable generation. This optimization process is referred to as economic dispatch (ED), which is performed as the secondary control in the three-level hierarchical control actions conducted by the MEMS [3]. Actual ED implementation in the MEMS relies on an optimization solver. The optimization solver is a software package developed for solving optimization problems according to pre-programmed solution algorithms. Various off-the-shelf optimization solvers, including the commercial and open-

source-based ones, are available for ED applications. Commercial optimization solvers generally have superior computational capability compared with their open-source counterparts. The ED implementation environment is then programmed either in general-purpose programming languages (GPLs), e.g., Python and Julia, or in algebraic modeling languages (AMLs). AML is a high-level programming language dedicated to optimization applications. Popular AMLs include GAMS and AMPL.

Table 1. Summary of previous studies on the implementation of economic dispatch (ED) strategies.

Ref	Optimization Algorithms ⁽¹⁾	Solver Types ⁽²⁾			Language Types ⁽³⁾		Demonstration Methods ⁽⁴⁾		
		OS	CS	Solver	AML	GPL	S	I	Test-Beds
[4]	LP	✓		GLPK		✓	✓		Grid-connected MG
[5]	CONVEX		✓	MOSEK		✓	✓		Grid-connected MG
[6]	MIP		✓	CPLEX		✓	✓		Grid-connected MG
[7]	MIP (T-S)		✓	CPLEX		✓	✓		Grid-connected MG
[8]	MIP (T-S)		✓	CPLEX		✓	✓		Grid-connected MGs
[9]	MIP (T-S)		✓	Gurobi		✓	✓		Islanded MG
[10]	MIP (T-S)		✓	SCIP		✓	✓		IEEE 33-bus radial system
[11]	MIP		✓	CPLEX	✓			✓	Budapest Tech Renewable System, Hungary
[12]	MIP		✓	CPLELX	✓			✓	Institute of Nuclear Energy Research MG, Taiwan
[13]	MIP (T-S)		✓	CPLEX	✓			✓	MG Research Lab. in Aalborg Univ., Denmark
[14]	MIP		✓	AOA	✓			✓	MG Research Lab. in Aalborg Univ., Denmark
[15]	MIP (T-S)	✓		CBC		✓		✓	Grid-connected Campus MG, South Korea

(1) LP/MIP: linear/mixed-integer programming, T-S: two-stage scheduling strategy, (2) OS/CS: open-source/commercial optimization solver, (3) AML/GPL: algebraic modeling/general-purpose programming language, (4) S/I: simulation/implementation, MG: microgrid.

Numerous studies on ED have been conducted from the perspectives of its optimization algorithms [4–10] and actual implementation methods [11–15]. Table 1 summarizes the optimization algorithms, optimal solvers, programming languages, and demonstration methods used in these studies. The optimization algorithms to the ED problem have been well studied. Particularly, in references [4–10], various analytic-based optimization methods, such as linear programming (LP), mixed-integer programming (MIP), and convex optimization, have been investigated for the application to ED. In reference [4], day-ahead ED was formulated as an LP problem to determine the hourly planned operation set points (i.e., charging and discharging power references) of battery-based energy storage systems (ESSs) and electric vehicles (EVs). In reference [5], convex optimization method was used to determine the optimal charging and discharging schedules of an ESS. In reference [6], an online battery power control method based on an MIP formulation was used over a rolling horizon window. The ED strategies proposed in [4–6] planned the operation schedule only once and the schedule is followed without any modifications during operation. However, one-time scheduling strategies are vulnerable to uncertainty in the operational conditions, such as forecasting errors in load demand and renewable generation. To address this problem, a two-stage ED strategy proposed in [7–10] was employed. In the first stage, the optimal operation schedule is developed one day before the operating day (i.e., day-ahead scheduling). In the second stage, the operation schedule

obtained from the first-stage ED is updated repeatedly during actual operation (i.e., real-time scheduling). In references [7,8], the two-stage ED strategy was applied for combined cooling, heating and power (CCHP) MG, and networked MGs. Two-stage ED strategy determines the daily charging and discharging power profiles of batteries in reference [9], and the schedules of ESSs and controllable loads in reference [10].

With regard to research on the actual implementation of ED, in references [11–15], field demonstrations were conducted to validate ED strategies under real MG operation. Specifically, all these studies considered battery-based ESSs as their dispatchable DERs for ED and most of them [11–14] established their ED environment using commercial optimization solvers, as shown in Table 1. In references [11–13], the ED implementation environment was implemented using GAMS and an optimal schedule was determined using a commercial optimization solver. In reference [13], an ESS scheduling model called a power generation-side strategy, which was defined as a MIP problem, was implemented to schedule the 2.2 kVA ESS under forecasting errors in load demand and photovoltaic (PV) generation. In reference [14], based on the power generation-side strategy used in [13], an MIP-based ED strategy was implemented considering Spanish self-consumption regulatory constraints, which were introduced to facilitate the high penetration of renewable energy resources into MGs, via a commercial AML of AIMMS and its add-in optimization solver AIMMS Outer Approximation (AOA). Due to the guaranteed high computational capabilities of commercial optimization solvers, they have been commonly adopted for an optimal scheduling application of power systems. They are particularly effective for the optimization of complex problems involving a substantial number of decision variables. However, such extreme computational capabilities provided by commercial solvers may not be necessary for the application to ED of MGs, which generally involves determining the operation schedules of a small number and simple combination of energy devices. The licensing fees to use the commercial solvers can lead to significant increase in the setup costs for MGs, which can aggravate the economic feasibility of the MG operation [16].

In this context, open-source-based optimization for power systems has recently received much attention [17,18]. For example, in reference [15], the ED was implemented in a completely open-source-based environment established using Python, with the incorporation of a coin-or branch and cut (CBC) open-source optimization solver. In this case, a campus MG scheduling was on time, and the daily operational costs were reduced by 58% using open-source optimization. In reference [19], a completely open-source-based ED environment was established using the open-source programming language Julia and its corresponding open-source optimization solver ECOS; computational capability was verified over a 1-year period for a massive operation dataset measured from a German transmission network. The results demonstrated that the open-source environment is suitable for hourly ED of a large-scale grid.

However, for an open-source-based optimization solver to be further widely adopted for ED of MGs, the suitability of their computational capabilities for ED application should be examined in detail. In particular, to lay a firm basis to use open-source-based solvers, the computation time and optimal results obtained using open-source-based solvers must be compared with those accomplished using the commercial ones. A comparison of the commercial and open-source optimization solvers under various operating conditions will help MG operators select an appropriate optimization solver that satisfies their technical as well as budgetary requirements. The differences in the computational capabilities of open-source-based solvers and commercial ones were examined for several test optimization problems in references [16,20], and the open-source-based solvers showed comparable computational capability as those of the commercial solvers in some cases.

Being motivated by the previous comparative studies on optimization solvers and the recent increase in the needs for open-source-based ED environment, this paper presents a comparative study on various optimization solvers to investigate their suitability for ED in MEMS environments. The main contributions of this manuscript are as follows:

- The computational capabilities of the four widely employed optimization solvers, including CPLEX and Gurobi as commercial solvers and GNU linear programming kit (GLPK) and CBC as

- open-source solvers, are compared in terms of optimal cost, scheduling results, and computational time. The results will be helpful for MG operators, who seek to find a cost-effective optimization solver that best fits their technical requirements and budgetary constraints.
- To examine the applicability to both day-ahead and real-time scheduling, the two-stage scheduling algorithm was adopted as an ED strategy, whereby operation set points of ESSs and the power injected from the main-grid are coordinated to minimize the operating costs of grid-connected MGs.
 - To compare the optimization solvers considering actual operating conditions, a 6-bus campus MG with actual renewable generation and load data [15] was used as a simulation test-bed. For a more comprehensive investigation, additional simulation case studies were conducted on IEEE 33-bus and 123-bus test systems [21], which were modified slightly to emulate mid- and large-scale MGs, respectively.
 - A modularized programming strategy is presented for a fair comparison between the optimization solvers. The modularized programming strategy provides an effective way to avoid redundant usage of functions and variable definitions. It also allows the overall architecture of the code to be developed easily with enhanced readability, which is critical for managing and debugging the programming code.

The rest of the manuscript is organized as follows. Section 2 presents an overview of the ED implementation environment. Section 3 reviews the two-stage ED algorithm and explains its implementation using modular programming architecture. Section 4 discusses simulation case study results, and Section 5 concludes the paper.

2. Overview of the ED Implementation Environment

2.1. Configuration of the ED Implementation Environment

Figure 1 presents an overview of the implementation environment for the ED in the MEMS. The implementation environment can be programmed either in GPLs or in AMLs. Compared with GPLs, AMLs have the advantage of having a syntax similar to that of the mathematical notation of optimization problems, which allows for intuitive writing and understanding of the code. In the GPL-based implementation, this difference can be complemented by adopting a modeling package. The modeling package provides a convenient programming interface similar to that of AMLs, so it is widely employed with optimization solvers when an optimization environment is established using GPLs. However, the overall computational speed slows significantly as additional processes are added due to the modeling package. This will be discussed in detail, along with the simulation results, in Section 4.

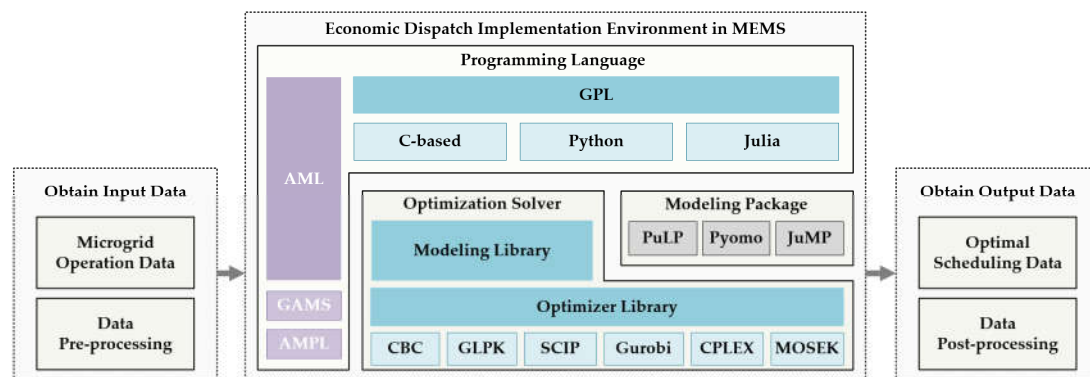


Figure 1. Overview of the economic dispatch (ED) implementation environment.

Optimization solvers consist of functional libraries, a set of functions dedicated to optimization. Functional libraries are often classified into two types according to their tasks: modeling libraries and optimizer libraries. The modeling library formulates the optimization problem for the ED using input

data, such as the predicted load demands and renewable generation, MG parameters and energy resources, and electricity price. The optimizer library is responsible for finding the optimal solution according to a pre-programmed solution algorithm, which is often based on analytic methods involving LP, MIP, and convex optimization.

When an optimization solver is used within the environment established in AMLs or employed with a modeling package, the modeling library is not used. Instead, the optimization problem is first formulated by the AML or modeling package and is then passed on to the optimizer library. An optimization solver is compatible with certain kinds of modeling packages. Table 2 lists the optimization solvers based on their type and compatible modeling packages. Each GPL supports different modeling packages. For example, PuLP and Pyomo are the modeling packages supported in Python environments, and JuMP is the modeling package available with Julia.

Table 2. Summary of optimization solvers and compatible modeling packages.

Solver	Solver Types ⁽¹⁾		Python Modeling Packages		Julia Modeling Packages
	OS	CS	PuLP	Pyomo	JuMP
CBC	✓		✓	✓	✓
GLPK	✓		✓	✓	✓
CPLEX		✓	✓	✓	✓
Gurobi		✓	✓	✓	✓
SCIP		✓		✓	✓
XPRESS		✓	✓	✓	✓
MOSEK		✓		✓	✓

(1) OS/CS: open-source/commercial optimization solver

2.2. Modularized Programming Architecture

For a fair comparison of computational capability, the optimization solvers should be executed within the ED environment programmed with the same architecture. In this study, this was achieved by modularizing the entire code according to the tasks. Figure 2 presents the modularized architecture for implementing the two-stage ED strategy. Module 1 (M1) is responsible for loading the data, e.g., load demands, renewable generation forecasts, parameters of the energy resources, and electricity price, into the ED environment. Such information can be obtained from the forecasting system or the supervisory control and data acquisition (SCADA) system established in the MEMS [22]. Based on the acquired information, M2 and M3 formulate the **A** and **b** matrices, representing the equality and inequality constraints of the ED, respectively. The equality constraints include equality operating conditions, as well as physical models of the MGs and devices (e.g., power balance equations and battery energy conservation). The inequality constraints comprise inequality operating conditions for reliable MG operation, such as the state-of-charge (SOC) limits and charging and discharging power limits of ESSs; Section 3 presents the details of the constraints considered in this investigation. In M4, **A** and **b** matrices are provided as inputs to the modeling package or optimization solver to formulate and solve the optimization problem.

Figure 2 presents the first and second scheduling stages of the two-stage ED strategy, executed using an identical program architecture. Therefore, redundant usage of functions and variable definitions that occur when programming the two scheduling stages separately is effectively avoided. Additionally, the overall structure of the code can be developed easily with enhanced readability to make the writing and debugging of the programming code more efficient.

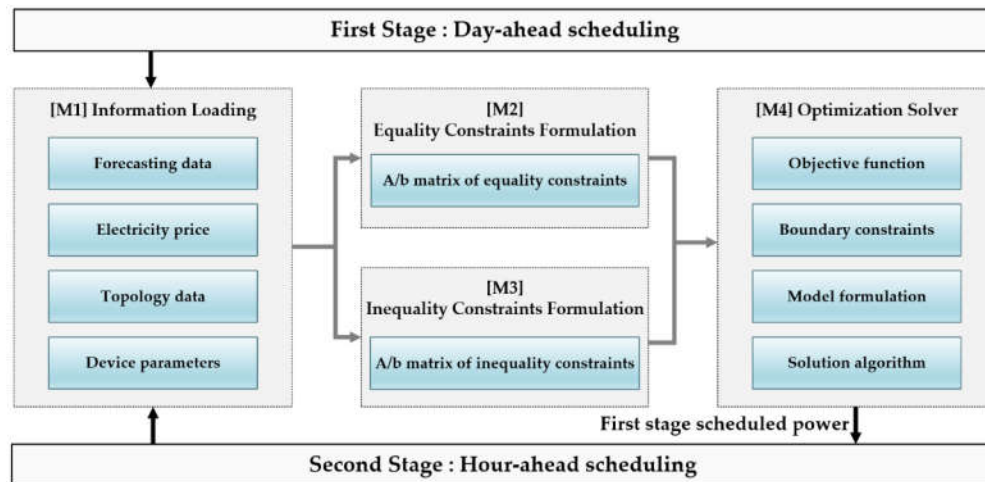


Figure 2. Framework of the two-stage ED modularization architecture.

3. Description of the Two-Stage ED Strategy

3.1. MG Configuration

Figure 3 presents a schematic diagram of a grid-connected MG, where PV generators and Li-ion battery-based ESSs are considered as DERs. Battery-based ESSs have high energy densities and rapid dynamic responses, which make them suitable for high-power and high-frequency cyclic operation [23]. The operation set points (i.e., charging and discharging power references) of the ESSs were dispatched from the MEMS via communication links. The operating cost of an MG is given by the total daily electricity fee charged to the imported power from the main grid (i.e., utility power). To minimize operating costs, the MEMS implements the two-stage ED strategy, while considering the forecasted load demand and renewable generation. Note that the performance of the optimization module in the MEMS depends on the accuracy of the forecasting module. Even though the forecasting error is inevitable, it is assumed that the forecasting module provides accurate forecasts of load demand and renewable generation output power. This is not a radical assumption, and already there have been many studies that show high accuracy with many forecasting algorithms such as linear regression, clustering, and support vector machine (SVM) [24,25]. When the amounts of accumulated data in the database increase over time, the accuracy of forecasts will improve. In another approach to deal with the variation and uncertainties, robust ED strategies based on stochastic optimization were proposed in [26,27]. This approach can provide robust and efficient solutions, but modeling with scenarios can increase the computational difficulty of the optimization problem. To address this problem, a problem modeling approach proposed in [28] and a scenario selection algorithm in [29] were proposed. However, we did not consider the representative scenarios since we applied not the stochastic optimization method but the deterministic optimization approach to the proposed ED strategy. For brevity, this stochastic optimization approach is not further discussed in this paper, where instead we focus on the deterministic one. Furthermore, it is possible to get more accurate forecasts as real-time operation becomes more imminent [30]. Thus, we proposed the two-stage scheduling to mitigate forecast errors and improve the accuracy of scheduling.

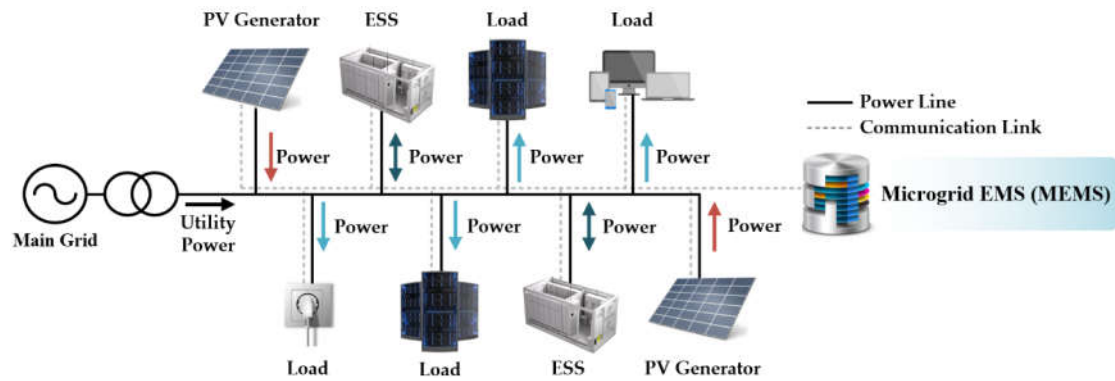


Figure 3. Configuration of a grid-connected microgrid (MG).

3.2. Two-Stage ED Strategy

In the two-stage ED strategy, the operation schedule of the first stage is such that it minimizes the total daily operating cost of the MG. First-stage scheduling is conducted only once, one day prior to the actual day of operation (i.e., day-ahead scheduling). The single scheduling time interval of the first stage is 1 h. The scheduled profiles do not change during an hour. The second stage is implemented to mitigate the difference between the scheduled utility power and the second-stage scheduled utility power, which arises due to variabilities in load demand and renewable generation. Figure 4 presents the timeframe of the ED strategy. In second-stage scheduling, the operating set points of ESSs and the utility power scheduled in the first stage are updated every 5 min during actual operation, with a unit scheduling time step of 5 min. The second stage is carried out in an hour-ahead scheduling mode, with a unit scheduling time step of 5 min. The time horizon of the second stage diminishes by 5 min at each execution, as shown in Figure 4, and recovers over the subsequent hour. For the first 5-min interval of every hour, the operating set points of the ESSs are given as scheduled in the first stage. After the first 5-min interval, the operation schedule is updated every 5 min by the second-stage scheduling during the remaining 55 min, with a diminishing time horizon. Thus, the initial scheduling of every hour has more time steps to schedule (i.e., 11 time steps) compared with subsequent scheduling within the hour. In this way, second-stage scheduling effectively compensates for the uncertainties inherent in day-ahead scheduling.

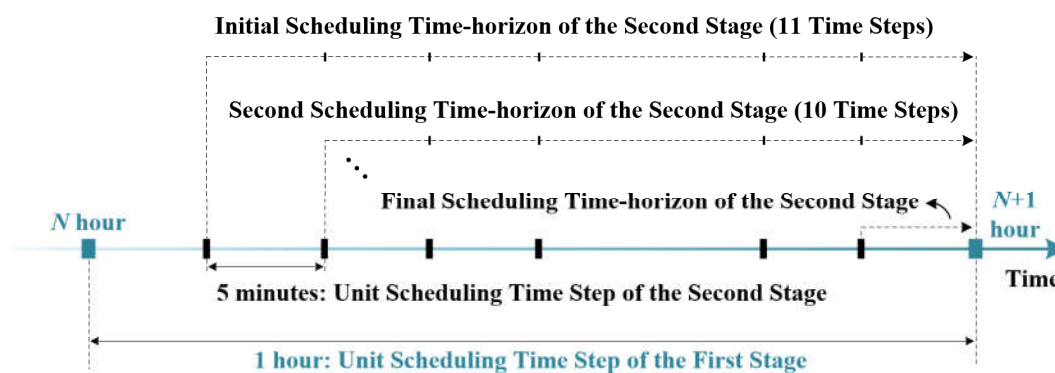


Figure 4. Timeframe of the two-stage ED strategy.

The objective, time window, and time interval of each stage are specified as follow:

1. First-stage scheduling: In this first-stage (i.e., day-ahead scheduling), hourly schedules for a day are performed one day in advance. The objective function is to minimize the operating cost. The total time window is 24 h, and the single scheduling time interval is 1 h. The scheduled profiles do not change during an hour.

2. Second-stage scheduling: In the second stage (i.e., hour-ahead scheduling), the optimal schedule obtained from the first stage is updated every 5 min during the day of operation to compensate for the uncertainties in load demands and PV output power. It should be noted that hour-ahead scheduling is not performed at the beginning of every 5-min interval, but rather follows the day-ahead scheduled profile. In other words, the second stage runs 11 times per an hour. Only the first interval of each run set serves as a final decision and the rest of the intervals are for reference only.

The following subsections present the detailed objective functions, constraints, and model for implementing the two-stage ED strategy; most of these have been employed in the author's previous work [15].

3.2.1. Objective Function

The objective function of the first stage is given as

$$\min \left\{ \sum_t (c_t P_{u,t}^d) + \alpha_1 U_{1,t} \right\} \quad \forall t \in T_d, \quad (1)$$

$$U_{1,t} = \max(0, P_{u,t}^d - P_{cont}) \quad \forall t \in T_d, \quad (2)$$

where c_t is the hourly electricity price of the main grid and $P_{u,t}^d$ is the imported power from the utility at time t ; α_1 is the weighting coefficient for the penalty function and $U_{1,t}$ is the continuous decision variable determining the penalty function at time t ; P_{cont} is the contracted power and T_d is set of hourly periods. Equation (1) represents the objective to minimize daily operation costs, which consist of the charged electricity fee on the imported power from the utility side and the penalty function to maintain the utility power within the contracted power. For the hourly varying electricity fee, time-of-use (TOU) rates were considered in this study. Equation (2) describes the penalty term.

The objective of second-stage scheduling is to mitigate the difference between scheduled utility power and second-stage scheduled power due to uncertainties in load demands and renewable generation. The objective function of the second stage is given as

$$\min \{ (|P_{u,t}^d - P_{u,t}^h|) - \alpha_2 U_{3,t} + \alpha_3 U_{4,t} \} \quad \forall t \in T_h, \quad (3)$$

$$U_{3,t} = \min(0, SOC_{n,t}^h - SOC_n^{min}) \quad \forall n \in N, \quad \forall t \in T_h, \quad (4)$$

$$U_{4,t} = \max(0, SOC_{n,t}^h - SOC_n^{max}) \quad \forall n \in N, \quad \forall t \in T_h, \quad (5)$$

where $P_{u,t}^h$ is the hour-ahead scheduled utility power at time t ; α_2 and α_3 are the weighting coefficients for penalty functions; $U_{3,t}$ and $U_{4,t}$ are the continuous decision variables determining the penalty function at time t , which consider the exceeded SOC value; $SOC_{n,t}^h$ is the state-of-charge of n th ESS in the second stage at time t ; SOC_n^{min} and SOC_n^{max} are the minimum and maximum SOC limit of the n th ESS; T_h is the set of second stage scheduling time periods. In Equation (3), the second and third terms are penalty functions to prevent the overcharge and discharge of ESSs due to parameter uncertainties and nonlinear operation characteristics, such as natural battery discharge.

3.2.2. Equality Constraints

The equality constraints consist of the physical model of the MG and ESSs, as well as equality operating conditions to assure the uniform and continuous participation of ESSs in the ED. The physical model of MG and ESSs are represented by the power balance equation and energy convergence in the battery, respectively. The power balance between net power production and consumption in an MG is represented by the linearized DistFlow formulation [31]. The power balance equation used in day-ahead and hour-ahead scheduling is given as

$$P_{u,t} + \sum_{j: (i,j) \in L} P_{j,i,t} = \sum_{k: (k,i) \in L} P_{i,k,t} + (P_{ch,n,t} - P_{dch,n,t}) + (P_{L_i,t}^f - P_{PV,t}^f) \quad (6)$$

$$\forall i, j, k \in B, \quad \forall t \in \{T_d, T_h\}, \quad \forall n \in N,$$

where $P_{j,i,t}$ and $P_{i,k,t}$ are the power flow from bus j to bus i and from bus i to bus k at time t , respectively; $P_{L_i,t}^f$ and $P_{PV,t}^f$ are the forecasted load demand and PV generation at bus i at time t , respectively; $P_{ch,n,t}$ and $P_{dch,n,t}$ are the scheduled charging and discharging power of n th ESS at time t , respectively; B is the set of buses; and L is the set of power lines in an MG.

The battery-based ESSs are modeled by the energy convergence stored in a battery, which is represented by the variation in SOC level with respect to the input power, given by

$$SOC_{n,t} = SOC_{n,t-1} + ((\eta_{ch,n} P_{ch,n,t} - P_{dch,n,t} / \eta_{dch,n}) \Delta t) / CAP_n^{max} \quad \forall n \in N, \quad \forall t \in \{T_d, T_h\}, \quad (7)$$

where $SOC_{n,t}$ is the SOC level of the n th ESS at time t ; $\eta_{ch,n}$ and $\eta_{dch,n}$ are the charging and discharging efficiency of the n th ESS; CAP_n^{max} is the maximum capacity of the n th ESS.

Generally, charging and discharging of batteries is strictly determined by cost-effective operation constraints. However, without using constraint (8), the final SOC level might be extremely low, especially when the PV output power is deeply low and load demand is high. To ensure the continuous participation of ESSs with uniform performance in next-day ED, the SOC levels at the initial and final time steps of first-stage scheduling should be equal [32]. The energy stored at the last time of scheduling should be set to its initial value, as

$$SOC_{n,final} = SOC_{n,init} \quad \forall n \in N, \quad (8)$$

where $SOC_{n,init}$ and $SOC_{n,final}$ are the SOC levels at the initial and final time steps of a day, respectively. The equality constraint (8) may be relaxed to inequality constraints that represent the acceptable variation in the SOC with respect to the initial SOC level, to allow for more flexible operation of ESSs [33].

Meanwhile, the objective functions given in Equations (1) and (3) should be linearized to be applied to the two-stage MIP strategy. Here, the piecewise linearization method is employed. The linearization introduces additional equality constraints and variables, as

$$P_{u,t}^d = U_{1,t} + U_{2,t} + r_0 \quad \forall t \in T_d, \quad (9)$$

$$SOC_{n,t}^h = U_{3,t} + U_{4,t} + U_{5,t} + r_3 \quad \forall t \in T_h, \quad (10)$$

where r_0 and r_3 are the parameters for objective function linearization and $U_{2,t}$ and $U_{5,t}$ are the continuous variables for objective function linearization at time t .

3.2.3. Inequality Constraints

Inequality constraints include charging/discharging power capacity and SOC limits of ESSs, as well as contracted power penalty functions. The SOC limit constraint is given by

$$SOC_n^{min} \leq SOC_{n,t} \leq SOC_n^{max} \quad \forall n \in N, \quad \forall t \in \{T_d, T_h\}, \quad (11)$$

Charging/discharging power limits of an ESS is given as

$$0 \leq P_{ch,n,t} \leq u_{n,t} P_{ch,n}^{max} \quad \forall t \in \{T_d, T_h\}, \quad (12)$$

$$0 \leq P_{dch,n,t} \leq (1 - u_{n,t}) P_{dch,n}^{max} \quad \forall t \in \{T_d, T_h\}, \quad (13)$$

where $u_{n,t}$ is a binary decision variable to determine the operation status of the n th ESS at time t (i.e., $u_{n,t} = 1$: charging and $u_{n,t} = 0$: discharging).

To prevent the utility power from exceeding the contracted power in real-time scheduling, a surcharging system is considered in the second scheduling stage. The electric meter installed by Korea Electric Power Corporation (KEPCO), South Korea, estimates the peak power over a 15-min period [34]. When the peak power that updated from the electric meter exceeds the contracted power,

KEPCO imposes extra charges for overuse. The surcharging system requires that the utility power averaged over a 15-min period be maintained within the contracted power to stabilize the supply. This requirement is imposed by

$$(P_{u,t-1} + P_{u,t} + P_{u,t+1}) / 3 \leq P_{cont} \quad \forall t \in T_h, \quad (14)$$

Additionally, linearizing the first-stage objective function (1) using the piecewise linearization method yields the inequality constraints

$$(r_1 - r_0)S_{1,t} \leq U_{2,t} \leq (r_1 - r_0) \quad \forall t \in T_d, \quad (15)$$

$$0 \leq U_{1,t} \leq (r_2 - r_1)S_{1,t} \quad \forall t \in T_d, \quad (16)$$

where r_1, r_2 are parameters for objective function linearization and $S_{1,t}$ is a binary variable for linearization of objective function at time t .

Similarly, linearizing the second-stage objective function (9) gives additional inequality constraints expressed by

$$(r_4 - r_3)S_{2,t} \leq U_{3,t} \leq (r_4 - r_3) \quad \forall t \in T_h, \quad (17)$$

$$(r_4 - r_3)S_{3,t} \leq U_{5,t} \leq (r_5 - r_4)S_{2,t} \quad \forall t \in T_h, \quad (18)$$

$$0 \leq U_{4,t} \leq (r_6 - r_5)S_{3,t} \quad \forall t \in T_h, \quad (19)$$

where r_4 and r_5 are parameters for objective function linearization and $S_{2,t}$ and $S_{3,t}$ are binary variables for objective function linearization at time t . Finally, nonlinear objective functions (1) and (3) are linearized to a set of equality constraints (9), (10), and inequality constraints (15)–(19), which are in the form of an MIP problem tractable for analytic method-based optimization solvers.

3.3. Implementation of the Two-Stage Strategy with the Modular Architecture

Figure 5 presents the coupling of the two-stage ED strategy using a modular program architecture, as discussed in Section 2. Four modules were deployed in both the first and second scheduling stages. The detailed tasks of each module are described below.

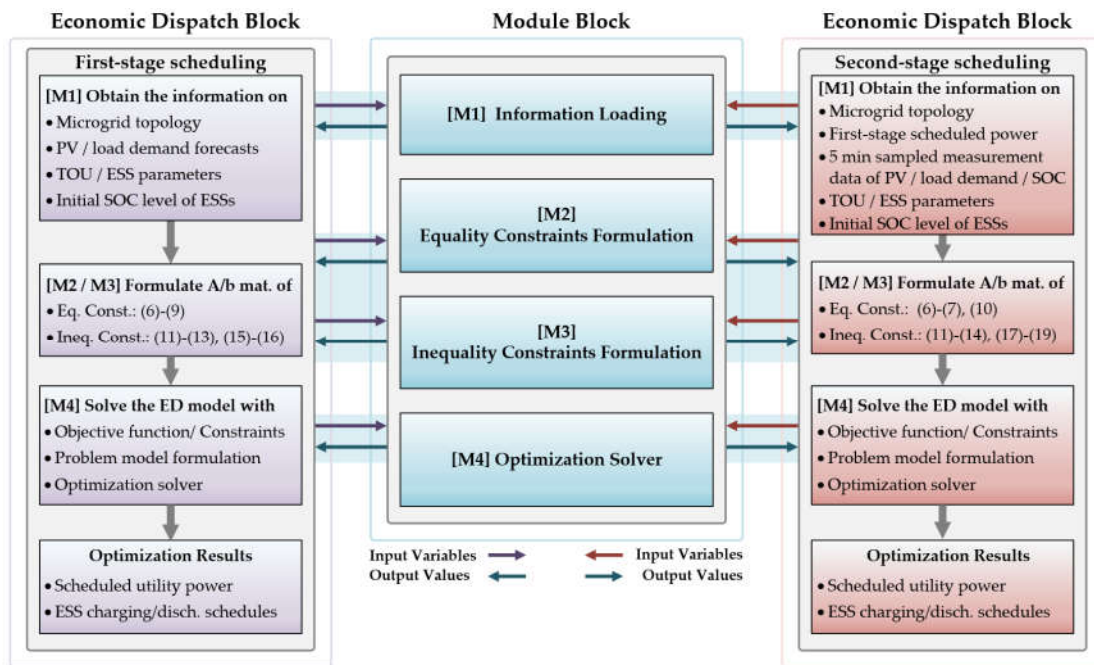


Figure 5. Coupling of the two-stage ED problem block and the module block.

Module (1) Loading the input data for each scheduling stage as follows:

- First-stage scheduling: MG topology; PV generation and load demand forecasts; electricity price; initial SOC level of ESSs; ESS parameters.
- Second-stage scheduling: First-stage scheduled utility power; MG topology; 5-min sampled measurements of PV generation, load demand, SOC levels, and utility power; electricity price; initial SOC level of ESSs; ESS parameters.

Module (2) Creating the **A** and **b** matrices that represent the equality constraints of an optimization problem using the following Equations:

- First-stage scheduling: Equations (6)–(9).
- Second-stage scheduling: Equations (6), (7), and (10).

Module (3) Creating the **A** and **b** matrices that represent the inequality constraints of an optimization problem using the following Equations:

- First-stage scheduling: Equations (11)–(13), (15), and (16).
- Second-stage scheduling: Equations (11)–(14), (17), and (19).

Module (4) Formulating the optimization problem from the **A** and **b** matrices obtained from M2 and M3 using the modeling package or modeling library of the optimization solver in a GPL-based platform or using the inherent interface provided in the AML-based environment; solving the optimization problem by executing the optimizer library of the optimization solver.

4. Simulation Case Studies for Comparison

4.1. Simulation Conditions

Four widely adopted optimization solvers, CPLEX, Gurobi, GLPK, and CBC, were applied for the two-stage ED strategy in simulation case studies, to compare computational capabilities. Figure 6 presents three grid-connected MGs considered for the test-beds containing loads, PV generators, and ESSs. The 6-bus MG shown in Figure 6a represents a small-scale campus MG in operation [15]. The IEEE 33-bus and 123-bus radial test systems [21] shown in Figure 6b and Figure 6c, respectively, were also examined, with slight modifications to emulate mid- and large-scale MG operations.

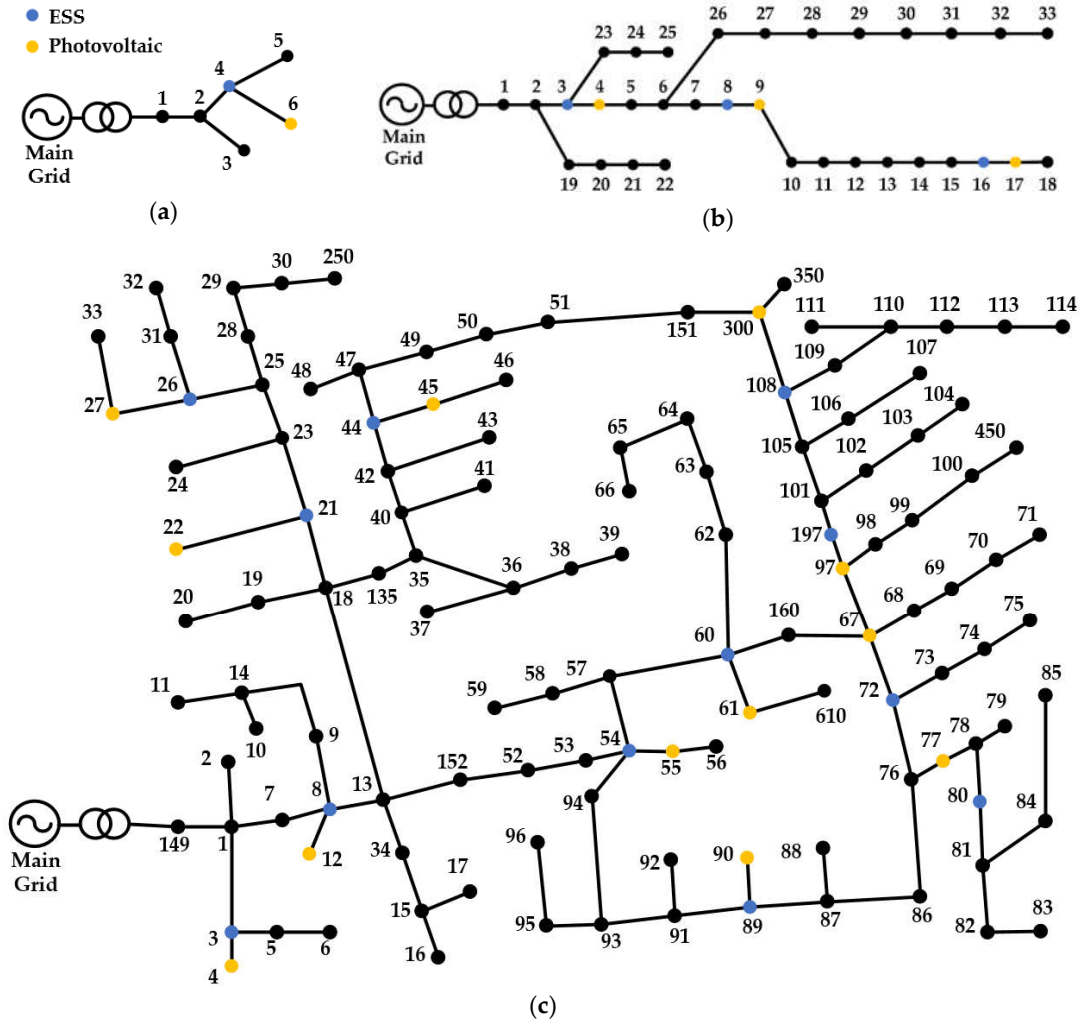


Figure 6. Grid-connected MGs for simulation case study. (a) 6-bus campus MG; (b) IEEE 33-bus radial test system; (c) IEEE 123-bus radial test system.

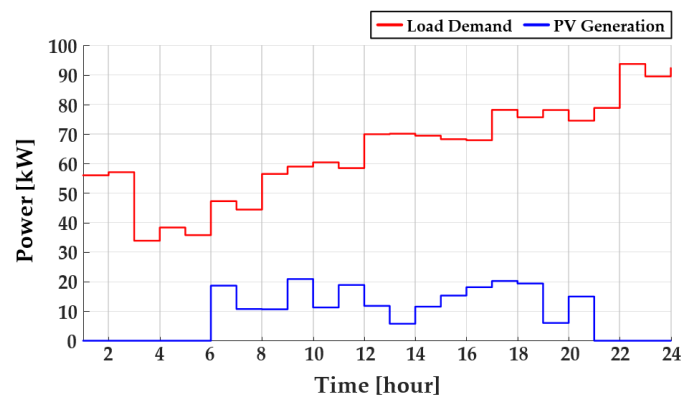
All ESSs in the test-bed MGs were assumed to have the same parameters listed in Table 3, which is an actual parameter set provided by an ESS manufacturer. Table 4 lists the three-level (i.e., off-peak, mid-, and peak-loads) TOU rates used in the simulation case studies; these are the rates used by KEPCO, South Korea [35]. Figure 7 presents the hourly sampled actual load demand and PV generation forecasting data provided in [36]. For simplicity, the PV generation and load demand information of each bus are assumed to have the same forecasting profiles. The real-time variations in load demands and PV generations were emulated by applying the 5-min sampled stochastic variations to the forecasting data shown in Figure 7, which are uniformly distributed within the range of $\pm 5\%$ of them. The contracted utility power was set to 2 MW. Table 5 lists the specifications of the implementation environment. The compared optimization solvers were applied with the open-source modeling package PuLP in Python, using an Intel Xenon Silver 4114 CPU with 192.0 GB RAM. The computational capability of the optimization solvers was evaluated by investigating the optimal costs, scheduling results, and computation time. Simulations were performed 100 times for each optimization solver, and the average computation times were compared.

Table 3. Actual energy storage system (ESS) parameters.

Parameters	Values	Units
P_{ch}^{max}	50	[kW]
P_{dch}^{max}	50	[kW]
EC^{max}	250	[kWh]
SOC^{max}	90	[%]
SOC^{min}	20	[%]
η_{ch}	80	[%]
η_{dch}	80	[%]

Table 4. Three-level time-of-use (TOU) rates on utility power.

Classifications	Time Periods	Prices (\$/kWh)
Off-peak load	23:00–09:00	0.038
	09:00–10:00	
Mid-load	12:00–13:00	0.076
	17:00–23:00	
Peak-load	10:00–12:00	0.130
	13:00–17:00	

**Figure 7.** Forecasting load demand and photovoltaic (PV) generation power profile.**Table 5.** Detailed specifications of the implementation environment.

Functionality	Software	Types	Versions
GPL	Python	Open-source	3.6.9 [37]
Modeling Package	PuLP	Open-source	1.6.8 [38]
Optimization Solver	CBC	Open-source	2.10 [39]
	GLPK		4.65 [40]
	Gurobi	Commercial	8.1.1 [41]
	CPLEX		12.9 [42]

4.2. Comparison of Optimal Costs and Scheduled Profiles

All the compared optimization solvers identically determined the minimal operating costs of the three MGs as USD 86.20, USD 1698, and USD 8610; thus, the same optimal results were achieved. Figure 8 presents the scheduled power profiles of the ESS and utility power for the three MGs. The ESS power profiles shown in Figure 8b and Figure 8c are those of arbitrarily selected ESSs in the 33-bus and 123-bus systems (Figure 6b and Figure 6c, respectively). Note that positive power represents discharging of the ESS. In contrast, the ESS power profiles differed from each other, as shown in Figure 8a–c. However, they have a common tendency; the ESSs are charged when the TOU is low

(e.g., from 18:00 to 24:00) and discharged when the TOU is high (e.g., from 10:00 to 18:00). Note that all optimization solvers scheduled the ESS power to be maintained well within its limit (i.e., 50 kW).

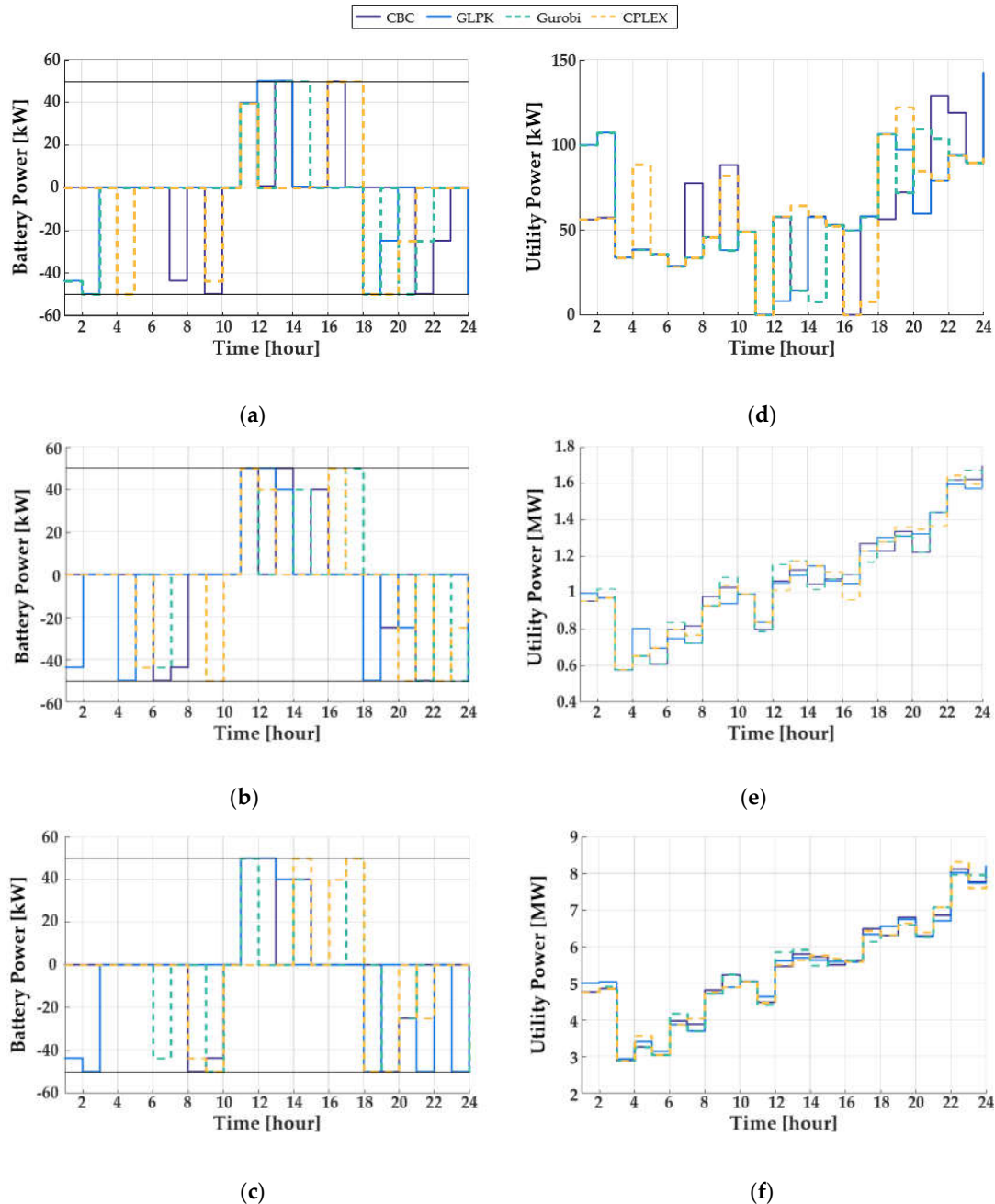


Figure 8. Energy storage system (ESS) and utility power profiles scheduled by the optimization solvers. (a,d) case 1: 6-bus system; (b,e) case 2: 33-bus system; (c,f) case 3: 123-bus system.

Similarly, the scheduled utility power profiles shown in Figure 8d–f have a common tendency to import the high power from the main grid when the TOU is inexpensive (e.g., from 18:00 to 24:00) to enhance MG operation efficiency by lowering operating costs. This tendency was due to not only the low TOU price but also the increased load demand at that time, as shown in Figure 7.

Figure 9 presents the variations in the scheduled SOC levels corresponding to the ESS power profiles of Figure 8a–c. All of the scheduled SOC levels were well maintained within the limits for secure operation (i.e., 20% to 90%). Additionally, all of the compared optimization solvers succeeded

in making the SOC levels at the final time step equal to the initial SOC levels, assuring continuous participation in the next-day ED with uniform contribution.

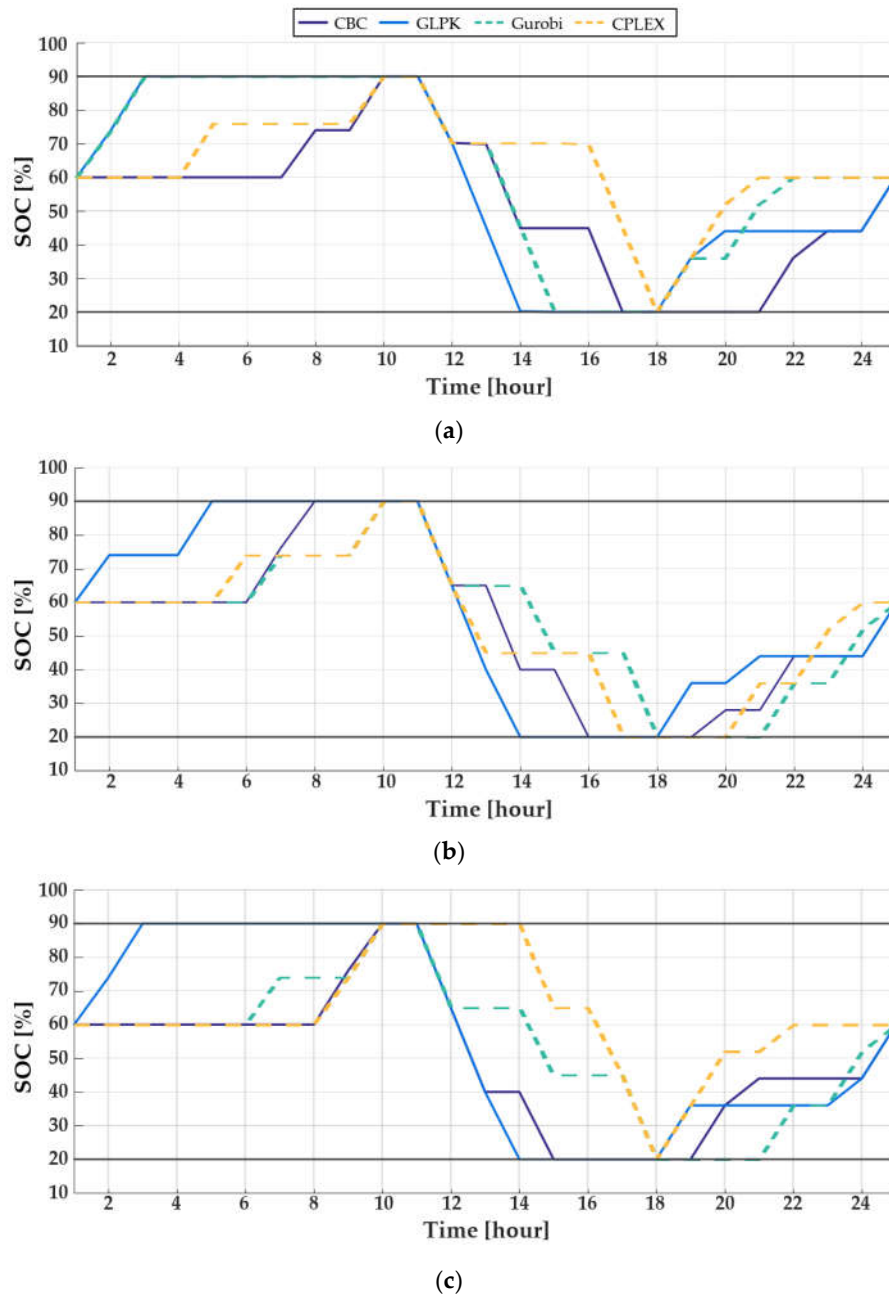


Figure 9. Variation of the the state-of-charge (SOC) levels corresponding to the scheduled ESS power. (a) case 1:6 bus system; (b) case 2:33 bus system; (c) case 3:123 bus system.

4.3. Comparison of Computation Time

Table 6 lists the computation times of the optimization solvers for the three simulation cases. To investigate the computation time without a modeling package, additional tests were conducted with CPLEX and Gurobi, which provide application programming interfaces (APIs) for the Python environment. Using APIs, the optimization solver can be interfaced directly with the ED environment without a modeling package. In first-stage scheduling, all of the compared optimization solvers yielded a suitable computation time for the day-ahead scheduling application in all cases.

Specifically, for all three cases, the open-source optimization solver CBC yielded the maximum computation time (i.e., 1.32, 12.42, and 113.64 s, respectively).

Table 6. Computation times of the optimization solvers.

Cases	Modeling Packages	Solvers ⁽¹⁾	First Stage (s)	Second Stage (s)	
				Avg. Initial Scheduling	Max. Initial Scheduling
Case 1 6 Bus system	PuLP	CBC	1.32	0.44	0.47
		GLPK	1.25	0.42	0.47
		Gurobi	1.02	0.38	0.43
		CPLEX	1.03	0.41	0.49
	Gurobi (API)		0.26	0.03	0.03
	CPLEX (API)		0.13	0.02	0.02
Case 2 33 Bus system	PuLP	CBC	12.42	3.80	3.93
		GLPK	11.55	3.81	3.96
		Gurobi	10.94	3.59	3.68
		CPELX	10.98	3.76	3.98
	Gurobi (API)		2.67	0.25	0.28
	CPLEX (API)		0.61	0.07	0.08
Case 3 123 Bus system	PuLP	CBC	113.64	37.61	42.42
		GLPK	110.87	38.74	42.70
		Gurobi	105.13	36.52	39.11
		CPELX	105.36	35.57	39.10
	Gurobi (API)		27.82	2.42	2.52
	CPLEX (API)		3.55	0.76	0.80

(1) API: application programming interface

For the second-stage scheduling, the maximum and average computation times of the initial scheduling of each hour were investigated. This is because the initial scheduling has the most time steps to schedule (i.e., 11 time steps) as shown in Figure 4. Thus, the longest computation time will occur during the initial scheduling, and subsequent scheduling will have shorter computation times. For all optimization solvers, the computation time for the second-stage scheduling also yielded sound results for the hour-ahead or real-time scheduling application, with the maximum and the average computation time for initial scheduling of all the optimization solvers completed within 300 s (i.e., 5 min).

Figure 10 presents the computation times obtained with the modeling package. In both first-stage and second-stage scheduling, the open-source optimization solvers (i.e., CBC and GLPK) and the commercial optimization solvers (i.e., Gurobi and CPLEX) yielded similar computation times. The main reason for the similar computation times from the different optimization solvers is due to the additional processes by the modeling package: converting the input data syntax given in high-level mathematical notation into low-level syntax that can be processed by optimization solvers took up most of the total computation time. Therefore, when the optimization solvers were executed using their APIs (i.e., Gurobi (API) and CPLEX(API)), they yield much shorter computation times; however, as noted above, the prolonged computation time due to the modeling package is a trade-off to convenient interfaces. Another factor that determines the computation times is a programming language, and the study that compared programming languages with respect to the computational time, readability, accessibility, and strengths/weaknesses was examined for general-purpose computational problems in reference [43]. However, this is not further discussed in this paper, where we instead focus on the optimization solvers and modeling package.

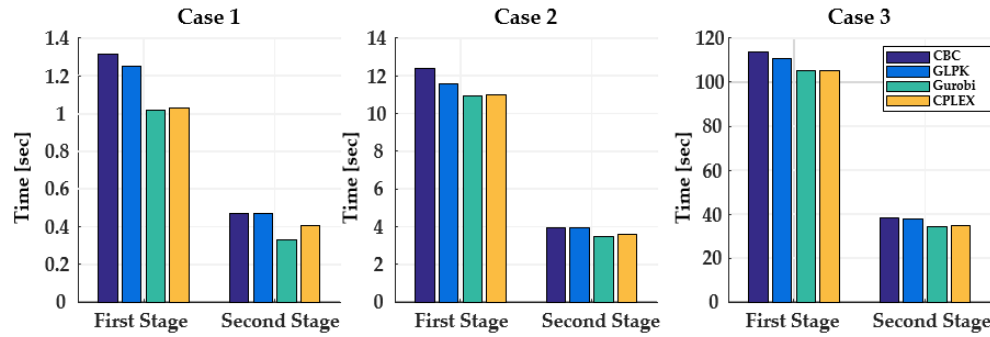


Figure 10. Comparison of the total computation times with modeling package PuLP.

Table 7 lists the detailed computation times spent by the modeling package with respect to its performing of tasks (i.e., formulating equality and inequality constraints). The time ratio represents the portion they occupy in the total computation times listed in Table 6. In all cases, the modeling package accounts for more than half of the total computation time. For further investigation, the computation time data are visualized in Figure 11. In both scheduling stages, the computation time for formulating constraints becomes more dominant than the other processes for optimization in the total computation time as the scale of an MG increases. This result means that the computation speed of an ED environment is largely affected by a modeling package as well as an optimization solver. Furthermore, adopting an efficient modeling package becomes the more important issue, as opposed to which type of optimization solver to use, when the MG involves a large number of buses and DERs.

Table 7. Detailed computation time for the modeling package.

Case	MIP Solver	First Stage			Second Stage		
		Inequality Const. (s)	Equality Const. (s)	Time Ratio (%)	Inequality Const. (s)	Equality Const. (s)	Time Ratio (%)
Case 1 6 Bus system	CBC			67.42			61.93
	GLPK	0.39	0.50	71.20	0.14	0.13	64.88
	Gurobi			87.25			71.71
	CPLEX			86.40			66.46
Case 2 33 Bus system	CBC			82.77			74.90
	GLPK	2.61	7.67	89.00	1.06	1.78	74.71
	Gurobi			93.96			79.29
	CPLEX			93.62			75.70
Case 3 123 Bus system	CBC			91.18			78.97
	GLPK	17.19	86.43	93.46	8.87	20.82	76.66
	Gurobi			98.56			81.33
	CPLEX			98.35			83.50

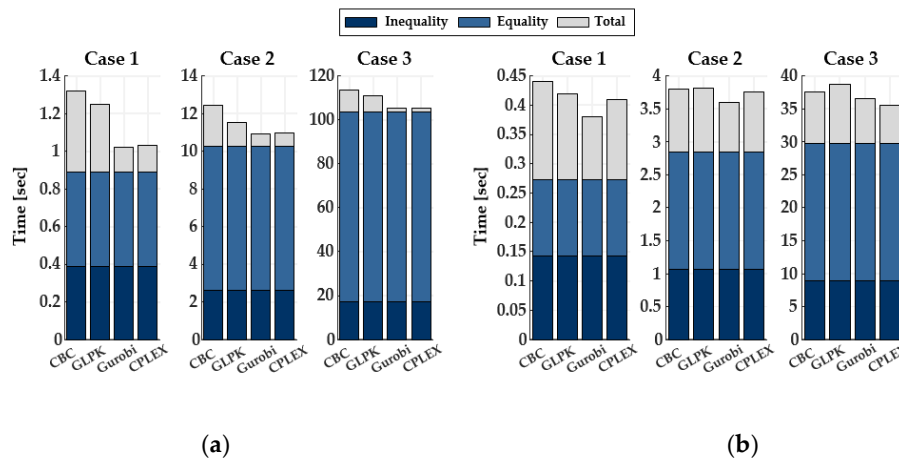


Figure 11. Total computation times. (a) first scheduling stage; (b) second scheduling stage.

5. Conclusions

This paper presents a comparative study on optimization solvers to investigate their suitability for ED applications. Four widely employed optimization solvers, both commercial and open-source-based ones, were compared in terms of optimal cost, scheduling results, and computation time. A two-stage scheduling strategy was applied for the ED algorithm. Equations representing the operating conditions and models of MGs and DERs were formulated and combined into a set of constraints to constitute the MIP problem, which is tractable for off-the-shelf optimization solvers adopting analytic solution algorithms. A modularized programming strategy was also introduced to allow a fair comparison between the optimization solvers and the efficient establishment of the two-stage ED environment. Simulation case studies were conducted on three MGs of varying size. The simulation results revealed that all of the compared optimization solvers were achieved the same optimal costs and operated within the time required for each step of the two-stage scheduling strategy. With regard to computational capability, a much shorter computation time was achieved when the optimization solvers were used with their APIs compared to the cases where the modeling package was employed together with the optimization solvers. However, the delay in computation time due to the modeling package can be compensated for by the convenient interfaces that the modeling package provides. Additionally, for large-scale MG applications, the modeling package is a critical factor that determines the computation speed of the ED environment.

Further work is required to investigate how the computational efficiency of an ED environment varies under different combinations of modeling packages, optimization solvers, and programming languages. Specifically, PuLP was adopted as a modeling package in this study, though other modeling packages, such as Pyomo or JuMP, need to be employed and analyzed in subsequent studies. The relative computational efficiency of AML-based ED environments (e.g., GAMS and AMPL) compared with those that are GPL-based needs to be examined, because the simulation results obtained in this study revealed that formulating the constraints accounts for a large portion of the computation time.

Author Contributions: conceptualization, G.-H.L., J.-Y.P., and Y.-J.K.; methodology, G.-H.L., and S.-J.H.; software, G.-H.L., and S.-J.H.; validation, G.-H.L., J.-Y.P., and Y.-J.K.; formal analysis, G.-H.L., and J.-Y.P.; investigation, G.-H.L., and J.-Y.P.; resources, G.-H.L.; data curation, G.-H.L.; writing—original draft preparation, G.-H.L.; writing—review and editing, J.-Y.P., G.-H.L., and Y.-J.K.; visualization, G.-H.L., and J.-Y.P.; supervision, Y.-J.K.

Funding: This research was funded by Korea Electric Power Corporation, grant number R18XA01.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Acronyms
AML: Algebraic modeling language
AOA: AIMMS outer approximation algorithm
API: Application programming interface
CBC: Coin-or branch and cut
CCHP: Combined cooling, heating and power
DER: Distributed energy resource
ED: Economic dispatch
ESS: Energy storage system
EV: Electric vehicle
GAMS: General algebraic modeling system
GLPK: GNU linear programming kit
GPL: General-purpose programming language
KEPCO: Korea electric power corporation
LP: Linear programming
MEMS: Microgrid energy management system
MG: Microgrid
MIP: Mixed-integer programming
OPF: Optimal power flow
PV: Photovoltaic
SCADA: Supervisory control and data acquisition
SOC: State-of-charge
SVM: Support vector machine
TOU: Time-of-use
Sets and Indices
d : Day-ahead stage values
h : Hour-ahead stage values
T_d : Set of hourly periods in the first stage (i.e., $t \in T_d$ and $T_d = [0, 1, \dots, 23]$)
T_h : Set of 5 minute periods in the second stage (i.e., $T_h = [5, 10, \dots, 55]$)
t : Index for a scheduling time step (i.e., $t \in T_d$ and $t \in T_h$)
i, j, k : Indices for buses in an MG
L : Set of lines of the topology in the MG (i.e., $(i, j), (k, i) \in L$)
B : Set of buses in the MG (i.e., $i, j, k \in B$)
n, N : Index and total number of ESSs, (i.e., $n \in N$)
f : Forecasted values
max, min : Maximum/minimum values
Parameters and Constants
P_{cont} : Contracted power (kW)
α : Weighting coefficients for penalty functions
r : Parameter for objective function linearization
SOC_n^{min}, SOC_n^{max} : Minimum/maximum SOC limit of the n th ESS (%)

$\eta_{ch,n}, \eta_{dch,n}$: Charging/discharging efficiency of the n th ESS (%)
CAP_n^{max} : Maximum capacity of the n th ESS (kWh)
$P_{ch,n}^{max}, P_{dch,n}^{max}$: Maximum charging/discharging power of n th ESS (kW)
$SOC_{n,init}, SOC_{n,final}$: SOC levels of n th battery at the initial/final times steps of a day (%)
Variables
c_t : Hourly electricity price at time t (\$/kWh)
$P_{u,t}$: Imported power from the utility at time t (kW)
$P_{u,t}^h$: Hour-ahead scheduled utility power at time t (kW)
U_t : Continuous decision variable determining the penalty function at time t
S_t : Binary variable for objective function linearization at time t
$P_{j,i,t}, P_{i,k,t}$: Power flow from bus j to bus i and from bus i to bus k at time t
$P_{Li,t}^f, P_{PV,t}^f$: Forecasted load demand/PV generation at bus i at time t (kW)
$P_{ch,n,t}, P_{dch,n,t}$: Scheduled charging/discharging power of n th ESS at time t (kW)
$SOC_{n,t}$: SOC level of the n th ESS at time t (%)
$u_{n,t}$: Binary decision variable to determine the operation status of the n th ESS at time t

References

- Valencia, F.; Collado, J.; Sáez, D.; Marín, L.G. Robust Energy Management System for a Microgrid Based on a Fuzzy Prediction Interval Model. *IEEE Transactions on Smart Grid* **2016**, *7*, 1486–1494.
- Kim, Y.; Wang, J.; Lu, X. A Framework for Load Service Restoration Using Dynamic Change in Boundaries of Advanced Microgrids With Synchronous-Machine DGs. *IEEE Transactions on Smart Grid* **2018**, *9*, 3676–3690.
- Olivares, D.E.; Mehrizi-Sani, A.; Etemadi, A.H.; Cañizares, C.A.; Iravani, R.; Kazerani, M.; Hajimiragha, A.H.; Gomis-Bellmunt, O.; Saeedifard, M.; Palma-Behnke, R., et al. Trends in Microgrid Control. *IEEE Transactions on Smart Grid* **2014**, *5*, 1905–1919.
- Fanti, M.P.; Mangini, A.M.; Roccotelli, M.; Ukovich, W. District Microgrid Management Integrated with Renewable Energy Sources, Energy Storage Systems and Electric Vehicles. *IFAC-PapersOnLine* **2017**, *50*, 10015–10020.
- Zhang, Y.; Gatsis, N.; Giannakis, G.B. Robust Energy Management for Microgrids With High-Penetration Renewables. *IEEE Transactions on Sustainable Energy* **2013**, *4*, 944–953.
- Malysz, P.; Sirouspour, S.; Emadi, A. An Optimal Energy Storage Control Strategy for Grid-connected Microgrids. *IEEE Transactions on Smart Grid* **2014**, *5*, 1785–1796.
- Luo, Z.; Wu, Z.; Li, Z.; Cai, H.; Li, B.; Gu, W. A two-stage optimization and control for CCHP microgrid energy management. *Applied Thermal Engineering* **2017**, *125*, 513–522.
- Wang, D.; Qiu, J.; Reedman, L.; Meng, K.; Lai, L.L. Two-stage energy management for networked microgrids with high renewable penetration. *Applied Energy* **2018**, *226*, 39–48.
- Tuffaha, M.; Gravidahl, J.T. Discrete state-space model to solve the unit commitment and economic dispatch problems. *Energy Systems* **2017**, *8*, 525–547.
- Zhang, C.; Xu, Y.; Dong, Z.Y.; Ma, J. Robust Operation of Microgrids via Two-Stage Coordinated Energy Storage and Direct Load Control. *IEEE Transactions on Power Systems* **2017**, *32*, 2858–2868.
- Morais, H.; Kádár, P.; Faria, P.; Vale, Z.A.; Khodr, H.M. Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming. *Renewable Energy* **2010**, *35*, 151–156.
- Chen, Y.-H.; Lu, S.-Y.; Chang, Y.-R.; Lee, T.-T.; Hu, M.-C. Economic analysis and optimal energy management models for microgrid systems: A case study in Taiwan. *Applied Energy* **2013**, *103*, 145–154.
- Luna, A.C.; Diaz, N.L.; Graells, M.; Vasquez, J.C.; Guerrero, J.M. Mixed-Integer-Linear-Programming-Based Energy Management System for Hybrid PV-Wind-Battery Microgrids: Modeling, Design, and Experimental Verification. *IEEE Transactions on Power Electronics* **2017**, *32*, 2769–2783.
- de la Hoz, J.; Martín, H.; Alonso, A.; Carolina Luna, A.; Matas, J.; Vasquez, J.C.; Guerrero, J.M. Regulatory-framework-embedded energy management system for microgrids: The case study of the Spanish self-consumption scheme. *Applied Energy* **2019**, *251*, 113374.

15. Jeong, B.-C.; Shin, D.-H.; Im, J.-B.; Park, J.-Y.; Kim, Y.-J. Implementation of Optimal Two-Stage Scheduling of Energy Storage System Based on Big-Data-Driven Forecasting—An Actual Case Study in a Campus Microgrid. *Energies* **2019**, *12*, 1124.
16. Gearhart, J.L.; Adair, K.L.; Detry, R.J.; Durfee, J.D.; Jones, K.A.; Martin, N. Comparison of open-source linear programming solvers. *Sandia National Laboratories, SAND2013-8847* **2013**.
17. Pfenninger, S.; DeCarolis, J.; Hirth, L.; Quoilin, S.; Staffell, I. The importance of open data and software: Is energy research lagging behind? *Energy Policy* **2017**, *101*, 211–215.
18. Groissböck, M. Are open source energy system optimization tools mature enough for serious use? *Renewable and Sustainable Energy Reviews* **2019**, *102*, 234–248.
19. Weibezahn, J.; Kendzior, M. Illustrating the Benefits of Openness: A Large-Scale Spatial Economic Dispatch Model Using the Julia Language. *Energies* **2019**, *12*, 1153.
20. Meindl, B.; Templ, M. Analysis of commercial and free and open source solvers for linear optimization problems. *Eurostat and Statistics Netherlands within the project ESSnet on common tools and harmonised methodology for SDC in the ESS* **2012**, *20*.
21. Kersting, W.H. Radial distribution test feeders. *IEEE Transactions on Power Systems* **1991**, *6*, 975–985.
22. Vergara, P.P.; López, J.C.; Rey, J.M.; da Silva, L.C.P.; Rider, M.J. Energy Management in Microgrids. In *Microgrids Design and Implementation*, Zambroni de Souza, A.C., Castilla, M., Eds. Springer International Publishing: Cham, 2019; 10.1007/978-3-319-98687-6_7pp. 195–216.
23. Kim, Y.; Del-Rosario-Calaf, G.; Norford, L.K. Analysis and Experimental Implementation of Grid Frequency Regulation Using Behind-the-Meter Batteries Compensating for Fast Load Demand Variations. *IEEE Transactions on Power Systems* **2017**, *32*, 484–498.
24. Feng, C.; Cui, M.; Hodge, B.-M.; Lu, S.; Hamann, H.F.; Zhang, J. Unsupervised Clustering-Based Short-Term Solar Forecasting. *IEEE Transactions on Sustainable Energy* **2019**, *10*, 2174–2185.
25. Bae, K.Y.; Jang, H.S.; Sung, D.K. Hourly Solar Irradiance Prediction Based on Support Vector Machine and Its Error Analysis. *IEEE Transactions on Power Systems* **2016**, *10.1109/tpwrs.2016.2569608*, 1–1.
26. Tuohy, A.; Meibom, P.; Denny, E.; O'Malley, M. Unit Commitment for Systems With Significant Wind Penetration. *IEEE Transactions on Power Systems* **2009**, *24*, 592–601.
27. Ruiz, P.A.; Philbrick, C.R.; Zak, E.; Cheung, K.W.; Sauer, P.W. Uncertainty Management in the Unit Commitment Problem. *IEEE Transactions on Power Systems* **2009**, *24*, 642–651.
28. Ruiz, P.A.; Philbrick, C.R.; Sauer, P.W. Modeling Approaches for Computational Cost Reduction in Stochastic Unit Commitment Formulations. *IEEE Transactions on Power Systems* **2010**, *25*, 588–589.
29. Papavasiliou, A.; Oren, S.S.; Rountree, B. Applying High Performance Computing to Transmission-Constrained Stochastic Unit Commitment for Renewable Energy Integration. *IEEE Transactions on Power Systems* **2015**, *30*, 1109–1120.
30. Zhang, C.; Xu, Y.; Dong, Z.Y.; Ma, J. Robust operation of microgrids via two-stage coordinated energy storage and direct load control. *IEEE Transactions on Power Systems* **2016**, *32*, 2858–2868.
31. Wang, Z.; Chen, B.; Wang, J.; Begovic, M.M.; Chen, C. Coordinated Energy Management of Networked Microgrids in Distribution Systems. *IEEE Transactions on Smart Grid* **2015**, *6*, 45–53.
32. Nemati, M.; Braun, M.; Tenbohlen, S. Optimization of unit commitment and economic dispatch in microgrids based on genetic algorithm and mixed integer linear programming. *Applied Energy* **2018**, *210*, 944–963.
33. Kwon, S.; Park, J.; Kim, Y. Optimal V2G and Route Scheduling of Mobile Energy Storage Devices Using a Linear Transit Model to Reduce Electricity and Transportation Energy Losses. *IEEE Transactions on Industry Applications* **2020**, *56*, 34–47.
34. Amoasi Acquah, M.; Kodaira, D.; Han, S. Real-Time Demand Side Management Algorithm Using Stochastic Optimization. *Energies* **2018**, *11*, 1166.
35. Electric Rates Table. Available online: <http://cyber.kepco.co.kr/ckepco> (accessed on 20 February 2020)
36. Open Innovation Bigdata Center. Available online: <http://oibc.postech.ac.kr> (accessed on 20 February 2020)
37. Python. Available online: <https://www.python.org> (accessed on 20 February 2020)
38. PuLP. Available online: <https://www.coin-or.org/PuLP/> (accessed on 20 February 2020)
39. CBC. Available online: <https://projects.coin-or.org/Cbc> (accessed on 20 February 2020)
40. GLPK. Available online: <https://www.gnu.org/software/glpk> (accessed on 20 February 2020)
41. Gurobi. Available online: <https://www.gurobi.com/> (accessed on 20 February 2020)
42. CPLEX. Available online: <https://www.ibm.com/analytics/cplex-optimizer> (accessed on 20 February 2020)

43. Comparison of Programming Languages. Available online: <https://modelingguru.nasa.gov/docs/DOC-2625/diff?secondVersionNumber=19> (accessed on 20 February 2020)



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).