Energy Reports 8 (2022) 9777–9786

Contents lists available at ScienceDirect

Energy Reports

journal homepage: www.elsevier.com/locate/egyr

Research paper

Accelerated optimal maintenance scheduling for generation units on a truthful platform

Jianfeng Fu [a,*], Alfredo Núñez [b], Bart De Schutter [a]

[a] Delft Center for Systems and Control, Delft University of Technology, Building 34, Mekelweg 2, Delft, The Netherlands
[b] Department of Engineering Structures, Delft University of Technology, Gebouw 23, Stevinweg 1, Delft, The Netherlands

ARTICLE INFO

Article history:
Received 28 March 2022
Received in revised form 27 July 2022
Accepted 1 August 2022
Available online xxxx

Keywords:
Maintenance of generation units
Benders decomposition
Valid inequality
Truthful maintenance platform

ABSTRACT

Maintenance of generation units is a measure to ensure the reliability of power systems. In this paper, a novel blockchain-based truthful condition-based maintenance of generation units (T-CBMGU) platform is proposed to innovate and upgrade state-of-the-art CBMGU. In addition, two valid inequalities are proposed to accelerate the convergence speed of Benders decomposition in maintenance scheduling process. The proposed valid inequalities are formulated based on technical/physical analysis and greedy-based heuristic initialization. More specifically, for data acquisition and failure rate diagnosis/prognosis processes, T-CBMGU can ensure the immutability of the collected operational data. In this way, the influence of tampered data on the diagnosis/prognosis results in state-of-the-art CBMGU can be reduced. For maintenance scheduling and bidding to change scheduled time slot processes, in state-of-the-art CBMGU, the decision makers, i.e., independent system operators (ISOs), may not be trusted. However, in T-CBMGU, the scheduling and bidding processes are implemented automatically via smart contracts rather than by the ISOs; as such, incentives to manipulate data can be avoided. Finally, regarding performance of maintenance actions, in contrast to state-of-the-art CBMGU, the implementation process can be truthfully recorded by the T-CBMGU platform, which facilitates backtracking of responsibility. Then, the T-CBMGU platform and the valid inequalities are tested for the IEEE 300-bus power system. Furthermore, cases with tampered data and distrust caused by fairness manipulation are simulated to show the importance of using T-CBMGU. Finally, the Benders decomposition algorithm with valid inequalities is compared with other solvers to demonstrate its fast convergence speed.

1. Introduction

Maintenance is a necessary measure for ensuring the reliability of power systems (Fu et al., 2020; Toubeau et al., 2021). A maintenance decision-making strategy based on the evaluation of the condition of the components is called condition-based maintenance. Mainly, condition-based maintenance of power systems can be categorized as short-term when determining the maintenance schedule for the upcoming days or weeks, mid-term for the upcoming several weeks or months, and long-term for the upcoming several months or years (Alvarez-Alvarado and Jayaweera, 2020). The scope of this paper is long-term condition-based maintenance of generation units.

In deregulated power transmission networks, the generation companies, equipment manufacturers, transmission companies, and the independent system operators (ISOs) may be different entities. Sometimes these entities have to share their data for cooperative tasks. These tasks are essential for operating the transmission network safely and economically, e.g., condition-based maintenance of generation units (CBMGU). However, since the shared data for cooperative tasks may be tampered with, other entities may receive tampered data that may influence the final outcomes of the cooperative tasks. Thus, to make the entities trust the shared data in the CBMGU tasks, data security should be ensured as the most basic requirement for the cooperative tasks. Consequently, this paper proposes a truthful maintenance platform for CBMGU to ensure data security among all the processes of CBMGU.

Generally, long-term state-of-the-art CBMGU mainly encompasses five processes: acquisition of operational data, failure rate diagnosis/prognosis, maintenance scheduling, bidding to change the scheduled time slots,[1] and performance of the maintenance actions. All five processes of state-of-the-art CBMGU have been

* Corresponding author.
E-mail address: J.Fu-1@tudelft.nl (J. Fu).

[1] While some authors do not consider such a rescheduling process, the current paper does include a rescheduling step, just as Wang et al. (2016) and Feng and Wang (2009).

https://doi.org/10.1016/j.egyr.2022.08.002
2352-4847/

Table 1
Comparison between state-of-the-art CBMGU and the proposed T-CBMGU.

Processes	State-of-the-art CBMGU	Proposed T-CBMGU
1. Acquisition of operational data 2. Failure rate diagnosis/prognosis 3. Performance of maintenance actions	1. Stored data can be tampered with 2. Stored data may be lost or deleted. 3. Sharing data with low efficiency. 4. Responsibility cannot be easily backtracked.	1. Stored data are immutable and trustworthy. 2. Loss of stored data is unlikely. 3. Sharing data with efficiency. 4. Responsibility can be easily backtracked.
1. Scheduling of maintenance actions 2. Bidding to change scheduled time slots	1. ISOs schedule the maintenance actions. 2. Actions that compromise fairness may occur. 3. Inefficient decision making processes due to manual work.	1. Decisions making via smart contract. 2. No actions that compromise the fairness. 3. Less manual work.

widely studied in the literature (Wang et al., 2016; Feng and Wang, 2009; Yildirim et al., 2016; Fallahi et al., 2021; Hsu et al., 2020; Moinian and Ameli, 2020; Rokhforoz et al., 2021; Ogieva et al., 2015).

Among the five processes of state-of-the-art CBMGU, three processes are related to data storage and sharing, i.e., the acquisition of operational data, the failure rate diagnosis/prognosis, and the performance of maintenance actions. In these three processes, the collected operational data, the failure rate diagnosis/prognosis results, and the maintenance action performance logs are recorded in the data storage center. However, using a data storage center may face these drawbacks. First, the data can be tampered with by hackers or personnel of the data storage center. Furthermore, the stored data may be lost or physically eliminated. Tampered and lost data can influence the diagnosis/prognosis results and the trained learning-based failure rate prediction model. Second, the identity verification process for data sharing is characterized by a low efficiency, especially due to manual verification. Third, since the data can be tampered with and lost, the data are hard to backtrack, as is the responsibility of, e.g., the maintenance implementer.

The other two processes among the five relate to decision making, i.e., scheduling of maintenance actions and bidding to change scheduled time slots. The decision making processes are managed by the ISOs. In the power system maintenance literature and in industry, these ISOs are assumed to be trusted. However, it cannot be guaranteed that ISOs can schedule maintenance actions and hold the bidding processes totally fairly. GENCOs may manipulate the fairness of the bidding processes by e.g., bribes and blackmails, and tamper with the bidding prices of other GENCOs to obtain benefits. Moreover, it is time-costly for a human team to, e.g., manually verify the identity of the GENCOs, especially when many GENCOs are involved.

Table 1 presents the drawbacks of state-of-the-art CBMGU. In this paper, a blockchain-based truthful CBMGU (T-CBMGU) platform is proposed to tackle these drawbacks. A blockchain is a chain of blocks that are linked via cryptography (Lycklama à Nijeholt et al., 2017). A block contains the data to be stored and hash points for encryption. Important features of blockchains include immutable data, no centralized authority, and traceable data (Chatterjee and Chatterjee, 2017).

T-CBMGU has three *main* advantages over state-of-the-art CBMGU. First, since data are stored on the nodes (participants) of the blockchain without referring to a data storage center, it is difficult to tamper with the stored data (Jiang et al., 2020). If the data on one node are tampered with, other nodes can verify the tampered data. Consequently, the stored data in the blocks of the blockchain are immutable. Second, data sharing can be more efficient since time-consuming processes, such as, identity verification, can be performed automatically. Furthermore, because of the immutability of the data and asymmetric cryptography (see Remark 1 in Section 3), data sharing is secure. Third, the scheduling and bidding processes can be implemented by smart contracts without involving a third party (Yu et al., 2021; Sharma et al., 2019). The smart contract is designed by the participants of the blockchain to automatically drive decision making processes. Thus, actions that compromise fairness can be avoided.

Moreover, the maintenance scheduling problem is a mixed-integer quadratic programming (MIQP) problem that is time-costly to solve. In the literature, Benders decomposition is widely leveraged for solving mixed-integer programming problems, especially when after fixing the integer variables, the remaining problem is convex (Helseth et al., 2018). In the literature, to tackle the slow convergence of Benders decomposition, acceleration methods, e.g., valid inequalities (Rodriguez et al., 2018), approximation (e.g., rounding Nowak, 2005, relaxation induction Wang et al., 2015, and outer approximation Alizadeh et al., 2021) have been studied. Among them, valid inequalities can efficiently accelerate the convergence process of Benders decomposition. Valid inequalities are usually designed for specified problems based on technical/physical analysis of the given problem. Regarding the use of valid inequalities in the maintenance scheduling of generation units, a methodology is proposed in Rodriguez et al. (2018) for hydrogenerators. The proposed valid inequalities are tailored for hydrogenerators. Thus, this paper proposes two valid inequalities tailored for accelerating Benders decomposition for CBMGU in power systems.

The contributions of the current paper are:

- A truthful maintenance platform is proposed for generation units in power systems. By using the proposed T-CBMGU platform, the stored data can be immutable, the data sharing can be efficient and secure, and the decision-making can be fair (see Table 1 for details).
- Maintenance scheduling problems are time-costly MIQP problems. Thus, two novel dedicated valid inequalities based on technical/physical analysis and greedy-based heuristic initialization are proposed for accelerating the convergence speed of Benders decomposition.

The remainder of this paper is organized as follows. Section 2 reviews the literature on the relevant topics of this paper. Section 3 explains the processes of state-of-the-art CBMGU. Section 4 describes the T-CBMGU platform. Section 5 explains the blockchain implementation in the T-CBMGU platform. In Section 6, a Benders decomposition algorithm with valid inequalities is proposed. In Section 7, a case study based on the IEEE 300-bus system is presented to show the importance of the proposed T-CBMGU by comparing its performance with that of state-of-the-art CBMGU that is affected by fairness manipulations and tampered data. Finally, in Section 8, conclusions are presented and further research is discussed.

2. Related work

In the literature, blockchain technology has been widely applied in various fields, e.g., truthful market design (Antal et al., 2021; Yapa et al., 2021). For example, in Esmat et al. (2021), Ethereum, a blockchain platform is leveraged in a decentralized power market. The prosumers can trade energy peer-to-peer conveniently via the smart contract of Ethereum without involving a third party. In Shaikh et al. (2022), a secure blockchain-based platform is designed for educational credential evaluation. The academic credentials are generated, verified, and validated via this platform. In Butt et al. (2022), blockchain is applied to sharing medical service records between different clinical jobs. In Mrissa et al. (2022), an edge computing platform is designed for decentralized household air quality monitoring devices for ensuring data security. In Ling et al. (2019), a blockchain-based radio access network is proposed for truthful network management and authentication. In Strepparava et al. (2022), the resources require for leveraging blockchain in decentralized local energy markets are studied. In Xie et al. (2022), a transaction platform for the energy storage market is proposed. By using blockchain technology, the platform can be secure and transparent.

However, there are only a few articles in the literature on leveraging blockchain in the maintenance industry. For example, in Chang et al. (2021), a knowledge-sharing platform is designed based on blockchain for maintaining a honing machine system with multiple components. The degradation knowledge of the components can then be shared securely. In Aleshi et al. (2019), blockchain is applied to formulate an aircraft maintenance logbook that cannot be tampered with or destroyed. In Abbas et al. (2020), blockchain is applied to the maintenance of rolling stock, and the truthfulness of the business logic and data is enhanced. All these articles in the literature use blockchain in maintenance for truthful data storing and sharing. However, the use of blockchain in CBMGU has not yet been studied. Moreover, truthful data storage and sharing are not sufficient for CBMGU since CBMGU also involves decision-making processes in which actions that compromise fairness, e.g., fairness manipulation, may occur. Thus, in this paper, to address the drawbacks of state-of-the-art CBMGU, the T-CBMGU platform is proposed and designed for all five CBMGU processes.

Moreover, the maintenance scheduling problem is an MIQP problem that is time-costly to solve. In the literature, Benders decomposition is widely leveraged for solving mixed-integer programming problems, especially when after fixing the integer variables, the remaining problem is convex (Helseth et al., 2018). In the literature, to tackle the slow convergence of Benders decomposition, acceleration methods, e.g., valid inequalities (Rodriguez et al., 2018), approximation (e.g., rounding Nowak, 2005, relaxation induction Wang et al., 2015, and outer approximation Alizadeh et al., 2021) have been studied. Among them, valid inequalities can efficiently accelerate the convergence process of Benders decomposition. Valid inequalities are usually designed for specified problems based on technical/physical analysis of the given problem. Regarding the use of valid inequalities in the maintenance scheduling of generation units, a methodology is proposed in Rodriguez et al. (2018) for hydrogenerators. The proposed valid inequalities are tailored for hydrogenerators. Thus, this paper proposes two valid inequalities tailored for accelerating Benders decomposition for CBMGU in power systems.

3. State-of-the-art CBMGU platform

The work processes of state-of-the-art CBMGU are shown in Fig. 1. In Fig. 1, the labels in yellow boxes identify the processes; the labels in blue boxes indicate the categories of data that are stored in the data banks; and the arrows represent the input or output directions of the data flows. Regarding the data banks used to store the data, the data banks in red represent the stored data that are related to one generation unit. Maintenance of generation units usually involves multiple generation units. However, for simplification, data banks for other generation units in CBMGU are not included in Fig. 1. CBMGU is implemented periodically, and each period of which the length is several months for long-term CBMGU corresponds to a time step. Regarding maintenance scheduling, this paper adopts a receding horizon mechanism. Maintenance scheduling is performed for one prediction horizon that always includes multiple time steps, but only the maintenance actions of the scheduling results for the first time step are performed; subsequently, one proceeds to the next time step. The duration of maintenance actions of generation units is expressed in time slots of the long-term CBMGU (i.e., weeks). In Fig. 1, only the work processes for one time step are shown since in other time steps, the work processes repeat.

In Fig. 1, five processes are shown, i.e., acquisition of operational data, failure rate diagnosis/prognosis, maintenance scheduling, bidding to change the scheduled time slots, and performance of the maintenance actions.

In the first two processes, operational data of the components, e.g., generators and turbines, are collected by sensors to obtain the failure rates and to train the failure rate prediction models of the components (Yildirim et al., 2016; Fallahi et al., 2021; Hsu et al., 2020). For example, operational data are collected by sensors and sent to a central data hub (Yildirim et al., 2016). Afterwards, the failure rate is predicted based on the data in the central data hub and Bayesian learning. In Fallahi et al. (2021), a sensor-driven method and a Bayesian model are applied to predict the remaining life of generators in microgrids. In Hsu et al. (2020), operational data and historical data are leveraged to train predictive degradation models for wind turbines using random forest and decision tree algorithms.

In the third process, the maintenance actions are scheduled by a centralized organization, i.e., an independent system operators (ISO). The ISO aims at maximizing the generation benefits and/or minimizing the maintenance costs for all generation units it manages while ensuring power system reliability (Moinian and Ameli, 2020; Rokhforoz et al., 2021).

In the fourth process, since the scheduled maintenance actions in the third process (overall optimal schedule) may conflict with the individual benefit of the owners of generation units, i.e., generation companies (GENCOs), a bidding process for GENCOs to change their scheduled time slots is managed by the ISO (Wang et al., 2016; Feng and Wang, 2009).

In the fifth process, the maintenance logs, the material consumption situation, and the implementer information, among other information, are recorded for management and to keep the know-how in the company (Ogieva et al., 2015).

In these processes of state-of-the-art CBMGU, the data banks may refer to data recorded on the paper lists, digital lists, or other data formats. The data banks may be managed by some of the entities involved in the maintenance processes, e.g., the ISO, or data storage companies. Since the data stored in data banks may be tampered with by cyber or physical attacks, the stored data may not be trustworthy. In addition, since the decision-making processes, i.e., maintenance scheduling and bidding to change the scheduled time slots, are implemented by the ISOs, the results of the decision-making processes may be influenced by, e.g., fairness manipulation, etc. Thus, the decision-making results also may not be trustworthy.

4. T-CBMGU platform

To make the stored data and the results of decision-making processes trustworthy, we propose a blockchain-based T-CBMGU. The work processes of the proposed T-CBMGU platform for one time step are illustrated in Fig. 2.

In Fig. 2, at the beginning of the period corresponding to a given time step, the operational data are collected by the sensors on the components of generation units, e.g., exciters and turbines. For each component of each generation unit, a block for storing the operating data of this component is built and validated in the blockchain.

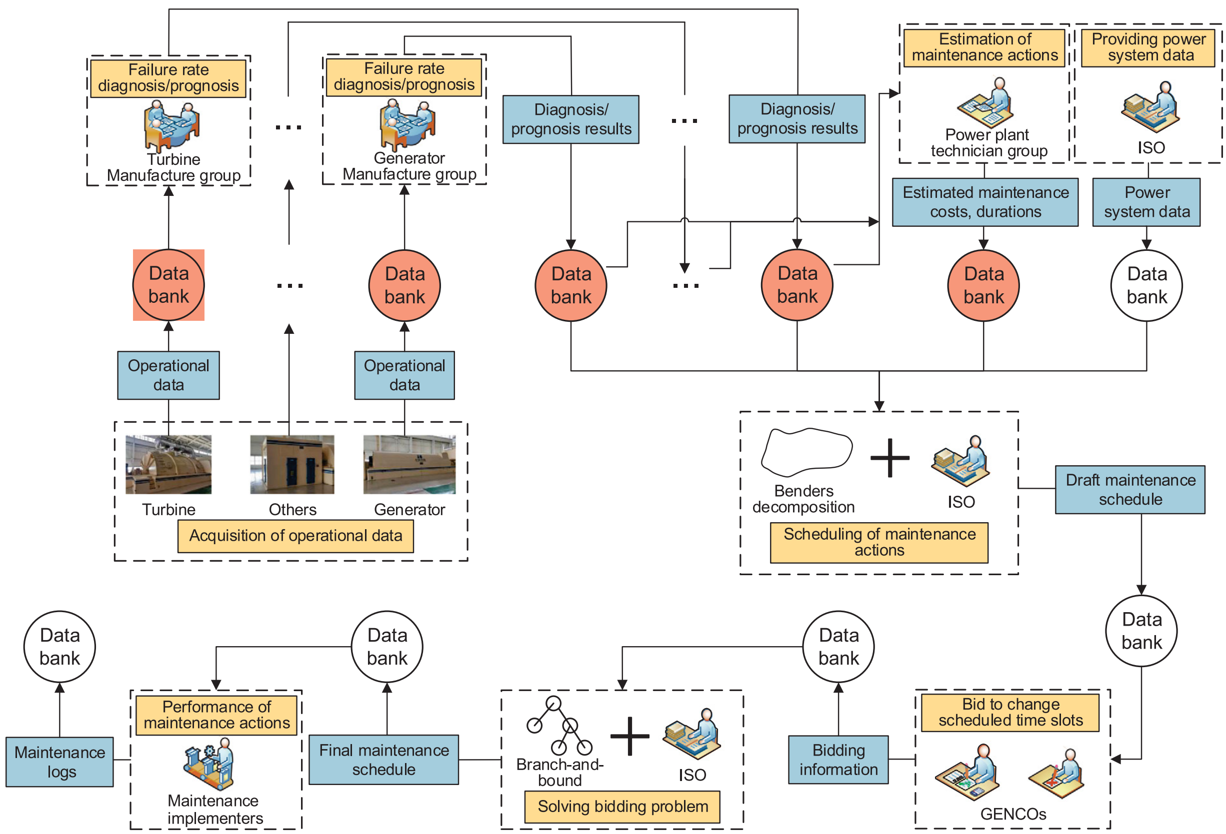


Fig. 1. Work processes of the state-of-the-art CBMGU platform for one time step.

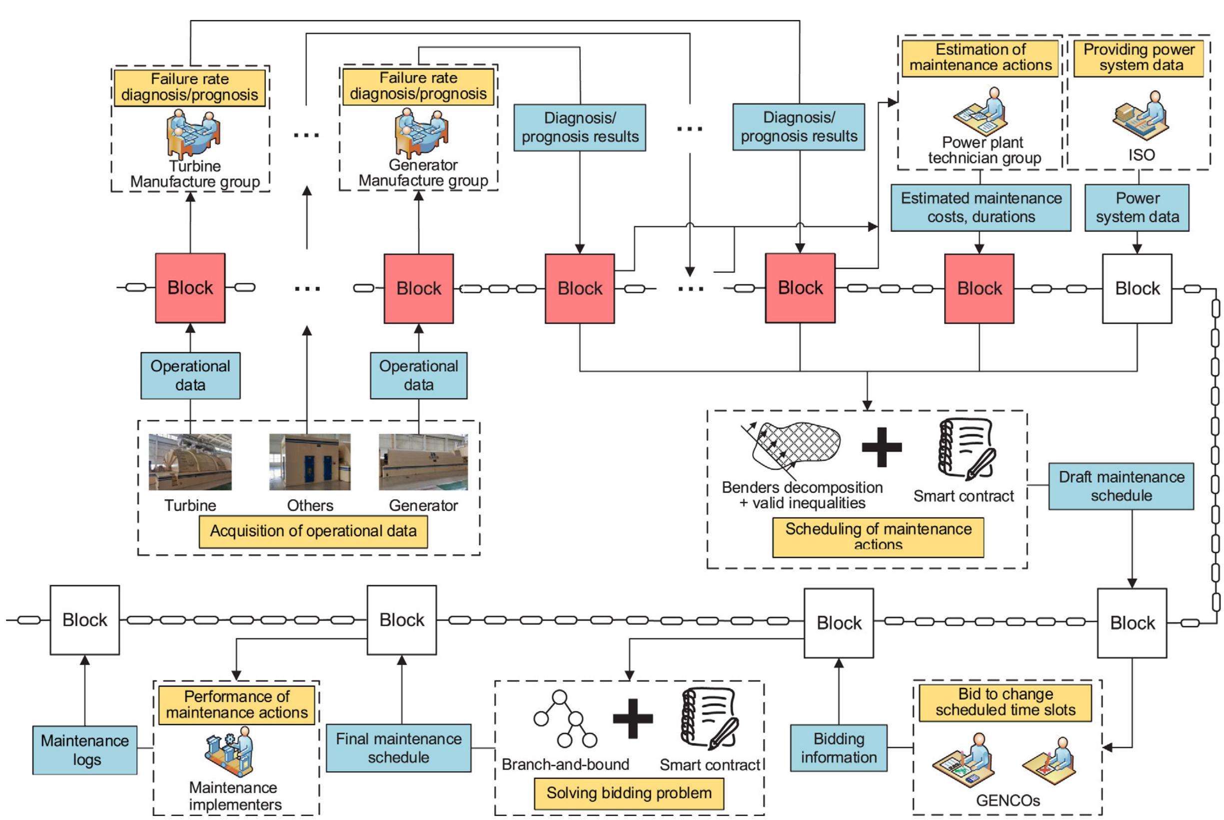


Fig. 2. Work processes of the T-CBMGU platform for one time step.

After that, the operational data in the blockchain block for each component of each generation unit are extracted by the corresponding manufacturing expert groups who provide or manufacture the component. The manufacturing expert groups may belong to the providers and/or the manufacturers of the components of the generation units. They use their knowledge, numerical models, and/or trained data-driven models to evaluate the failure rates of the components. Then, the block of the diagnosis/prognosis results for each component is built and validated.

Afterwards, the power plant technician groups estimate the maintenance costs and the maintenance durations according to the data extracted from the blocks of operational data and failure rate diagnosis/prognosis results. The estimation of the maintenance costs is based on, e.g., the replacement costs of the components, the costs for performing maintenance actions, and the outsourcing fee (if applicable). The estimation of the maintenance duration is based on, e.g., the arrangement and the internal structure of the generation units.

The block of the power system data is built by ISOs, including the load, the reserved energy level, the electricity prices, and the breakdown penalty fee.

The maintenance actions will be scheduled to minimize the overall maintenance cost and to maximize the overall benefits for all the generation units. Additionally, the scheduling problems include data extracted from previously built blocks, such as the failure rate diagnosis/prognosis data, the evaluation data of maintenance costs and durations, and the power system data as parameters. Since this paper formulates *maintenance scheduling problem* as an MIQP problem, a Benders decomposition algorithm with acceleration techniques is proposed for solving the problems more efficiently. The smart contract, rather than ISOs, implements this process automatically. Afterwards, the determined optimal maintenance schedule is stored in one block.

Then, the GENCOs extract the optimal maintenance plan from the block. According to scheduled time slots for performing maintenance actions, if the GENCOs are not satisfied with their scheduled maintenance actions, they can join in the bidding process to change their scheduled time slots. The GENCOs who intend to join the bidding process should provide their bidding information, including the time slots in which they intend to maintain their generation units and the bidding price the GENCOs intend to pay for changing their scheduled time slots. Then, each GENCO builds a block to store the bidding information (in Fig. 1, the blocks of GENCOs are integrated as one block for saving space).

By extracting the bidding information, the final schedule for performing maintenance actions is determined with the objective of maximizing the total amount of money bid by the GENCOs. Then the ISOs will use the money to improve, e.g., the reliability of the power system (Wang et al., 2016). The formulated *bidding problems* are mixed-integer linear programming problems that can be solved efficiently by using the branch-and-bound solvers. Similar to maintenance scheduling, bidding to change scheduled time slots is also implemented automatically via the smart contract. Then, the final maintenance schedule is stored in one block.

Finally, by extracting the final schedule, the maintenance implementer perform maintenance actions on their corresponding generation units according to the final schedule. While performing the maintenance actions, the maintenance logs are recorded, including the materials that are used, information on the implementer, on-the-spot measurements such as video and photos, and information on new components for replacement. Then, blocks are built to store the maintenance logs.

5. Implementation of blockchain

A blockchain can be fully public or permission-based (Jiang et al., 2021). Since the data stored in the blocks of T-CBMGU, e.g., failure rates of generation units, can be considered sensitive information, a permission-based blockchain is preferred.

In the T-CBMGU platform, the participants include the GENCOs, the providers or manufacturers of the components, the power plant technician groups, the ISO, and the maintenance implementers. When a new GENCO is founded or an existing GENCO goes bankrupt, the ISO should verify the identity of the new GENCO for participation or eliminate the bankrupted GENCO from the list of participants in the T-CBMGU platform.

In Fig. 1, the participants extract the data from the built blocks. After completing their individual tasks (e.g., failure rate diagnosis/prognosis), they build new blocks to store their obtained results. The steps for building a block are as follows:

Step I In one process, the participant authenticates and decrypts the block(s) that store the data that are required by the participant (see Remark 1). Then, the participant extracts the data and starts its tasks, e.g., diagnosis.

Step II The participant packages the data that it intends to store in the block, which are usually the results of tasks obtained in Step I, via a hash function. Then, it builds a block and encrypts the block.

Step III The built block is broadcast to all the participants.

Step IV All the participants validate the block by reaching consensus with, e.g., the proof-of-work.

Step V The validated block is linked to the blockchain and stored distributively.

Remark 1. Encryption/decryption and authentication can be done by asymmetric cryptography. In asymmetric cryptography, two kinds of keys, i.e., a public key and a private key, are used for encryption/decryption and authentication. The public key is published to all the participants, while the private key is kept only by the participant itself.

Regarding encryption/decryption, when e.g., the owner of a generation unit intends to store the operational data of a component into a block, the owner encrypts the block using the public key of the manufacturing expert group, and the manufacturing expert group can decrypt the block using its private key to extract the operating data of the component. By doing so, except for the owner of the generation unit and the manufacturing expert group of the component, other participants cannot decrypt the block to extract the operational data of the component. Hence, data privacy is guaranteed.

To determine whether, e.g., the extracted data truly originate from the corresponding generation unit, authentication can also be implemented by asymmetric cryptography. The owner of the generation unit signs on the block with stored operational data by using its private key. Then, the manufacturer expert group can authenticate the signature by using the public key of the owner of the generation unit.

6. Maintenance scheduling problem and benders decomposition with valid inequalities

In T-CBMGU, two decision making problems are solved, i.e., *maintenance scheduling problem* and *bidding problem*. This section focuses on the formulation and solution process of the maintenance scheduling problem. The formulation of the bidding problem is discussed in Wang et al. (2016). Bidding problems are mixed-integer-linear programming problems that can be solved efficiently by commercial solvers, e.g., CPLEX.

6.1. Problem formulation for maintenance scheduling

The objective of maintenance scheduling is to minimize the overall costs and to maximize the overall benefits of the generation units. Consequently, the objective function is:

$$\min_{\delta_{g,k},\Delta_{g,k},\sigma_{g,k},\varsigma_g,P^{\mathrm{G}}_{g,k}} \sum_{g\in\mathcal{G}}\sum_{k\in\mathcal{K}} c^{\mathrm{m}}_g \delta_{g,k}/\tau_g + \sum_{g\in\mathcal{G}} p_g c^{\mathrm{p}}_g \Big(1- \sum_{k\in\mathcal{K}} \delta_{g,k}/\tau_g\Big) + \sum_{g\in\mathcal{G}}\sum_{k\in\mathcal{K}} \Big(\sigma_{g,k} c^{\mathrm{st}}_{g,k} + \Delta_{g,k}\Big(c^{\mathrm{g}}_{2,g}(P^{\mathrm{G}}_{g,k})^2 + c^{\mathrm{g}}_{1,g}P^{\mathrm{G}}_{g,k} + c^{\mathrm{g}}_{0,g} - c^{\mathrm{e}}P^{\mathrm{G}}_{g,k}\Big)\Big) \tag{1a}$$

where $\mathcal{G}$ is the set of generation units, $\mathcal{K}$ is the set of time slots in one prediction window, $\delta_{g,k}$ equals 1 if the maintenance action on generation unit g is performed in time slot k and equals 0 otherwise, c^{m}_g is the maintenance cost for generation unit g, τ_g is the maintenance duration for generation unit g, p_g is the failure rate of generation unit g. In this paper, the health condition of the generation units is described by a failure rate. In (1a), c^{p}_g is the penalty fee of a failure on generation unit g, $\sigma_{g,k}$ equals 1 if generation unit g starts up in time slot k and equals 0 otherwise, ς_g equals to 1 if generation unit g is maintained, $c^{\mathrm{st}}_{g,k}$ is the start-up cost for generation unit g in time slot k, $\Delta_{g,k}$ equals 1 if generation unit g is connected to the grid in time slot k and equals 0 otherwise, $c^{\mathrm{g}}_{2,g}$, $c^{\mathrm{g}}_{1,g}$ and $c^{\mathrm{g}}_{0,g}$ are the coefficients of the power generation cost of generation unit g, c^{e} is the electricity price for generating power, and $P^{\mathrm{G}}_{g,k}$ is the power generated by generation unit g in time slot k. The first two terms in (1a) express that if a maintenance action is performed on one generation unit, the generation unit is recovered, and the failure rate of the generation unit becomes zero. The terms in (1a) represent the overall maintenance costs of the generation units, the penalty fees for failures, and the balance of the start-up costs, generation costs, and generation benefits. The constraints are:

$$1-\delta_{g,k} \geq \Delta_{g,k},\ \forall g\in\mathcal{G},\ \forall k\in\mathcal{K} \tag{1b}$$

$$\Delta_{g,k}P^{\mathrm{G-}}_g \leq P^{\mathrm{G}}_{g,k} \leq \Delta_{g,k}P^{\mathrm{G+}}_g,\ \forall g\in\mathcal{G},\ \forall k\in\mathcal{K} \tag{1c}$$

$$\sigma_{g,k} \geq \Delta_{g,k} - \Delta_{g,k-1},\ \forall g\in\mathcal{G},\ \forall k\in\mathcal{K} \tag{1d}$$

$$\sum_{k\in\mathcal{K}} \delta_{g,k} = \varsigma_g\tau_g,\ \forall g\in\mathcal{G} \tag{1e}$$

$$\sum_{k\in\mathcal{K}} |\delta_{g,k}-\delta_{g,k-1}| \leq 2,\ \forall g\in\mathcal{G} \tag{1f}$$

$$P^{\mathrm{D}}_k = \sum_{g\in\mathcal{G}} \Delta_{g,k}P^{\mathrm{G}}_{g,k},\ \forall k\in\mathcal{K} \tag{1g}$$

$$\sum_{g\in\mathcal{G}} \Delta_{g,k}P^{\mathrm{G+}}_g \geq P^{\mathrm{D}}_k + r^+,\ \forall k\in\mathcal{K} \tag{1h}$$

$$\delta_{g,k}\in\{0,1\},\ \Delta_{g,k}\in\{0,1\},\ \sigma_{g,k}\in\{0,1\},\ \varsigma_g\in\{0,1\},\ P^{\mathrm{G}}_{g,k}\geq 0,\ \forall g\in\mathcal{G},\ \forall k\in\mathcal{K} \tag{1i}$$

where (1b) indicates that if generation unit g is in maintenance, it cannot be connected to the grid. Constraint (1c) limits the power generated by the generation units, where $P^{\mathrm{G-}}_g$ and $P^{\mathrm{G+}}_g$ are the upper and lower bounds of the power generated by generation unit g. Constraint (1d) guarantees that when a previously disconnected generation unit is connected to the grid again, then there is a start-up action. Constraint (1e) represents that in each generation unit g the sum of time slots used for maintenance equals 0 if the maintenance action will not be performed (binary variable $\varsigma_g = 0$); otherwise, it equals the duration of the maintenance action of generation unit g ($\varsigma_g = 1$). Constraint (1f) represents that the maintenance actions should be performed consecutively, where $\delta_{g,0} = 0$. Constraint (1g) expresses the power balance of the power system, where P^{D}_k is the predicted load demand in time slot k. Constraint (1h) is the constraint for the reserved energy, where r^+ is the reserved energy level.

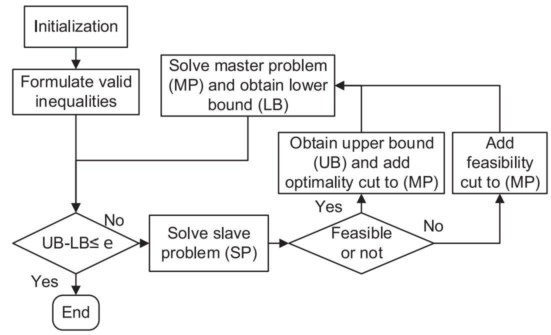


Fig. 3. Benders decomposition solving process.

The maintenance scheduling problem (1) can be formulated into an MIQP problem by using the approach of Fu et al. (2020).[2] To solve the formulated MIQP problem efficiently, in Section 4.C, two valid inequalities are proposed to accelerate the Benders decomposition solver.

6.2. Solution process based on Benders decomposition

For a mixed-integer programming problem, Benders decomposition separates the integer variables and continuous variables and solve a master problem and a slave problem separately. The solution process of Benders decomposition with the proposed valid inequalities is shown in Fig. 3. If the slave problem is feasible, an upper bound for the whole problem is obtained, and an extra constraint, called optimality cut is added to the master problem. If the slave problem is infeasible, an extra constraint, called feasibility cut is added to the master problem. Then, the master problem is solved to obtain the values of the fixed variables for the next iteration, and a lower bound is obtained. The termination condition involves the convergence of the upper and lower bounds. The general form of problem (1) is expressed as:

$$\begin{aligned}\min_{x\in(\mathbb{R}^+)^{n_x},\,y\in\{0,1\}^{n_y}} \quad & x^{\mathrm{T}}Hx + f_1^{\mathrm{T}}x + f_2^{\mathrm{T}}y \\ \text{s.t.}\quad & Ax+By\leq b\end{aligned} \tag{P}$$

In (P), since the continuous variable $P^{\mathrm{G}}_{g,k}$ and the continuous auxiliary variables $\psi_{g,k}$ are non-negative, the continuous vector $x = [P^{\mathrm{G}}_{g,k},\ \psi_{g,k}]^{\mathrm{T}}_{g\in\mathcal{G},k\in\mathcal{K}}$ is a vector of non-negative real variables, where n_x is the length of x. Besides, $y = [\delta_{g,k},\ \Delta_{g,k},\ \sigma_{g,k},\ \varsigma_g]^{\mathrm{T}}_{g\in\mathcal{G},k\in\mathcal{K}}$ is a vector of binary variables, where n_y is the length y. For a fixed y, we can define the slave problem (SP):

$$\begin{aligned}\min_{x\in(\mathbb{R}^+)^{n_x}} \quad & x^{\mathrm{T}}Hx + f_1^{\mathrm{T}}x + f_2^{\mathrm{T}}\bar{y} \\ \text{s.t.}\quad & Ax\leq b-B\bar{y}\end{aligned} \tag{SP}$$

where $\bar{y}$ is the fixed y that is determined by the initialization or the solution of the master problem of the last iteration. Since (SP) is a quadratic programming problem, the solving process can be driven by, e.g., interior-point-convex algorithm, so as to verify feasibility judgment and to return the Lagrangian multipliers

[2] Regarding (1a), since $\Delta_{g,k}(P^{\mathrm{G}}_{g,k})^2 = (\Delta_{g,k}P^{\mathrm{G}}_{g,k})^2 = \psi^2_{g,k}$, where $\psi_{g,k}$ is a continuous auxiliary variable, the objective function is in quadratic form.

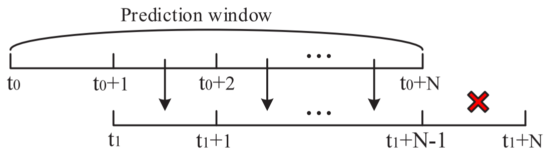


Fig. 4. Illustration of initialization mechanism.

(if (SP) is feasible). If (SP) is feasible, the obtained Lagrangian multiplier is defined as λ_m, where m is the current number of obtained optimality cuts. If (SP) is infeasible, an extreme ray μ_n is generated via a Phase I algorithm, e.g., the one in Floudas et al. (1989), where n is the current number of obtained feasibility cuts. Then, the master problem (MP) is:

$$\begin{aligned} &\min_{y\in\{0,1\}^{n_y},\eta} \quad \eta \\ &\text{s.t. } \eta \geq \lambda_i(b-By)+f_2^{\mathrm{T}}y,\ i\in\{1...m\}, \\ &\quad 0 \geq \mu_j(b-By),\ j\in\{1...n\}, \\ &\quad \text{with (VI-1) and (VI-2)} \end{aligned} \tag{1}$$

where η is an intermediate continuous variable; (VI-1) and (VI-2) are the proposed valid inequalities, which are linear constraints and which will be further explained in Section 4.C. Problem (MP) is a mixed-integer linear programming problem that can be solved by, e.g., branch-and-bound algorithm.

6.3. Formulation of valid inequalities

(1) VI-1: Since this paper adopts a receding horizon mechanism, the maintenance scheduling results of the previous time step can be used to formulate (VI-1) for the problem of the current time step. In Fig. 4, the initialization mechanism for dynamic scheduling problems with a receding horizon mechanism is illustrated. Fig. 4 assumes that a prediction window includes N time slots, where N is an integer. At the current time step, the maintenance decisions of the previous problem whose prediction window is from t_0+1 to t_0+N can be used as the initial solution from t_1 to t_1+N-1 of the current problem whose prediction window is from t_1 to t_1+N. Without loss of feasibility, no maintenance actions are performed from t_1+N-1 to t_1+N to formulate the initial solution.

However, this initial solution may not be feasible if the *currently predicted* loads from t_1 to t_1+N-1 differ from the *previously predicted* loads from t_0+1 to t_0+N. Thus, two scenarios are discussed.

First, if the currently predictive loads for the scheduling problem from t_1 to t_1+N-1 are no larger than the previously predicted loads for the problem from t_0+1 to t_0+N, the integer variables of the maintenance plan (δ, Δ, and σ) can be initialized as before. Then, the initial maintenance plan is substituted into (SP), and (SP) is solved to obtain P^{G}, and λ_0.

Second, if some of the currently predicted loads are larger than the previously predicted loads, some of the scheduled maintenance actions of the problem of the last time step cannot be performed to guarantee the reserved energy level (constraint (1 h)). To determine which maintenance actions cannot be performed, a greedy-based process is proposed:

Step 1 Define $\mathcal{P}$ as the set of time slots in which the scheduled results of the previous time step do not satisfy the reserved energy level. Rank the time slot $p\in\mathcal{P}$ from the largest reserved energy gaps (i.e., $\sum_{g\in\mathcal{G}}\Delta_{g,p}\,P_g^{\mathrm{G}+}-P_p^{\mathrm{D}}+r^+$) to the smallest.

Step 2 Select the first element $p\in\mathcal{P}$, i.e., the time slot p with the largest reserved spinning energy gaps.

Table 2
Hash values of the estimated maintenance costs and durations of a subset of the generation units.

Generation unit	Duration (week)	Cost (k$)	Hash value
1	3	104	b6586492 ... 408f1b8ceaf7
2	3	98	ff60ac92 ... 8835d6b250a1
3	2	98	d02ce89a ... 08a841147f70
4	4	106	d27be11d ... 61c460ceeb5c
5	3	110	2574f5b6 ... 8f215a519fb7

Step 3 Define $\mathcal{M}_p$ as the set of maintenance actions for p in the scheduled result of the previous problem. Then rank the elements of $\mathcal{M}_p$, according to $p_g c_g^{\mathrm{p}}-c_g^{\mathrm{m}}$, where $g\in\mathcal{M}_p$. Subsequently, keep removing the first entry in the ranked set $\mathcal{M}_p$ until (1h) is satisfied for p. Then remove p from $\mathcal{P}$.

Step 4 Repeat **Step 2** and **Step 3** until $\mathcal{P}=\emptyset$.

Step 5 Obtain δ according to the remaining maintenance actions. Let $\Delta_{g,k}=1-\delta_{g,k}$. Then, $\sigma_{g,k}=\Delta_{g,k}-\Delta_{g,k-1}$. Afterwards, substitute the obtained δ, Δ, and σ into (SP), and solve (SP) to obtain P^{G} and λ_0. End the process.

After obtaining P^{G} and λ_0 according to the applicable two scenarios, (VI-1) can be expressed as:

$$\eta \geq \lambda_0(b-By)+f_2^{\mathrm{T}}y \tag{VI-1}$$

(2) VI-2: The second valid inequality is formulated by considering the maintenance cost and the benefits. The second valid inequality is only applicable if the currently predicted loads from t_1 to t_1+N-1 are no larger than the previously predicted loads from t_0+1 to t_0+N. If generation unit g' is maintained from time slots k' to k'', then $\delta_{g',k}=1$ and $\Delta_{g',k}=0$, for $k\in\{k',\dots,k''\}$. Therefore, the cost from k' to k'' includes only the maintenance cost c_g^{m}. In comparison, if generation unit g is not maintained from k' to k'', the cost from k' to k'' includes $p_g c_g^{\mathrm{p}}$ and the balance containing the start-up costs, the power generation costs, and the benefits. The balance can always be smaller than 0 since if the balance (costs minus benefits) is larger than 0, disconnecting generation unit g (i.e., setting $\Delta_{g',k}=0$) will cause the balance to equal 0. Thus, if the penalty fee is smaller than the maintenance cost (i.e., $p_g c_g^{\mathrm{p}}<c_g^{\mathrm{m}}$), the generation unit must not be maintained:

$$\delta_{g,k}(p_g c_g^{\mathrm{p}}-c_g^{\mathrm{m}})\geq 0,\ \forall g\in\mathcal{G},\ \forall k\in\mathcal{K} \tag{VI-2}$$

7. Case study

In this case study, the IEEE-300 bus system with 69 generation units and 195 loads is investigated (Adibi, 2010). The simulation of T-CBMGU is based on Go-Ethereum, which is implemented on the GoLand platform. The hash values of the implementation of the proposed T-CBMGU via the GoLand platform are shown in Table 2. These hash values record the information of the estimated maintenance costs and durations of a subset of the generation units. As mentioned in Section 3, the information on the estimated maintenance costs and durations of a subset of the generation units are provided by the power plant technician group in the second process.

Two comparative studies are performed in this section. The first comparison is between the proposed T-CBMGU and state-of-the-art CBMGU. The second comparison is between the proposed Benders decomposition solver and the other three solvers.

7.1. Comparison between the proposed T-CBMGU and state-of-the-art CBMGU

Since data banks of state-of-the-art CBMGU may not be truthful, the data tampering and fairness manipulation may occur. To

Table 3
Comparison between state-of-the-art CBMGU and the proposed T-CBMGU.

Case	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Number of conflicts	12	21	27	14	19	21	15	8	25	10	21	15	7	12	8	15	21	10	21	20

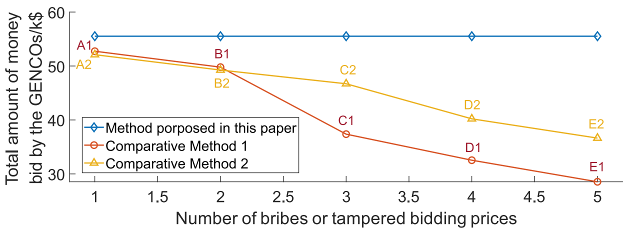


Fig. 5. Comparison between T-CBMGU and state-of-the-art CBMGU platforms with fairness manipulation or tampered data.

compare the proposed T-CBMGU platform with the state-of-the-art CBMGU platform, data tampering and fairness manipulation (i.e., bribes in this case study) are simulated in the bidding to change scheduled time slots with 7 GENCOs. On the state-of-the-art CBMGU platform, three cases (A1, B1, C1, D1, E1) with 1 to 5 bribes are simulated. In case A1, the briber with the highest bribe price fixes its preferred time slots. In case B1, based on the fixed time slots of the first briber, the briber with second-highest bribe price fixes its preferred time slots. Cases C1 to E1 are similar, but with 3 to 5 bribes. After each bribe occurs, the bidding problem is solved for feasibility checking (satisfaction of the reserved energy requirement). If feasible, then the next briber fixes its preferred time slots. If not, the briber should select other time slots to fix. Furthermore, five cases with 1 to 5 tampered bidding prices are simulated (A2, B2, C2, D2, E2). The hacker tampers with 1 to 5 bidding prices of others whose bidding prices are larger than those of his employer. The *bidding problems* (mixed-integer linear programming problems) are solved by CPLEX to obtain globally optimal solutions.

In the simulation of T-CBMGU, Fig. 5 compares the T-CBMGU and the state-of-the-art CBMGU platform, where fairness manipulation and tampering data may occur when solving *bidding problems*. In Fig. 5, comparative method 1 refers to simulations with bribes and comparative method 2 refers to simulations with tampered data. From Fig. 5, it is observed that as the numbers of bribes and tampered bidding prices increase, social welfare (i.e., total amount of money bid by the GENCOs) decreases. In cases A1 to E1, social welfare decreases by 5%, 10.32%, 32.69%, 41.39%, and 48.6%, respectively, compared to T-CBMGU. In cases A2 to E2, social welfare decreases by 6.16%, 11.29%, 15.87%, 27.58%, and 34.01%, respectively. Thus, T-CBMGU can avoid fairness manipulation and data tampering to obtain higher social welfare.

7.2. Comparison between the proposed Benders decomposition and other solvers

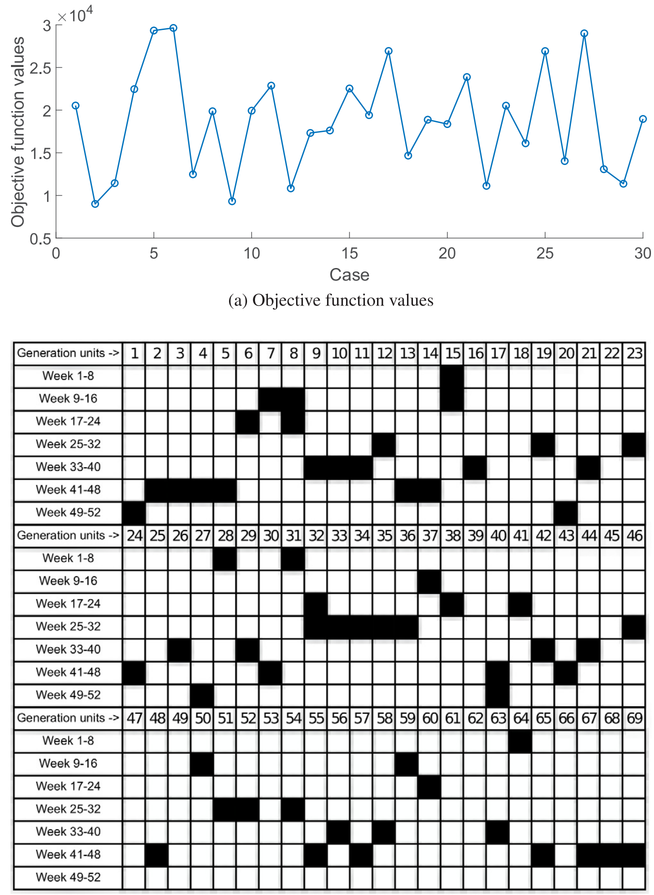
Regarding the *maintenance scheduling problems* the performance of the proposed Benders decomposition solver (BD+VIs) is evaluated by comparing it with three other *global* optimization solvers for MIQP problems. The three compared solvers include a branch-and-bound solver with *filter* sequential quadratic programming (BB+SQP) (Fletcher and Leyffer, 1998), a branch-and-bound solver with BQPD (BB+BQPD) (Fletcher, 2000), and a custom Benders decomposition solver (Rodriguez et al., 2018). Among them, BB+SQP and BB+BQPD are implemented by the Tomlab toolbox in MATLAB, while BD and the proposed BD+VIs are implemented by self-written coding scripts in MATLAB.

Moreover, to demonstrate the effectiveness of the proposed solver, 30 cases with various maintenance actions (maintenance costs and maintenance durations) and failure rates are tested. In each case, the valid inequalities are obtained using the scheduling result of the last time step according to Section 4.C. The length of one prediction window, one time step, and one time slot are 52, 13, and 1 week, respectively. To test the effectiveness of the valid inequalities under various initialization cases, 30 cases are simulated in total with different situations of conflicts between currently and previously predicted loads. From Cases 1 to 10, the currently predicted loads are the same as the previously predicted loads. In the next 10 cases, in some time slots, the currently predicted loads are smaller than the previously predicted loads, and in the other time slots, the currently predicted loads and the previously predicted loads are the same. Finally, in some of the time slots in the last 10 cases, the currently predicted loads are larger than the previously predicted loads, and in other time slots, the currently predicted loads and the previously predicted loads are the same. As presented in Table 3, in Cases 11–20, the number of conflicts represents the number of time slots in which the currently predicted loads are smaller than the previously predicted loads. In Cases 21–30, the number of conflicts represents the number of time slots in which the currently predicted loads are larger than the previously predicted loads.

In Fig. 6(a), the objective function values of 30 cases of the *maintenance scheduling problem* are shown, while Fig. 6(b) presents the maintenance scheduling results of 69 generation units in Case 1. In Fig. 6(a), the horizontal axis and vertical axis represent the objective function values in (1a) and cases respectively. In Fig. 6(b), the maintenance scheduling results of 69 generation units in Case 1 are shown. In Fig. 6(b), the boxes colored in black represent performing maintenance actions during this period, and the white boxes represent not performing any maintenance. Fig. 6(b) reflects the solution of $\Delta_{g,k}$ variables of the optimization problem (1a)–(1i). It can be observed that the maintenance actions of the generation units are to be performed over the whole prediction window, and not all the generation units are required to be maintained. That is because the reserved energy r^+ limits that the maintenance actions of all the generation units are performed in one or a few time slots.

Furthermore, since the solvers, i.e., BB+SQP, BB+BQPD, BD, and BD+VIs, can converge to the globally optimal solution, this paper compares the CPU times between the solvers, as shown in Fig. 7. The average CPU times for 30 cases for the BB+SQP, BB+BQPD, BD, and BD+VIs solvers are 54.9 min, 33.9 min, 38.7 min, and 15.1 min, respectively. Thus, the solving speed of BD is lower than those of BB+SQP and BB+BQPD, while the solving speeds of BB+SQP and BB+BQPD are nearly the same. The BD+VIs solver is the fastest among all the solvers that are compared in this paper.[3] More specifically, BD+VIs reduces computation times by nearly 50% compared with BB+SQP and BB+BQPD. Furthermore, the computing time of BD+VIs is 27.5% that of BD on average. Thus, our proposed VIs can efficiently reduce the computation time, especially compared with BD without adding the proposed valid inequalities.

[3] The BB+SQP and BB+BQPD algorithms are partially implemented in object code, while our self-written BD+VIs algorithm is implemented in Matlab code, which is in general slower than object code. Hence, in practice, i.e., when all algorithms are implemented in object code, the speed-up of BD+VIs w.r.t. BB+SQP and BB+BQPD will be even higher.



(a) Objective function values

(b) Maintenance scheduling result $\Delta_{g,k}$ of Case 1

Fig. 6. Simulation results for the network.

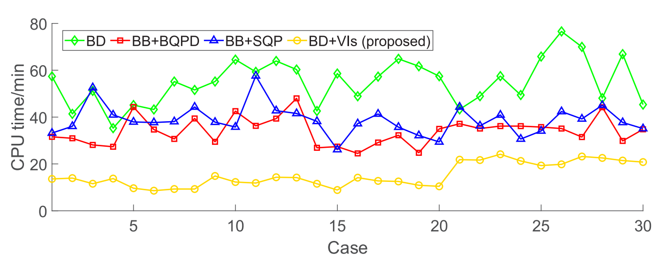


Fig. 7. Comparison of the CPU times with various solvers.

Moreover, for BD+VIs, the average CPU times of the first, second, and third 10 sets of cases are 11.6 min, 12.1 min, and 21.6 min, respectively. Regarding the first and second sets of 10 cases, the CPU times are nearly the same, while for the third set of 10 cases, the CPU times increase. This phenomenon shows that the effectiveness of the valid inequalities is determined by whether the currently predicted loads are larger than the previously predicted loads, i.e., the effectiveness of the proposed valid inequalities decreases when the "larger than" scenario occurs.

8. Conclusions and future work

This paper has proposed a truthful platform for the maintenance of generation units. The advantages of the proposed T-CBMGU over the state-of-the-art CBMGU can be summarized as follows. First, the maintenance data stored in the blockchain of T-CBMGU are immutable and trustworthy, and losing stored maintenance data is unlikely because of the distributed storage of the blockchain. Second, sharing maintenance data can be done in an efficient way because the verification when data sharing can be done without manual work. Third, because the smart contract automatically determines the maintenance decisions, actions that comprise fairness are unlikely to happen and the manual work can be reduced.

Furthermore, since the solving speeds of maintenance scheduling problems (MIQP problems) are low, an accelerated Benders decomposition algorithm is proposed by using two specially designed valid inequalities. The simulation results shows that the proposed T-CBMGU platform can obtain more social welfare by avoiding actions that compromise fairness, e.g., fairness manipulation and data tampering, compared with state-of-the-art CBMGU. Furthermore, the results demonstrate a faster solving speed of the proposed BD+VIs solver compared with three other solvers on maintenance scheduling problems. Finally, the solving speed reduction, w.r.t., the other solvers, of the proposed BD+VIs solver is large for the cases that the currently predicted loads are no larger than the previously predicted loads, while the solving speed reduction decreases when the currently predicted loads are larger than the previously predicted loads. However, the BD+VIs is still the fastest solver among the four compared in this paper.

Future work will focus on exploring additional potentials of blockchains in the maintenance of infrastructure. Moreover, the concept of truthful maintenance will be expanded further.

CRediT authorship contribution statement

Jianfeng Fu: Conceptualization, Methodology, Software, Writing – original draft. **Alfredo Núñez:** Tutorial, Review. **Bart De Schutter:** Tutorial, Review.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by CSC (China Scholarship Council) with funding number: 201806280023.

References

Abbas, Y., Martinetti, A., Moerman, J.-J., Hamberg, T., van Dongen, L.A., 2020. Do you have confidence in how your rolling stock has been maintained? a blockchain-led knowledge-sharing platform for building trust between stakeholders. Int. J. Inf. Manage. 55, 228.

Adibi, M., 2010. IEEE 300-bus system. https://site.ieee.org/pes-tccwg/links-to-test-cases/ Accessed April 4, 2010.

Aleshi, A., Seker, R., Babiceanu, R.F., 2019. Blockchain model for enhancing aircraft maintenance records security. In: 2019 IEEE International Symposium on Technologies for Homeland Security. pp. 1–7. http://dx.doi.org/10.1109/HST47167.2019.9032943.

Alizadeh, A., Fereidunian, A., Moghimi, M., Lesani, H., 2021. Reliability-centered maintenance scheduling considering failure rates uncertainty: A two-stage robust model. IEEE Trans. Power Deliv. http://dx.doi.org/10.1109/TPWRD.2021.3101458.

Alvarez-Alvarado, M.S., Jayaweera, D., 2020. Reliability-based smart-maintenance model for power system generators. IET Gener. Transm. Distrib. 14 (9), 1770–1780.

Antal, C., Cioara, T., Antal, M., Mihailescu, V., Mitrea, D., Anghel, I., Salomie, I., Raveduto, G., Bertoncini, M., Croce, V., et al., 2021. Blockchain based decentralized local energy flexibility market. Energy Rep. 7, 5269–5288.

Butt, G.Q., Sayed, T.A., Riaz, R., Rizvi, S.S., Paul, A., 2022. Secure healthcare record sharing mechanism with blockchain. Appl. Sci.-Basel 12 (5), 2307.

Chang, F., Zhou, G., Zhang, C., Ding, K., Cheng, W., Chang, F., 2021. A maintenance decision-making oriented collaborative cross-organization knowledge sharing blockchain network for complex multi-component systems. J. Clean. Prod. 282, 124541.

Chatterjee, R., Chatterjee, R., 2017. An overview of the emerging technology: Blockchain. In: 2017 3rd International Conference on Computational Intelligence and Networks. IEEE, pp. 126–127.

Esmat, A., de Vos, M., Ghiassi-Farrokhfal, Y., Palensky, P., Epema, D., 2021. A novel decentralized platform for peer-to-peer energy trading market with blockchain technology. Appl. Energy 282, 116123.

Fallahi, F., Yildirim, M., Lin, J., Wang, C., 2021. Predictive multi-microgrid generation maintenance: Formulation and impact on operations & resilience. IEEE Trans. Power Syst. http://dx.doi.org/10.1109/TPWRS.2021.3066462.

Feng, C., Wang, X., 2009. A competitive mechanism of unit maintenance scheduling in a deregulated environment. IEEE Trans. Power Syst. 25 (1), 351–359.

Fletcher, R., 2000. Stable reduced Hessian updates for indefinite quadratic programming. Math. Program. 87 (2), 251–264.

Fletcher, R., Leyffer, S., 1998. User Manual for filterSQP. Tech. rep., Department of Mathematics, University of Dundee.

Floudas, C., Aggarwal, A., Ciric, A., 1989. Global optimum search for nonconvex NLP and MINLP problems. Comput. Chem. Eng. 13 (10), 1117–1132.

Fu, J., Nunez, A., De Schutter, B., 2020. A short-term preventive maintenance scheduling method for distribution networks with distributed generators and batteries. IEEE Trans. Power Syst. 36 (3), 2516–2531.

Helseth, A., Fodstad, M., Mo, B., 2018. Optimal hydropower maintenance scheduling in liberalized markets. IEEE Trans. Power Syst. 33 (6), 6989–6998.

Hsu, J.-Y., Wang, Y.-F., Lin, K.-C., Chen, M.-Y., Hsu, J.H.-Y., 2020. Wind turbine fault diagnosis and predictive maintenance through statistical process control and machine learning. IEEE Access 8, 23427–23439. http://dx.doi.org/10.1109/ACCESS.2020.2968615.

Jiang, S., Cao, J., Wu, H., Yang, Y., 2021. Fairness-based packing of industrial IoT data in permissioned blockchains. IEEE Trans. Ind. Inf. 17 (11), 7639–7649. http://dx.doi.org/10.1109/TII.2020.3046129.

Jiang, Y., Zhou, K., Lu, X., Yang, S., 2020. Electricity trading pricing among prosumers with game theory-based model in energy blockchain environment. Appl. Energy 271, 115239.

Ling, X., Wang, J., Bouchoucha, T., Levy, B.C., Ding, Z., 2019. Blockchain radio access network (B-RAN): Towards decentralized secure radio access paradigm. IEEE Access 7, 9714–9723. http://dx.doi.org/10.1109/ACCESS.2018.2890557.

Moinian, F., Ameli, M.-T., 2020. An ISO-based security constrained generation maintenance coordination considering long-term SCOPF. Electr. Power Syst. Res. 189, 106538.

Mrissa, M., Tošić, A., Hrovatin, N., Aslam, S., Dávid, B., Hajdu, L., Krész, M., Brodnik, A., Kavšek, B., 2022. Privacy-aware and secure decentralized air quality monitoring. Appl. Sci.-Basel 12 (4), 2147.

Lycklama à Nijeholt, H., Oudejans, J., Erkin, Z., 2017. DecReg: A framework for preventing double-financing using blockchain technology. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts. pp. 29–34.

Nowak, I., 2005. Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming. Springer Science & Business Media.

Ogieva, F., Ike, A., Anyaeji, C., 2015. Egbin power station generator availability and unit performance studies. Int. J. Phys. Sci. 10 (4), 155–172.

Rodriguez, J.A., Anjos, M.F., Côté, P., Desaulniers, G., 2018. MILP formulations for generator maintenance scheduling in hydropower systems. IEEE Trans. Power Syst. 33 (6), 6171–6180.

Rokhforoz, P., Gjorgiev, B., Sansavini, G., Fink, O., 2021. Multi-agent maintenance scheduling based on the coordination between central operator and decentralized producers in an electricity market. Reliab. Eng. Syst. Saf. 210, 107495.

Shaikh, Z.A., Khan, A.A., Baitenova, L., Zambinova, G., Yegina, N., Ivolgina, N., Laghari, A.A., Barykin, S.E., 2022. Blockchain hyperledger with non-linear machine learning: A novel and secure educational accreditation registration and distributed ledger preservation architecture. Appl. Sci.-Basel 12 (5), 2534.

Sharma, P.K., Kumar, N., Park, J.H., 2019. Blockchain-based distributed framework for automotive industry in a smart city. IEEE Trans. Ind. Inf. 15 (7), 4197–4205. http://dx.doi.org/10.1109/TII.2018.2887101.

Strepparava, D., Nespoli, L., Kapassa, E., Touloupou, M., Katelaris, L., Medici, V., 2022. Deployment and analysis of a blockchain-based local energy market. Energy Rep. 8, 99–113.

Toubeau, J.-F., Pardoen, L., Hubert, L., Marenne, N., Sprooten, J., De Grève, Z., Vallée, F., 2021. Machine learning-assisted outage planning for maintenance activities in power systems with renewables. Energy 121993.

Wang, Y., Kirschen, D.S., Zhong, H., Xia, Q., Kang, C., 2016. Coordination of generation maintenance scheduling in electricity markets. IEEE Trans. Power Syst. 31 (6), 4565–4574.

Wang, Y., Zhong, H., Xia, Q., Kirschen, D.S., Kang, C., 2015. An approach for integrated generation and transmission maintenance scheduling considering N-1 contingencies. IEEE Trans. Power Syst. 31 (3), 2225–2233.

Xie, Y.-s., Chang, X.-q., Yin, X., Zheng, H., et al., 2022. Research on the transaction mode and mechanism of grid-side shared energy storage market based on blockchain. Energy Rep. 8, 224–229.

Yapa, C., de Alwis, C., Liyanage, M., Ekanayake, J., 2021. Survey on blockchain for future smart grids: Technical aspects, applications, integration challenges and future research. Energy Rep. 7, 6530–6564.

Yildirim, M., Sun, X.A., Gebraeel, N.Z., 2016. Sensor-driven condition-based generator maintenance scheduling—Part I: Maintenance problem. IEEE Trans. Power Syst. 31 (6), 4253–4262.

Yu, X., Tang, C., Palensky, P., Colombo, A., 2021. Blockchain: What does it mean to industrial electronics? technologies, challenges, and opportunities. IEEE Ind. Electron. Mag. http://dx.doi.org/10.1109/MIE.2021.3066332.