
Research article

Stochastic energy balancing in substation energy management

Hassan Shirzeh*, Fazel Naghdy, Philip Ciufu and Montserrat Ros

School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Australia

* **Correspondence:** Email: hs920@uowmail.edu.au; Tel: +6-140-581-0098.

Abstract: In the current research, a smart grid is considered as a network of distributed interacting nodes represented by renewable energy sources, storage and loads. The source nodes become active or inactive in a stochastic manner due to the intermittent nature of natural resources such as wind and solar irradiance. Prediction and stochastic modelling of electrical energy flow is a critical task in such a network in order to achieve load levelling and/or peak shaving in order to minimise the fluctuation between off-peak and peak energy demand. An effective approach is proposed to model and administer the behaviour of source nodes in this grid through a scheduling strategy control algorithm using the historical data collected from the system. The stochastic model predicts future power consumption/injection to determine the power required for storage components. The stochastic models developed based on the Box-Jenkins method predict the most efficient state of the electrical energy flow between a distribution network and nodes and minimises the peak demand and off-peak consumption of acquiring electrical energy from the main grid. The performance of the models is validated against the autoregressive moving average (ARIMA) and the Markov chain models used in previous work. The results demonstrate that the proposed method outperforms both the ARIMA and the Markov chain model in terms of forecast accuracy. Results are presented, the strengths and limitations of the approach are discussed, and possible future work is described.

Keywords: Smart grid; renewable energy sources; stochastic modelling; forecasting power flow; energy balancing.

1. Introduction

Distributed electrical power production using small size renewable energy sources is growing rapidly due to electricity price increases and environmental policies. Based on the report from the Clean Energy Council [1] in Australia, the number of photovoltaic (PV) systems connected

to the distribution grid has increased from 20,000 in 2008 to over 1 million by March 2013. This phenomenon has led to highly geographically distributed power generation with the consequence of operational uncertainty due to the stochastic and uncontrollable nature of primary power generation resources [2]. Management of electrical energy flow (hereafter simply referred to as energy flow) faces difficulties brought about by the unpredictable nature of renewable energy sources such as solar irradiance and wind speed. These unpredictable characteristics of renewable energy sources lead to fluctuations and disturbances of energy flow in the distribution power grid [3][4].

Another difficulty is that the fluctuation in power demand between peak and off-peak times can result in further disturbance to the power flow in the main grid. In order to mitigate such disturbances, the energy flow in the main grid can be scheduled and modelled stochastically. Leveraging multi-timescale dispatch and scheduling of stochastic power generation is a critical factor for the distribution network service provider (DNSP) in coordinating a large number of power demand and stochastic supplies [5][6] as well as to guarantee smooth energy consumption on the load and demand sides [7].

In this study, the smart grid is modelled as a network of a large number of independent energy nodes represented by renewable energy sources and generators ('source nodes'), batteries ('store nodes') and loads ('sink nodes'), distributed across the grid. Using a one-step-ahead forecast, the behaviour of the nodes in terms of their availability and the amount of power they inject/consume is predicted. The energy produced by the renewable energy sources depends on weather patterns that are, in turn, probabilistic and stochastic. In an ideal network, the behaviour of the nodes must be continuously predicted and analysed for the purpose of energy management.

An increase in the number of nodes leads to an increase in the complexity of managing the smooth energy consumption. In order to reduce this complexity and simplify the modelling process, nodes are clustered. The clusters have the lowest possible power flow residuals after considering the power consumption by sink nodes, estimated based on the predicted power produced by the source nodes.

In each cluster, the difference between the energy demand of peak and off-peak hours is balanced using batteries as store nodes. Each battery stores energy from a renewable energy source when the demand is low. The stored energy is consumed by loads when the demand is high and there is not enough energy produced by renewable energy sources. In order to determine the amount of energy that needs to be stored in batteries or released to the network in a cluster in one-step-ahead, linear programming [8] is applied to keep the state of charge (SOC) of the batteries at the highest level by using the maximum power flow from renewable energy resources and the minimum power flow acquired from the main grid. Linear programming deploys a set of objective functions designed based on various constraints governing the network such as charge/discharge capacity limitation on the SOC of the batteries.

In this paper, the proposed approach consisting of cluster formation, stochastic modelling, and linear programming, is referred to as the stochastic adaptive optimisation (SAO) algorithm. This algorithm establishes an orderly connection and disconnection of renewable energy resources to/from a distribution substation and also determines the amount of power that must be injected/consumed by nodes. SAO minimises the power acquired from the main grid to

smooth and balance the energy consumption and to maximise the SOC of the batteries.

The resulting algorithm is validated using real solar irradiance and wind speed data recorded at one-minute intervals from a weather station in Perth, Australia. Using linear programming methods in MATLAB, the maximum and minimum power exchanges between the battery and the main grid are calculated. The proposed algorithm is validated using a network based on the IEEE 34-bus test feeder.

The remainder of this paper is structured as follows. Sections 1.1 and 1.2 highlight the background and contribution of this paper. Section 2.1 explains the work and problem formulation on development of the adaptive optimisation algorithm. In Section 2.2, the modeling of the nodes and the stochastic models proposed in the work are described. In Section 2.3, the proposed SAO algorithm and the constraints on power flow in the power network are explained. Section 3 provides the results of the simulation and validation of the proposed algorithm. Finally, Section 4 draws some conclusions and describes future work.

1.1. Stochastic modelling of energy flow in the smart grid

The majority of stochastic models proposed in the literature for modelling renewable energy sources use transition probability matrix models and assume that the dynamics of such systems are repetitive and cyclic. For example, Nfaoui, et al. [9] derive the transition probability matrix from wind speed data for a stochastic Markov chain model to forecast wind speed. Hocaoglu [10] calculates the transition probability matrix for the presented hidden Markov modelling of solar irradiance by monitoring both solar irradiance and air temperature. Air temperature is observed as the hidden element for accurate prediction of solar irradiance.

All the proposed Markov chain models are developed based on the transitional probability matrices of the observed states. Once the transitions among all states are observed, the transition probability matrices are calculated for the period of observation. The transition probability between state i and j is calculated by dividing the number of previously observed transitions from state i to j by the number of all observed transitions from state i to all states including state j .

In addition to the transition probability matrix, there are studies that deploy the Box-Jenkins (B-J) method for stochastic modelling of renewable energy sources. Chen et al.[11] demonstrate that time series analysis methods such as limited-ARIMA, as outlined by Box-Jenkins[12], can be more efficient than that of the ten-state first-order discrete Markov model in terms of probability distribution and prediction accuracy of wind power generation. Time series analysis is used in the stochastic models to predict one-step-ahead values based on previously observed data. For example, the predicted value of power flow, $\hat{P}(t+1)$, is calculated based on the n previously observed values, $\{P(t-i) \mid i = 0, \dots, n\}$.

In the majority of the previous work, data used in the modelling is sampled at long intervals such as hourly or daily. For example, in the work introduced in [13], the hourly average wind speed time series is modelled stochastically using the Box-Jenkins method to predict wind speed 24 hours ahead. The daily timeframe, however, causes problems in power flow prediction for small sized PV. The power flow produced by PV or a small size wind turbine can show completely different behaviour using a five-minute sampling interval. For example, with a small cloud, the injected power from a PV can drop from its peak to its minimum in less than five minutes [14].

Small power sources are sensitive to any small variation in the wind speed or solar irradiance. It is possible to predict short-term variations of the power source based on the latest behaviour of the energy source, such as the previous 1020 samples, and its dynamic model. Using this approach, forecasts can be obtained rapidly with smaller memory requirements. In the study presented in this paper, two methods of time series analysis based on the moving average are developed and it is shown that they outperform ARIMA(0,1,1) and Markov chain models based on climate and consumer power consumption patterns in Australia.

1.2. Multi-agent systems and energy flow scheduling

Agent technology can be deployed to design information system modelling for energy flow management in smart grids. An agent acts like a physical entity with no physical presence in the environment [15]. When agents are distributed throughout the network, they have the potential to control the whole network autonomously with a minimum amount of data exchange and minimum computational demands. In such models, each autonomous agent takes the responsibility for making decisions and interacting with other agents. For example, contract net protocol (CNP) [16] is an energy flow scheduling model for dynamic task allocation via negotiation among multiple agents by constantly exchanging environmental information among themselves. The result over a limited number of nodes, demonstrates that negotiation about the nodes' potential helps the implementation of energy management algorithms.

In the current research, the negotiation strategy is used to manage the increasing number of nodes with cluster formation techniques. This negotiation strategy considers the cooperation between agents based on the node's parameters, such as cost or distance between nodes. For example, the sink node may be selected to form a cluster with source nodes based on their lowest energy prices. After cluster formation, an energy flow scheduling algorithm is used in each cluster to balance the difference between the energy demand of peak and off-peak hours. In this energy flow scheduling algorithm within the cluster, only the power flow residual is considered without taking into account the other parameters of the nodes such as the energy price and cost. This is in contrast to the other studies such as [17] [18] that are focused on exploring the effects of locational pricing on energy flow scheduling.

There are various studies reported in the literature concerning the scheduling of the energy flow to estimate the energy storage needs based on linear programming subject to equality or inequality constraints [19][20]. For example, a multistage interval-stochastic integer linear programming method is deployed to determine electric power system (EPS) planning under uncertainty [21]. A two-stage stochastic mixed-integer programming (SMIP) optimisation is presented in a smart grid using a wind turbine for scheduling load and source [3]. The results of these show the efficiency of linear programming in the formulation of the power flow constraints. In this work, linear programming is used because of its simplicity and the robustness of scheduling constraint formulation for nodes such as batteries and energy store components.

The majority of proposed energy flow scheduling algorithms are based on demand response (DR) and load levelling/peak shaving [22] concepts deployed in load scheduling. For example, an energy storage system (ESS) is used to suppress the short-term power fluctuation by reducing the peak-valley load differences of the active distribution network (ADN) [23]. The interconnection of distribution power systems and the increasing number of ESS also presents challenges to the

energy management of the batteries. One of the main objectives of the energy flow scheduling algorithm proposed in the current study is to avoid setting limits on the number of sources and energy consumers.

2. Materials and Method

2.1. Problem definition

The approach applied in the current research is developed within the context of a zone-substation. In this paper, a zone substation refers to an infrastructure consisting of a number of incoming high voltage (transmission line) connections and multiple outgoing medium voltage (MV) distribution lines. The MV lines are further transformed to low voltage (LV) lines for connection to customer equipment in a distribution substation. As illustrated in Figure 1, the information produced by the proposed SAO algorithm, which includes the value of power flow in each node, is transferred by the multi-agent system management (MASM) agent to the lower layers of the smart grid responsible for functions such as voltage or power quality control. The agents have their own local control and are able to report their status to the MASM agent located at the distribution substation. As a result, the low-level control of the nodes in terms of voltage and current adjustment is decoupled from energy management. This coordinated approach means that the zone substation has the autonomy to make a decision about peak demand requirements simply by supervising the distribution substations.

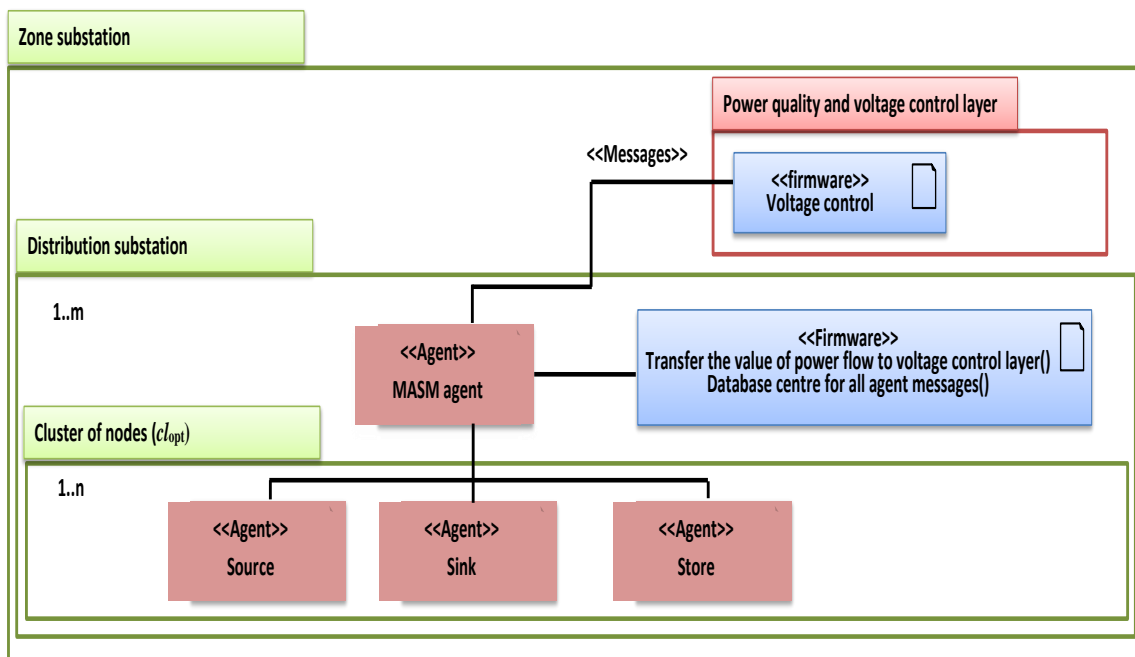


Figure 1. A typical power system modelling.

The nodes, N_i , in the distribution substation are assumed to be grouped into clusters as $cl_{opt} = \{N_i \mid i = 1, \dots, s\}$, where s is the number of nodes within cl_{opt} . A cluster is formed (i)

to manage the increased number of nodes in terms of minimising the power flow fluctuation and (ii) to reduce the number of messages exchanged between the nodes. The source and store nodes are clustered with sink nodes to minimise the one-step-ahead residual of power flow which is the difference between the predicted power flow injected and consumed within the cluster. The value of the power flow is obtained at regular sampling intervals, T_s . The one-step-ahead forecast of the power flow at time t , $P(t)$, is represented by as $\hat{P}(t+1)$. Consequently, $LO_{\hat{P}}(t+1)$, $WT_{\hat{P}}(t+1)$, $PV_{\hat{P}}(t+1)$ are the one-step-ahead forecast of power flow at time $t+1$, from load, wind turbine, and PV respectively.

Using multi-agent methodology[24][25], agents are assigned to three types of nodes present in a substation: (i) A source agent is assigned to an energy source with the ability to predict and control the injected power flow by wind turbine ($WT_{P(t)}$) or photovoltaic ($PV_{P(t)}$), (ii) a sink agent is assigned to a load (or group of loads) with the ability to predict and control the consumed power flow ($LO_{P(t)}$), and (iii) a store agent is assigned to a battery with the ability to regulate the injected/consumed power flow from batteries ($BA_{P(t)}$). In the proposed model, parameter ($PL_{P(t)}$) represents power flow between the main power line and local loads and sources in a cl_{opt} . A store agent is assigned to the power line to regulate ($PL_{P(t)}$).

Figure 2 shows the application of the proposed agent system within a typical cl_{opt} cluster. Each node in the grid is represented by a set of parameters (defined in Section 2.2) representing its ability to inject power into the grid or draw power from it. These parameters are continuously predicted and exchanged between agents, providing a global awareness for each node about the characteristics of other nodes present in the distribution substation.

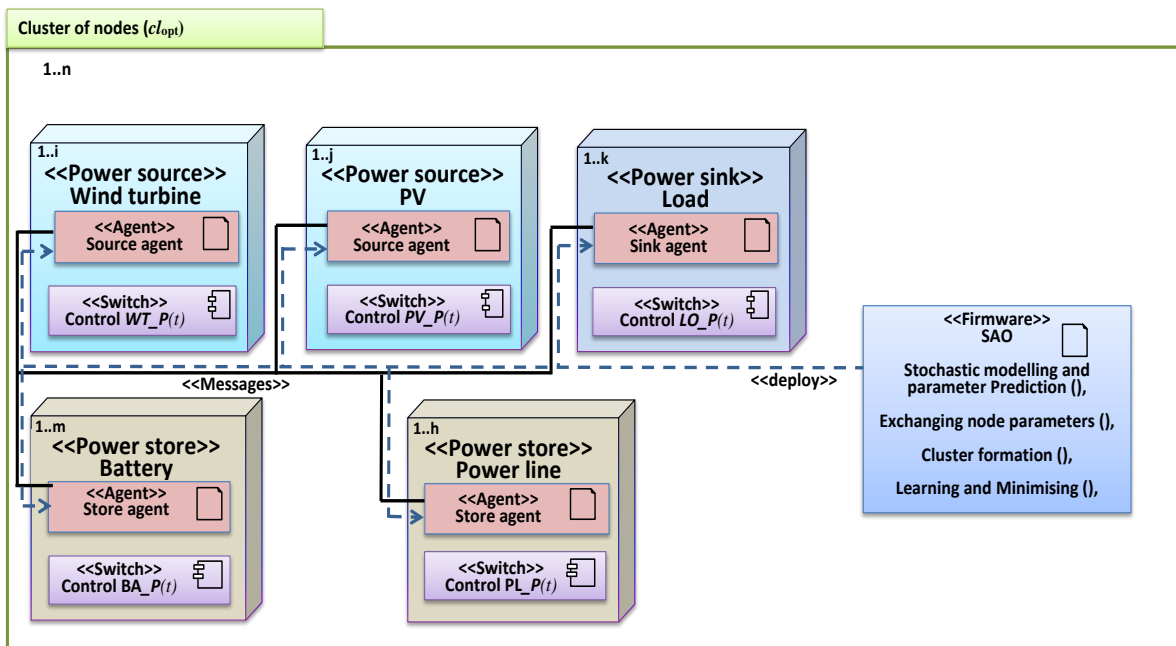


Figure 2. A typical cluster modelling with multi agent system.

Typical daily power flow, which is the sum of power flow from/to wind turbine, PV, and load within a cl_{opt} , is illustrated in Figure 3. As an alternative approach to robust design, energy balancing is deployed in the proposed algorithm to smooth the fluctuations in the energy

flow. Energy smoothing will minimise fluctuation of power flow at early stages before it affects the whole network. The algorithm proposed in this work applies a valley filling/peak shaving scenario to minimise the gap between peak (point y) and off-peak demand (point z). During valley filling, energy produced by the renewable energy sources is stored in batteries when the demand is low. For the purpose of peak shaving, the stored energy is consumed by loads when the demand is high and there is not enough energy produced by the renewable energy sources.

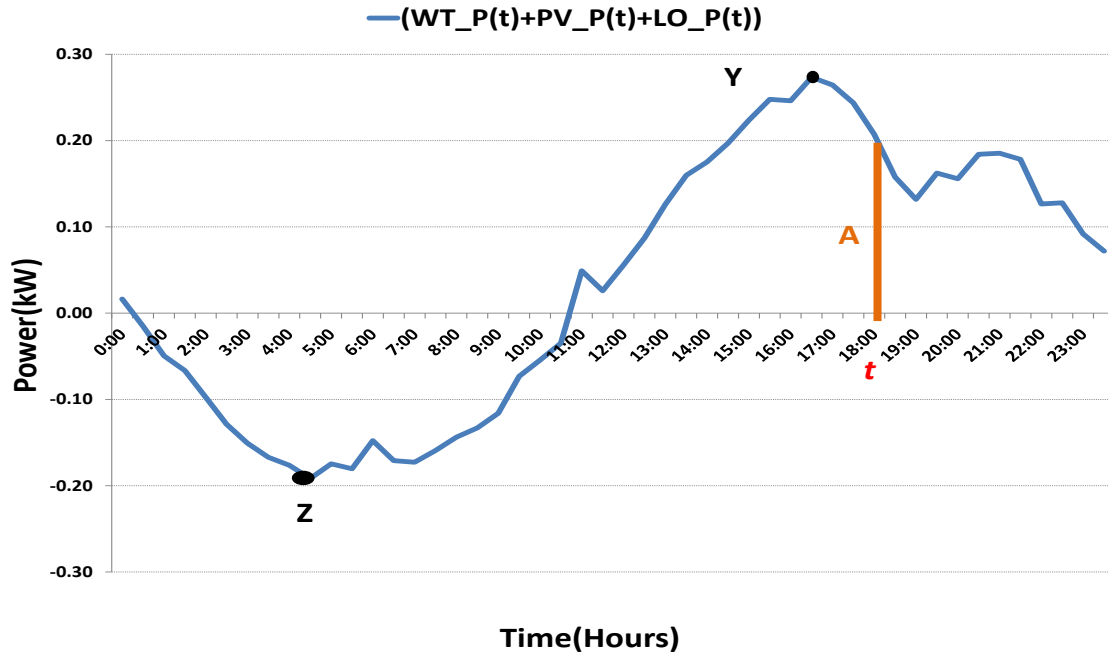


Figure 3. Valley Filling/Peak Shaving to minimise power acquisition from the main grid.

At each sampling interval, the overall objective of the SAO is achieved by minimising the difference between the sum of the injected and consumed power flow by source and sink nodes within a cl_{opt} (minimising line A in Figure 3). All the agents associated with nodes perform the SAO algorithm in three stages. In the first stage, a stochastic model (proposed in Section 2.2.2) estimates the one-step-ahead forecast of the power flow associated with each node to exchange with other agents. In the second stage, cl_{opt} is formed based on: (i) the parameters of the nodes and (ii) the predicted consumed/injected energy by the nodes. Linear programming is then applied in the third stage to the residual of power flow to estimate the energy that should be stored in batteries ($BA_{\hat{P}}(t+1)$) or released through the main network ($PL_{\hat{P}}(t+1)$). In the third stage, the concept of adaptive estimation of power flow for the valley filling/peak shaving scenario is introduced with constraint formulation and using a linear programming algorithm within a cl_{opt} .

2.2. Definition of Node's parameters

The smallest elements of the proposed system are nodes. Agents are assigned to each node to manage the power flow within the distribution substation. This means that a distribution

substation, Z , consists of m nodes that are controlled by a set of m agents, $Z_{\text{agent}} = \{AN_i \mid i = 1, \dots, m\}$.

2.2.1. Stochastic model of a node

The parameters associated with node N_i in conjunction with node N_j considered in cluster formation can be represented by a 7-tuple set (Figure 4):

$$N_i = \langle N_i-P(t), T_s, N_i-Pr, N_i-Av(t), N_{i,j}-Co, N_{i,j}-Di, N_{i,j}-Ca(t) \rangle.$$

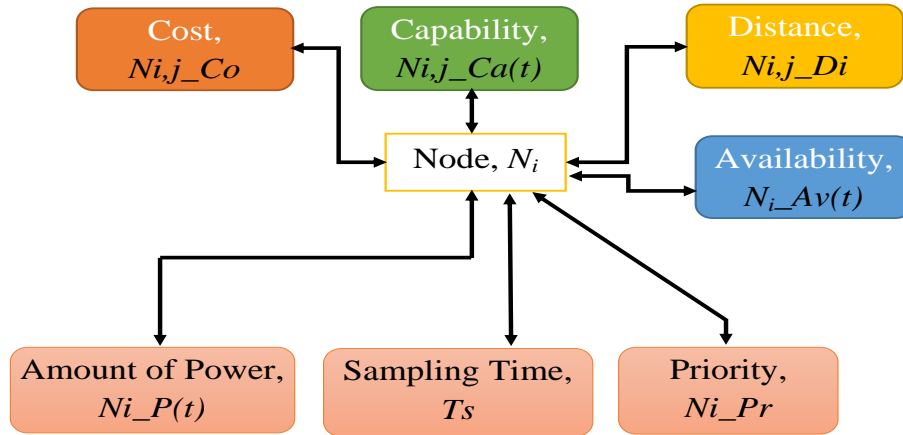


Figure 4. The parameters associated with node N_i in conjunction with N_j .

In this tuple, $N_i-P(t)$ is the amount of power that the node exchanges with other nodes. The one-step-ahead forecast of $N_i-P(t)$, shown by $N_i-\hat{P}(t+1)$, is estimated using a stochastic model (Section 2.2.2).

The time series analysis proposed by Box-Jenkins deploys a sequence of data points, measured typically at uniform time-intervals of T_s [12][26]. In power systems, the optimum value of the T_s varies according to the dynamics of the nodes deployed. For example, the required sampling interval of power flow from a small size wind turbine is lower than PV because of the fast variation in the wind speed compared to the smooth change of solar radiation.

The priority, N_i-Pr , is used by the node during the cluster formation process. N_i-Pr is a binary value that is assigned 0 (low priority) or 1 (high priority) depending on the type of node, as shown in Table 1. A store node can be a source or a sink depending on whether the power is in surplus or deficit. For example, N_i-Pr for a battery, which operates as a source by a power rating of 0.8 kW and injecting power for five minutes, can be defined with a priority of zero because the battery is given the lowest priority compared to other sources of power such as a wind turbine, during cluster formation with load nodes [27].

The parameters $N_{(i,j)-Ca(t)}$, $N_{(i,j)-Co}$, and $N_{(i,j)-Di}$ define the characteristics of a node during cluster formation. $N_{(i,j)-Ca(t)}$ represents the capability of a node at time t , calculated based on the difference between the power required by node N_j and the amount of power that N_i can provide during cluster formation. For example, during cluster formation between two load nodes and two source nodes, if the total load power is 2.1 kW and total source power is 3.4 kW, then the capability of each source node in the cluster is 1.3 kW. The one-step-ahead forecast of capability, $N_{(i,j)-\hat{C}a}(t+1)$ is estimated based on $N_i-\hat{P}(t+1)$ and $N_j-\hat{P}(t+1)$.

Table 1. The nodes priority based on the node's type.

Renewable energy sources	Batteries (any kind of energy storage)	The other components
1	0	0
N/A	1	0

$N_{(i,j)}Co$ is cost and is defined as the cost of the power exchanged between N_i and N_j . For example, electricity pricing or an electricity tariff that varies widely from one locality to another within a particular time such as between peak demand and off-peak demand hours. $N_{(i,j)}Di$ is the physical distance between N_i and N_j , representing the power loss during power exchange between them.

The parameter $N_iAv(t)$ is defined as the availability of a node at time t , providing an indication of how active or inactive the node is during this period. Variation of $N_iAv(t)$ has stochastic behaviour because of the active/inactive status of renewable energy sources. When $P(t)$ is above a pre-defined threshold, N_i is considered to be an active node ($N_iAv(t)=1$), otherwise inactive ($N_iAv(t)=0$). The threshold is determined based on the size of injected/consumed power by the node. For example, for a wind turbine with maximum 900 W output power, the threshold is 300 W due to the wind turbine's technical specifications. $N_iAv(t)$ is an indication of the node's potential for cluster formation.

The one-step-ahead forecast of availability, $N_i\hat{Av}(t+1)$, is estimated by a stochastic model built on the historical data of $N_iAv(t)$ accumulated over the last n sampled data. For estimation of $N_i\hat{Av}(t+1)$, initially the simple moving average (SMA) of historical data is calculated by (1) in order to track the availability of the node during the last n sampling points. Then, the exponential moving average (EMA) of the last n samples is calculated by (2) in order to give more weight to the most recent available nodes rather than the average of the last n samples.

The values of α and β , which are the SMA (α) and EMA (β) respectively, are compared [28] to determine the value of $N_i\hat{Av}(t+1)$. If β is larger than α , then the nodes have been more available recently compared to the average of their availability over the last n sampling intervals. Hence, $N_i\hat{Av}(t+1)=1$, otherwise (if the value of β is less than α), $N_i\hat{Av}(t+1)=0$.

$$\alpha = \frac{\sum_{k=0}^n N_iAv(t-k)}{n} \quad (1)$$

$$\beta = \frac{N_iAv(t) + \sum_{k=1}^n (1-\theta)^k N_iAv(t-k)}{1 + \sum_{k=1}^n (1-\theta)^k}, \quad \text{where: } 0 < \theta < 1 \quad (2)$$

2.2.2. Stochastic models of power flow

The purpose of stochastic modelling is to estimate the one-step-ahead forecast of the power flow, $N_i\hat{P}(t+1)$, produced by the source nodes. The models developed are based on the exponential smoothing moving average Box-Jenkins method that can provide robust and accurate short-term forecasts in highly fluctuating power flow time series [29][30]. Two different stochas-

tic models known as DeMarker and K%D, which are developed according to the Box-Jenkins approach [31][32], are deployed and optimised to produce the highest prediction accuracy.

In these models, the prediction accuracy is the gap between the one-step-ahead forecast of power flow $N_{i-\hat{P}}(t+1)$ and actual value of power flow $N_{i-P}(t+1)$. The predicted power flow is dynamically estimated in real time based on the most recent data such as wind speed and solar irradiance. The number of samples used in the forecast model is significantly smaller than the longer historical data required in other prediction techniques such as the Markov chain.

There is an infinite number of possible values for the continuous-time signal of power flow from renewable energy sources. Based on a quantising algorithm and in order to represent the unique amplitude-value for such a continuous range of the power flow, a series of states is defined according to the amount of the power flow. Each state represents a specific range of power flow between the available minimum and maximum power flow based on the technical specification of nodes. If the maximum and minimum injection or consumption of power is represented by P_{\max} and P_{\min} , respectively, then each state for a node is represented by $[m,n]$ where $P_{\min} \leq m < n \leq P_{\max}$. The amount of power flow in each sample lies in one of the states of $[m,n]$. The value of $((m+n)/2)$ represents any value within the state of $[m,n]$.

DeMarker model The DeMarker model takes into account the most recent data. Consequently, any minor fluctuation in the data affects the forecast. Forecasting using the DeMarker model is performed based on a time series, $DeMar(t)$, which is calculated based on the relationship defined in (3). In order to calculate the one-step-ahead forecast, $DeMar(t)$ is compared against two reference values of ($B1$ and $B2$) (Figure 5). The value of $DeMar(t)$ calculated by (3), has a maximum value of 1.0 if the summation of $DeMin(t)$ (denominator of (3)) in the previous steps is zero. The minimum value of $DeMar(t)$ is 0.0 if the summation of $DeMax(t)$ (numerator and denominator of (3)) in the previous step is zero. The typical variation of $DeMar(t)$ is illustrated in Figure 5.

The values of $B1$ and $B2$, varying between 1 and 0 as determined by the maximum and minimum values of $DeMar(t)$, are derived using Monte Carlo simulation [33] based on historical data of solar irradiance and wind speed from Australian weather stations. If we assume that the value of $B2$ is higher than $B1$, then this process determines an upper ($B2$) and lower ($B1$) bound so that the prediction error is minimised. The difference between the actual energy flow ($N_{i-P}(t)$) and the related one-step-ahead forecast estimated in previous step ($t-1$), ($N_{i-\hat{P}}(t)$), is called the prediction error of energy flow. A lower prediction error increases the prediction accuracy.

$$DeMar(t) = \frac{\sum_{k=1}^n DeMax(t-k)}{\frac{\sum_{k=1}^n DeMax(t-k)}{n} + \frac{\sum_{k=1}^n DeMin(t-k)}{n}} \quad (3)$$

where:

1. α is an integer value greater than zero.
2. $\min(P, \alpha)$ is the lowest value of $P(t)$ in previous α samples.
3. $\max(P, \alpha)$ is the highest value of $P(t)$ in previous α samples.
4. If $P(t) = P(t-1)$ then $DeMar(t) = 0.5$ otherwise :

i) If $\max(P, \alpha) > \max(P, \alpha + 1)$ then $DeMax(t) = \max(P, \alpha) - \max(P, \alpha + 1)$, otherwise $DeMax(t) = 0$.

ii) If $\min(P, \alpha) < \min(P, \alpha + 1)$ then $DeMin(t) = \min(P, \alpha + 1) - \min(P, \alpha)$, otherwise $DeMin(t) = 0$.

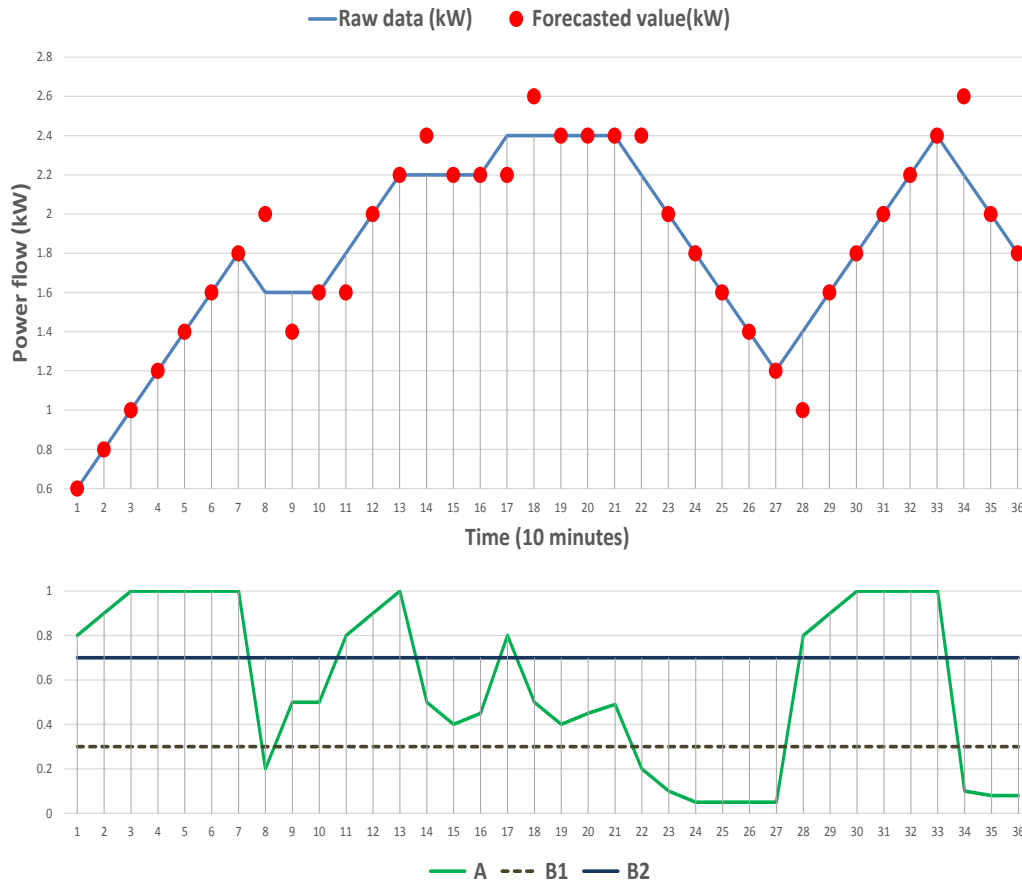


Figure 5. Example of reference lines and oscillating line in DeMarker model.

Referring to Figure 5, the forecast is estimated by comparing $DeMar(t)$ (oscillating line *A*) against two reference values of (*B1*, *B2*) according to the following algorithm:

1. If line *A* falls below line *B1* then a downturn movement and a lower state is expected in the one-step-ahead forecast.
2. If line *A* rises above line *B2* then an upward movement and a higher state is expected in the one-step-ahead forecast.
3. Else, the state for the one-step-ahead forecast will be set the same as the current state.

In the example illustrated in Figure 5, the power flow is 1.6 kW at sample number 8. The one-step-ahead forecast is 1.4 kW (one step lower at sample number 9) because the value of indicator $DeMar(t)$ has been fallen below line *B1* in the previous sample. In another example, the power flow is 1.4 kW at sample 28. The one-step-ahead forecast at sample 29 is 1.6 kW because the value of indicator $DeMar(t)$ has risen above line *B2* in the previous sample.

K%D model The K%D model gives more weight to older data and therefore it is less influenced by transients in the data. Forecasting using the K%D model is performed based on a time series, $KI(t)$ (oscillating line $C1$), which is calculated based on the relationship defined in (4). $KI(t)$ is calculated based on previously observed data. The value of the time series, $KDI(t)$ at time t (oscillating line $C2$), is determined based on the moving average of $KI(t)$ (5) over n previous samples (Figure 6). $KDI(t)$ and $KI(t)$ are compared against two reference values of ($D1$ and $D2$) to forecast the next step-ahead forecast. The value of $KI(t)$, oscillating line $C1$, calculated in (4), has a maximum value of 100.0 if $\text{MIN}(P, \alpha)$ is zero and the minimum value of $KI(t)$ is 0.0 if $P(t)$ is equal to $\text{MIN}(P, \alpha)$.

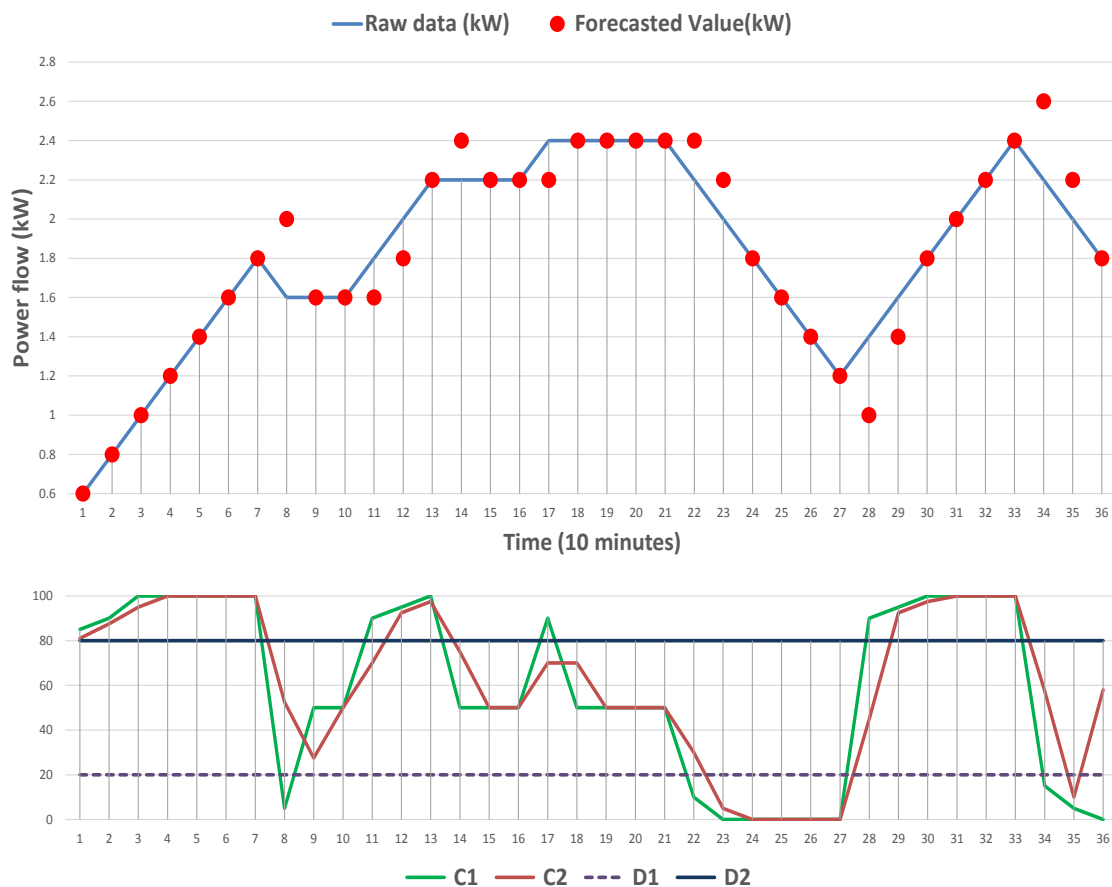


Figure 6. Example of reference lines and oscillating line in K%D model.

The values of $D1$ and $D2$ are derived using Monte Carlo simulation based on historical data of the solar irradiance and the wind speed from Australian weather stations. The reference values vary between 0 and 100, the maximum and minimum values of $KI(t)$. If it is assumed that $D2$ is higher than $D1$, then this process determines an upper ($D2$) and lower ($D1$) bound so that the prediction error is minimised.

$$KI(t) = \frac{(P(t) - \min(P, \alpha))}{(\max(P, \alpha) - \min(P, \alpha))} * 100 \quad (4)$$

$$KDI(t) = \frac{\sum_{k=1}^n KI(t-k)}{n} \quad (5)$$

where:

1. α is an integer value greater than zero.
2. If $P(t) = P(t-1)$ then $KI(t) = 50$ otherwise:
 - i) $\min(P, \alpha)$ is the lowest value of $P(t)$ in previous α sample.
 - ii) $\max(P, \alpha)$ is the highest value of $P(t)$ in previous α sample.

The forecast is estimated by comparing line $C1$ and line $C2$ against two reference values of $D1$ and $D2$ as illustrated in Figure 6 according to the following algorithm:

1. If both $C1$ and $C2$ are above line $D2$, and $C1$ rises above $C2$ then an upward movement and a higher state is predicted in the one-step-ahead.
2. If both $C1$ and $C2$ are below $D1$, and $C1$ falls below $C2$ then a downward movement and a lower state is expected in the step-ahead.
3. Else, the state for the step-ahead will be set as the same as the current state.

In the example illustrated in Figure 6, the power flow is 1.6 kW at sample number 8. The one-step-ahead forecast remains at 1.6 kW, because indicator $C2$ is not below $D1$ in the previous sample. In another example, the power flow is 1.4 kW at sample number 28. The predicted one-step-ahead forecast for sample number 29 is 1.4 kW because the values of $C1$ and $C2$ are not above line $D2$ in the previous step.

The values of $D1$ and $D2$ in the K%D model and the value of $B1$ and $B2$ in the DeMarker model remain valid until the end of the forecast as they are determined based on the weather and the environmental conditions.

2.3. Stochastic adaptive optimisation algorithm

The SAO algorithm is triggered when a node, N_i , in the distribution substation is activated when there is a change greater than threshold k , in the produced/consumed power within the node from $(t-1)$ to (t) , i.e, the absolute value of $|P(t) - P(t-1)| > k$, where k is determined based on technical specification of electric components when the node is activated. When $P(t)$ is positive, the node is consuming power whereas when $P(t)$ is negative, the node is an energy source. On activation, the SAO algorithm is applied to N_i through the following steps and as illustrated by the flow chart shown in Figure 7:

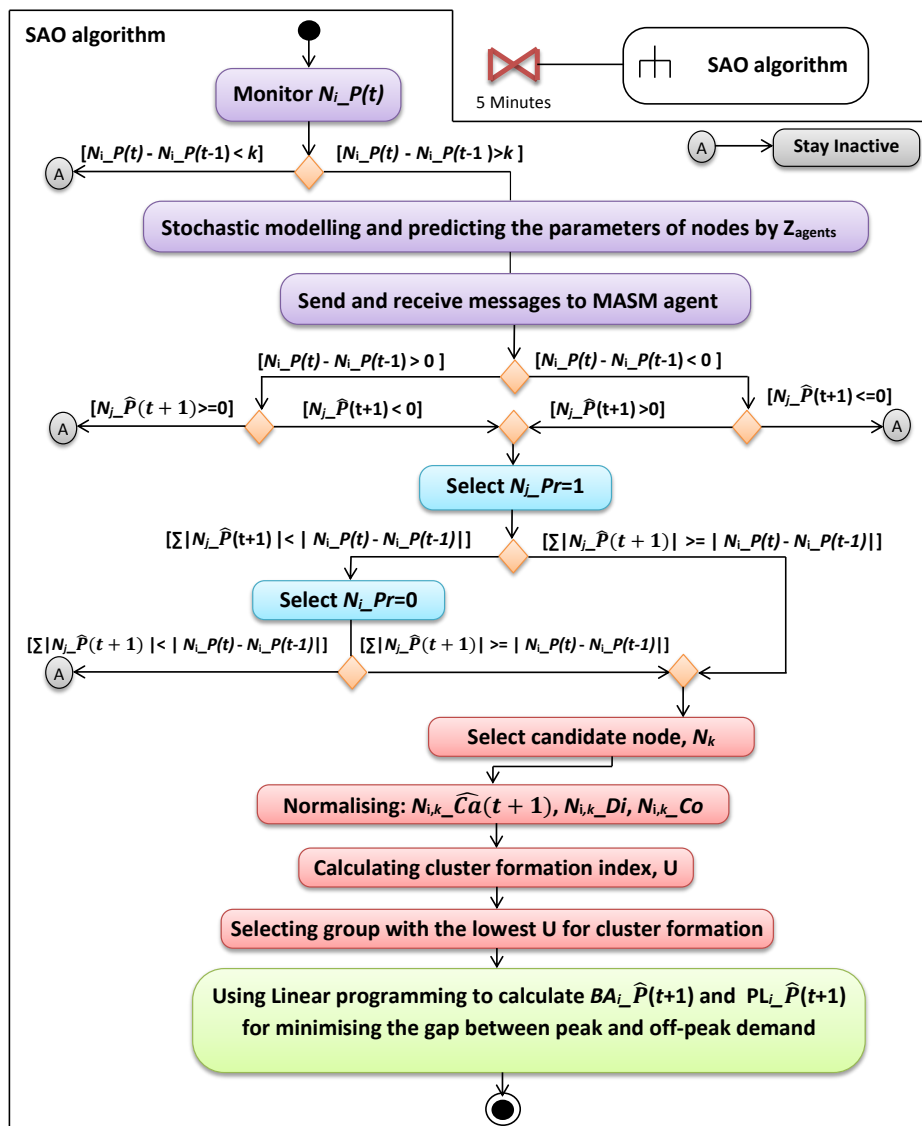


Figure 7. Proposed SOA Algorithm.

2.3.1. Cluster formation

a) In the first step, agent AN_i , broadcasts a message to all other agents in Z_{agent} , asking for their node parameters at $(t+1)$.

b) On receipt of the request, each agent AN_j (where $j = 1, \dots, m$ but $j \neq i$) in Z_{agent} estimates its one-step-ahead forecast of its availability, $N_j-\hat{Av}(t+1)$, and power flow, $N_j-\hat{P}(t+1)$. The forecast of the power flow is calculated based on two different stochastic models explained in Section 2.2.2.

c) Upon completion of the forecast, each agent AN_j provides its estimated parameters to node AN_i .

d) AN_i analyses the received data and selects a set of nodes with potential to form a cluster with them. A node N_j is selected as a candidate node if:

1. $N_j \hat{A}v(t+1)=1$.
2. N_j has a function opposite to N_i . This means that if N_i is source, then N_j should be a load and vice versa.
3. $|N_j \hat{P}(t+1)| > 0$ and $N_j Pr = 1$. In the absence of a high priority node, a node with $N_j Pr = 0$ is selected.

If at least one potential node cannot be found, then N_i switches to an inactive state and the step e) is skipped.

e) AN_i evaluates different permutations of candidate nodes to determine the best cluster formation, cl_{opt} . The parameters of the nodes which have different units and scales are normalised using a sigmoid curve function before they participate in cluster formation [27]. For example, cost can be based on expenses of power injection (e.g. dollars) or the amount of power loss among nodes (e.g. Watts). The cluster formation is performed by calculating an index $U(t)$ defined in (6), which is the sum of normalised capability, cost and distance of N_i in connection with the candidate nodes over each permutation. The permutation with highest $U_i(t)$, is selected as the best cluster. If pm is the number of candidate nodes in a permutation, then:

$$\hat{U}_i(t+1) = \sum_{n=1}^{pm} NN_{i,n}Co + \sum_{n=1}^{pm} NN_{i,n}\hat{C}a(t+1) + \sum_{n=1}^{pm} NN_{i,n}Di \quad (6)$$

The normalised values of $NN_{i,n}Co$, $NN_{i,n}\hat{C}a(t+1)$, $NN_{i,n}Di$ for every node in the permutation are calculated as follows:

$$NN_{i,n}\hat{C}a(t+1) = \frac{1}{1 + e^{(N_{i,n}\hat{C}a(t+1)-l)}} \quad (7)$$

$$NN_{i,n}Co = \frac{1}{1 + e^{(N_{i,n}Co-g)}}, \quad NN_{i,n}Di = \frac{1}{1 + e^{(N_{i,n}Di-h)}}$$

In (7) the parameters l , g and h are the average values of capability, cost and distance between N_i and the nodes considered in the permutation [27].

Finally, adaptive estimation of power flow using linear programming minimises the differences between power injection and consumption in cl_{opt} . Forming the cluster in the previous step has the advantage of reducing the size and complexity of the relationship used in the linear programming algorithm, resulting in a lower computation time. The main aim of this final step is to estimate the required power injection/consumption for storage nodes (such as batteries) and main power lines at $t+1$ to achieve valley filling/peak shaving scenarios.

2.3.2. Adaptive estimation of power flow

At each sampling interval, the overall objective of the SAO for load levelling and/or peak shaving is achieved by minimising the difference between the sum of the injected and the consumed power flow within a cl_{opt} at time $t+1$. Two scenarios are considered: (i) the nodes within cl_{opt} are either in a remote area with no connection to the main grid without store nodes or (ii) are connected to the distribution substation deploying store nodes. In the first scenario, if cl_{opt} is isolated from the main grid, the sum of the injected and consumed power flow by all the

source and sink nodes is zero. Hence, if the one-step-ahead forecast of the renewable energy is more than the one-step-ahead forecast of overall energy usage, power curtailment is activated. On the contrary, if the stochastic forecast of the energy generated by the renewable sources is less than the energy consumption, then sink nodes face outage due to insufficient power supply. This is illustrated by (8) in which $WT_i-\hat{P}(t+1)$, $PV_i-\hat{P}(t+1)$, and $LO_i-\hat{P}(t+1)$ represent the one-step-ahead forecast of power flow at time $t+1$, in the wind turbine and the PV, and the overall load in a cl_{opt} , respectively.

$$WT_i-\hat{P}(t+1) + PV_i-\hat{P}(t+1) + LO_i-\hat{P}(t+1) = 0 \quad (8)$$

Obviously, in the first scenario, sink and source nodes face problems in terms of outage and wastage of energy but in the second scenario, cl_{opt} will consume energy drawn from store nodes and the main grid. Store nodes and the main grid can be a source or a sink depending on whether the power is in surplus or in deficit. Hence, equation (8) is adjusted at each sampling interval due to the imbalance between injected/consumed energy flow from sink and source nodes.

In the second scenario, if the one-step-ahead forecast of the injected energy of the source node is more than the one-step-ahead forecast of overall energy usage, the optimisation algorithm estimates the amount of energy that should be either stored in battery ($BA_i-\hat{P}(t+1)$) or injected directly into the main grid ($PL_i-\hat{P}(t+1)$). On the other hand, if the stochastic forecast of the energy generated by the renewable sources is less than consumption, then the calculation of ($BA_i-\hat{P}(t+1)$) and ($PL_i-\hat{P}(t+1)$) represents the energy that is drawn directly from the main grid or batteries. The main challenge in this scenario is to estimate the amount of the injected/consumed energy in the main grid or the battery at time $t+1$. Linear programming is used to optimise ($BA_i-\hat{P}(t+1)$) and ($PL_i-\hat{P}(t+1)$). The objective function is formed by adding two parameters of ($BA_i-\hat{P}(t+1)$) and ($PL_i-\hat{P}(t+1)$) to (8) as shown in (9).

$$WT_i-\hat{P}(t+1) + PV_i-\hat{P}(t+1) + LO_i-\hat{P}(t+1) + BA_i-\hat{P}(t+1) + PL_i-\hat{P}(t+1) \quad (9)$$

There are limitations on the amount of injected/consumed energy in the optimisation of $BA_i-\hat{P}(t+1)$ and $PL_i-\hat{P}(t+1)$ which are imposed by factors such as the limited capacity of batteries. The constraints defined in this section govern the optimisation carried out by linear programming. The constraints, which are formulated based on s nodes of cl_{opt} , define the thresholds based on the amount of energy that is either stored in the battery or injected directly into the main grid.

Constraint 1 $BA_i-\hat{P}(t+1)$ is the amount of energy required to charge batteries from the current state of charge, $SOC(t)$, to its optimised state of charge at $t+1$, $S\hat{O}C(t+1)$. The required change in the battery state of charge can be described by (10).

$$BA_i-\hat{P}(t+1) = S\hat{O}C(t+1) - SOC(t) \quad (10)$$

C_{max} is the maximum energy that the battery can store and varies according to the capacity of the deployed battery. The degree of charging/discharging a battery in $S\hat{O}C(t+1)$, is governed by (11), showing the maximum energy that can be stored/produced by the deployed battery.

When $\hat{SOC}(t+1)$ is positive, the battery is consuming power whereas when $\hat{SOC}(t+1)$ is negative, the battery is an energy source.

$$-C_{\max} < \hat{SOC}(t+1) < C_{\max} \quad (11)$$

The charging/discharging power of a battery is subjected to a state of charge transition. In (12), the change of SOC during time period of $[t, t+1]$ is calculated, where, η_c and η_d are charging/discharging efficiencies, respectively. P_b is the real time charging/discharging power of the battery [34].

$$\hat{SOC}(t+1) = \begin{cases} SOC(t) - \int_t^{t+1} \eta_c P_b \cdot dx & \text{Charging} \\ SOC(t) - \int_t^{t+1} (1/\eta_d) P_b \cdot dx & \text{Discharging} \end{cases} \quad (12)$$

Constraint 2 The constraint for $PL_i \hat{P}(t+1)$, as shown by (9), is dependent on the standard of Distribution Network Service Providers connection requirement for typical power injection/consumption in a dwelling. Based on the standards of the major distributor of electricity in New South Wales, Australia [35], all connections or augmentations of load other than single urban residential premises must be less than 100 Amperes single phase. In addition, renewable energy generation systems that are connected to the main grid via an inverter, must comply with the Australian Standard 4777 and have a capacity of no more than 5 kW for a single-phase connections. By considering a typical dwelling with a PV of 2.5 kW and a wind turbine of 1 kW, (13) shows the corresponding constraints according to the Australian standards.

$$-3.5 \text{ kW} < PL_i \hat{P}(t+1) < 5 \text{ kW} \quad (13)$$

Constraint 3 The energy that flows into the deployed battery or the main grid, is produced by the sink and source nodes, respectively. Hence, the sum of the $BA_i \hat{P}(t+1)$ and $PL_i \hat{P}(t+1)$ is less than or equal to the sum of maximum power consumption by load (max_l), and the maximum power produced by the wind turbine (max_w) and the solar panel (max_p).

$$BA_i \hat{P}(t+1) + PL_i \hat{P}(t+1) \leq max_l + max_w + max_p \quad (14)$$

Constraint 4 At each sampling interval, one-step-ahead power flow from the sink and the source nodes is predicted. Generally, there will be an error between the one-step-ahead forecast and the actual value at time t . The difference between the energy flow ($N_i P(t)$) and the related one-step-ahead forecast estimated in the previous step ($t-1$), ($N_i \hat{P}(t)$), is considered as a perturbation and is called the one-step-ahead forecast error of energy flow, $N_i PE(t)$.

$$N_i PE(t) = N_i \hat{P}(t) - N_i P(t) \quad (15)$$

$N_i PE(t)$ is the unexpected forecast error in energy flow which is regulated by the main grid or store nodes. Hence, this error affects the limitation on battery state of charge and main grid in terms of consumption/injection flow. As shown in (16), optimisation of $BA_i \hat{P}(t+1)$, $PL_i \hat{P}(t+1)$ is adjusted by the value of $N_i PE(t)$.

$$\begin{aligned} \text{BA}_i\hat{P}(t+1) + \text{PL}_i\hat{P}(t+1) = \text{WT}_i\hat{P}(t+1) + \\ \text{PV}_i\hat{P}(t+1) + \text{LO}_i\hat{P}(t+1) - N_iPE(t) \end{aligned} \quad (16)$$

After reviewing all the constraints for the estimation of $\text{BA}_i\hat{P}(t+1)$, $\text{PL}_i\hat{P}(t+1)$, the deployed optimisation algorithm minimises (9), subject to the constraints of (10), (11), (13), (14), and (16).

3. Results and Discussion

The agent model representing the proposed algorithm is validated through computer simulation using a JADE [36] platform. The stochastic model considers two behaviour patterns of the power flow: (i) source nodes with regular and daily oscillation patterns such as the solar irradiance and (ii) source nodes that do not have a regular oscillation patterns such as the wind speed. The climatic conditions are recorded at one minute intervals. The simulation is carried out at different sampling intervals to explore its impact on prediction accuracy. At higher sampling intervals, the data is smoothed using moving average.

3.1. Computer simulation setup

In the simulation, it is assumed that seven dwellings with different levels of occupancy are connected to the distribution substation. In each dwelling there is a PV with a 2.5 kW maximum output, a wind turbine with the maximum output of 900 W, a battery with maximum capacity of 1500 Wh and a load with an average consumption of [0-1.5 kW] per hour. The maximum and minimum $SOC(t)$ are considered as 1400 Wh and 200 Wh, respectively, considering the highest efficiency in charging and discharging of the battery. The input data is obtained from seven typical households [35] and sampling the environmental conditions in Perth, WA, Australia at one-minute intervals [37].

The environmental data used in the simulation and their sources are summarised in Table 2. The parameters of n and θ in the calculation of $N_i\hat{Av}(t+1)$ are determined to produce the highest prediction accuracy based on Monte Carlo simulation with searching for their optimal values between 1 to 1440 in steps of 1 and 0.1 to 1 in steps of 0.1, respectively. The maximum value of 1440 is selected for n due to the daily pattern of environmental conditions, however the result shows that a value less than 60 achieves the highest prediction accuracy when the sampling interval is one minute. Based on the database deployed in this study, the value of n is found to be 14 and of θ , 0.2.

Table 2. The parameters of environmental characteristics.

Parameter	The methods of obtaining
Solar irradiance	Perth, WA, Australia (https://www.agric.wa.gov.au/weather-stations)
Wind speed	Perth, WA, Australia (https://www.agric.wa.gov.au/weather-stations)
Load consumption	Endeavour Energy Pty Ltd. (http://www.endeavourenergy.com.au/)
$N_{i,j}-Di$	The interval between buses in IEEE 34bus test feeder
$N_{i,j}-Co$	Electricity tariff which considered same for all agents
$N_{i,j}-Ca_t$	The current active power inside node minus requested power

The power components of a dwelling including PV, wind turbine, load, and battery are simulated in MATLAB/Simulink (Figure 8). As shown in Figure 8, the values of the variables of the network are sent to the JADE platform by an S-function, MACSimJX [38] which acts as a gateway to pass data between MATLAB/Simulink and JADE.

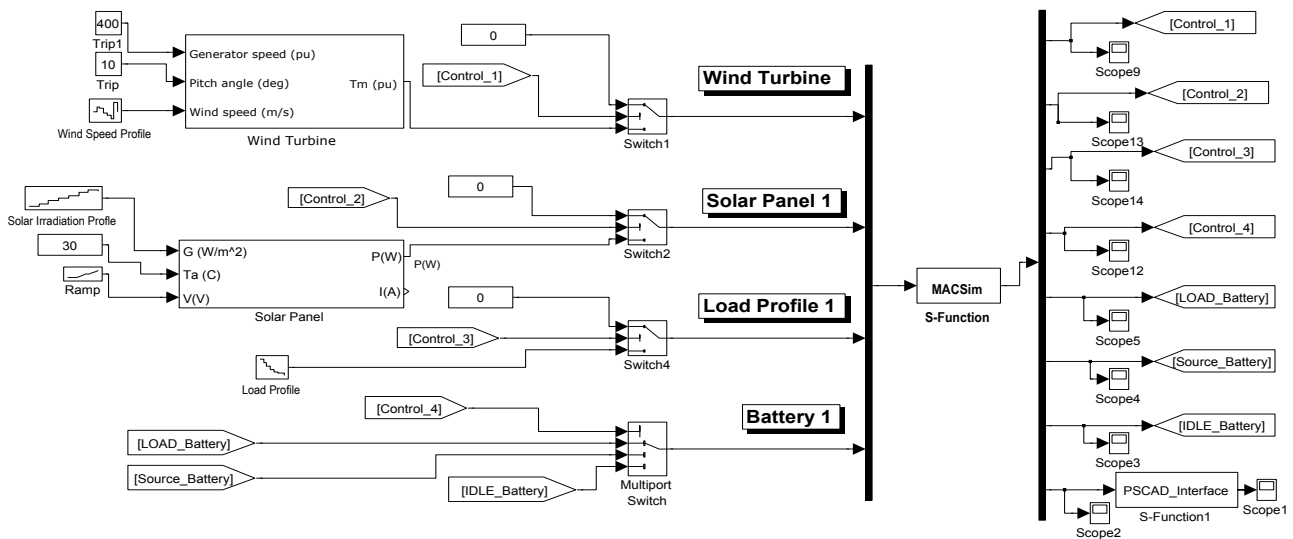
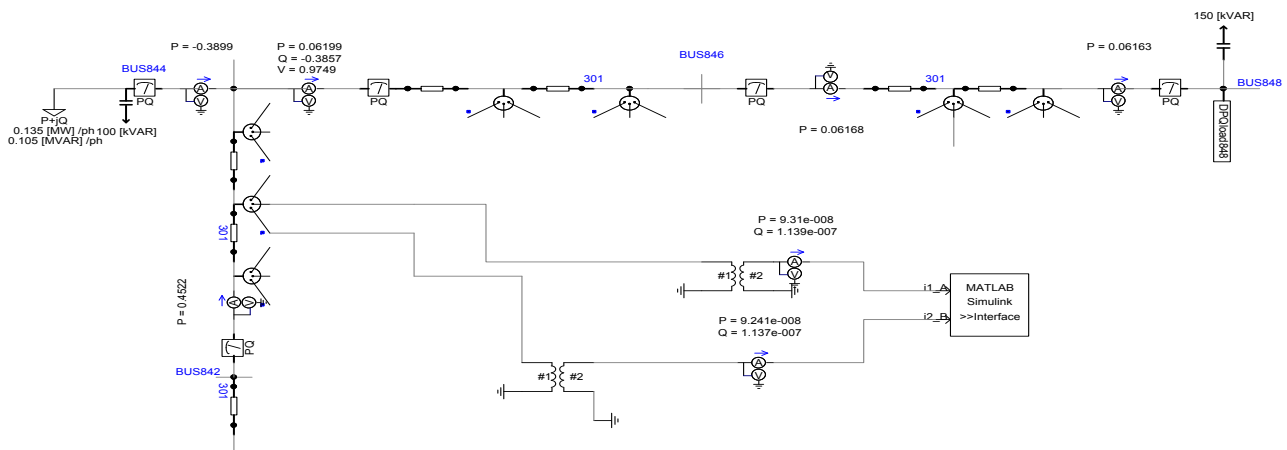


Figure 8. Modelling of one dwelling and wind turbine in MATLAB/Simulink.

The IEEE 34-bus test feeder is chosen for simulating the distribution power network because of its simplicity in monitoring the network parameters. The IEEE 34-bus test feeder is simulated in a PSCAD that is linked to MATLAB/Simulink. The dwelling substation is connected to bus 842 of the IEEE 34-bus test feeder (Figure 9). The values of the power flow are imported from MATLAB/Simulink after calculation by agents in the JADE platform. After calculation and exchange of messages in JADE, the data is sent back to the MATLAB/Simulink (Figure 8) and then to the PSCAD platform.

Table 3. Assigning the state number for every value of wind speed.

Wind Speed (m/s)	States	Output power (W)
0-4	1	0
4-5.5	2	0-150
5.5-7	3	150-300
7-8.5	4	300-450
8.5-10	5	450-600
10-11.5	6	600-750
more than 11.5	7	750-900

**Figure 9.** IEEE 34-bus Test Feeder.

The internal connection between JADE and the MATLAB/Simulink is achieved by MAC-SimJX, an open source software tool. Using JADE, MACSimJX provides provisions to receive data from Simulink and pass it on to agents for further processing. The reverse functionality is also possible [39].

3.2. Assigning the state for every value of climate condition

Regarding the definition of states in Section 2.2.2, the following example demonstrates how a series of states is defined for a typical wind turbine according to the amount of power injected. Based on the technical specification of a selected wind turbine (900 W maximum output), seven exclusive states for power injection are considered to capture the dynamic behaviour of the wind turbine caused by different wind speeds. Table 3 shows the assigned states for different wind speeds. Based on a 900 W maximum output for the wind turbine, every state is considered to provide an extra 150 W.

Figure 10 shows the comparison between the actual states, the predicted states by K%D stochastic modelling and the naive predictor [40] method over the period of eight hours. The naive predictor technique forecasts the next step-ahead as equal to the current state. The sampling interval is ten minutes using the moving average of the previous ten samples. The

wind speed has high fluctuation and an irregular pattern. The number of subsets for the K%D model are $\alpha = 5$ and $n = 21$. The values of α and n have been chosen based on Monte Carlo simulation to produce the highest prediction accuracy. In this scenario, the Monte Carlo method runs 50 estimates by random selection within the specified range of 1 to 100. This indicates that during certain hours, there is less than a one-state gap between the actual data and the one-step-ahead forecast. The result could be different, however, during other periods. This gap depends on the number of selected states, the sampling interval and the number of subsets in the K%D stochastic model.

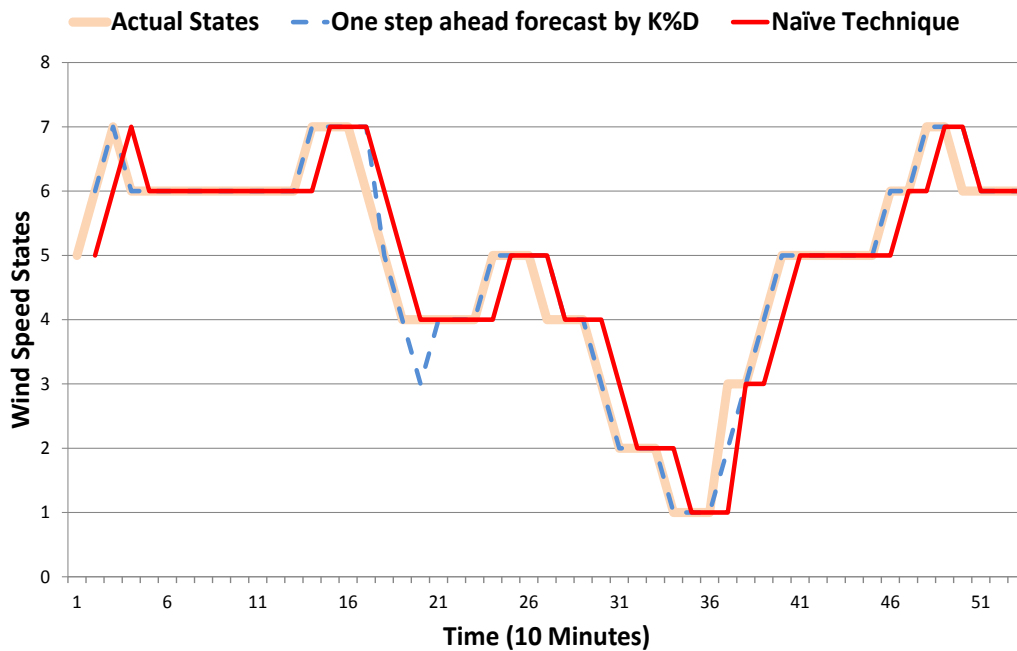


Figure 10. The comparison of actual wind speed and predicted values.

In the next step, a comparison is made between the performance of stochastic models at different sampling intervals based on their standard deviations for one-step-ahead forecast error with ARIMA (0,1,1) and a Markov chain. Using the sum of difference in actual value and the forecast, the standard deviation of the one-step-ahead forecast error is calculated for 24 hours. The climate data for the previous six months is used to calculate the transition matrix of the Markov chain.

3.3. Stochastic model performance for irregular patterns

Wind speed is selected as an example of the dataset that is highly irregular. There are seven states due to the size of the wind turbine (900 W), covering all variations in the wind speed. Figures 11 and 12 show the comparison of the K%D and the DeMarker models with the Markov chain, naive predictor and ARIMA (0,1,1) when the sampling interval is 1 or 10 minutes. The final value selection for the length of the time period for the DeMarker model is $B1 = 30$, $B2 = 70$, $\alpha = 5$ and $n = 18$ and for K%D model, the final values are $D1 = 20$, $D2 = 80$, $\alpha = 6$ and n

= 28. When the sampling interval is one minute (Figure 11), the K%D model shows less error in standard deviation compared to other methods by an average of 2.46 kWh error in average daily injection of 7.56 kWh over one month. When the sampling interval increases to 10 minutes (Figure 12), the performance of K%D and ARIMA(0,1,1) are the same.

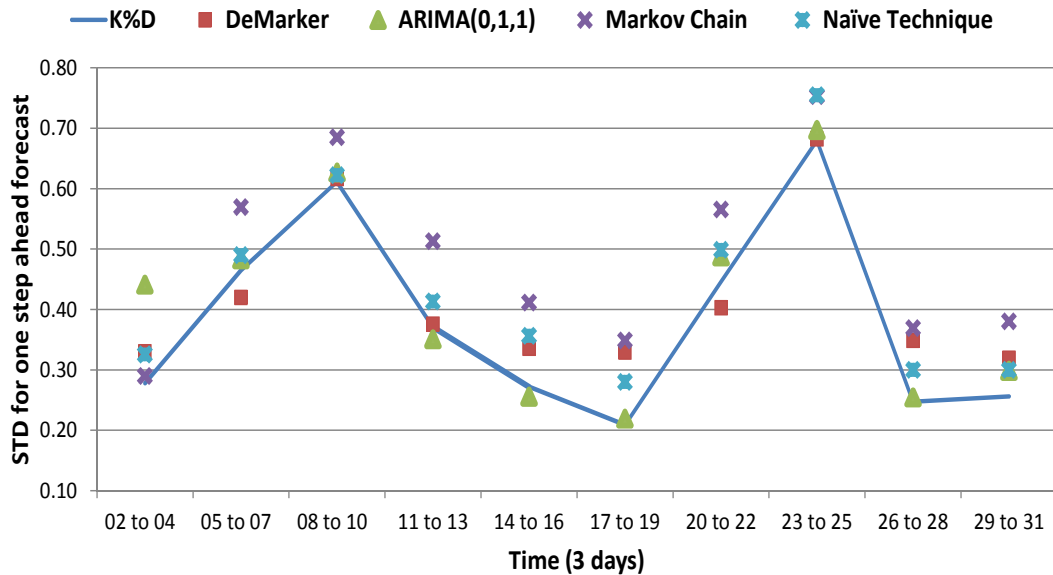


Figure 11. Comparison of stochastic model with the second order Markov chain and ARIMA(0,1,1) for wind speed when the sampling interval is 1 minute.

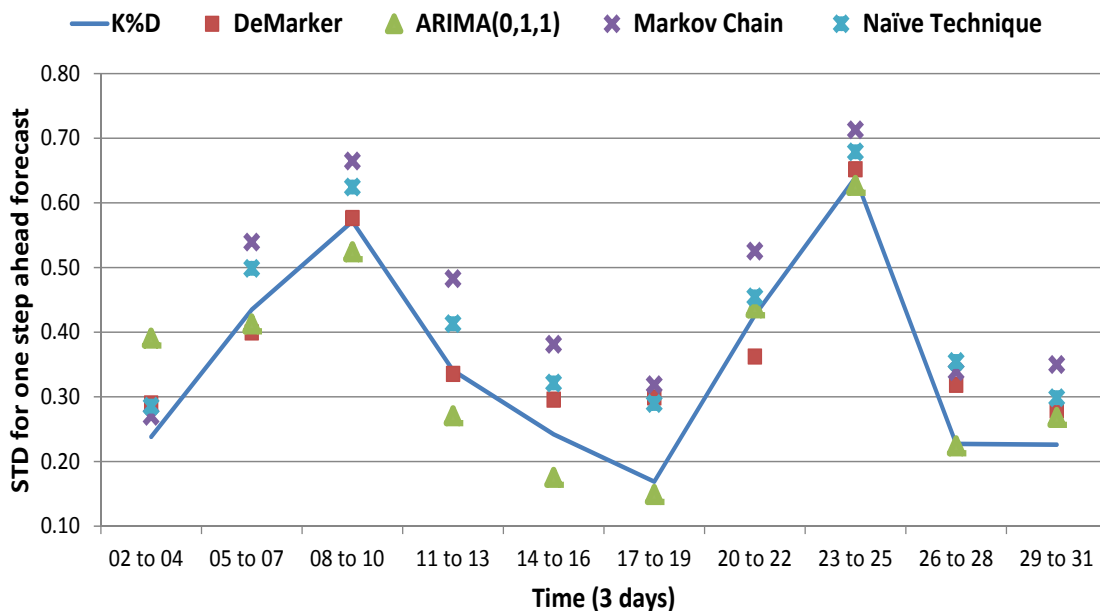


Figure 12. Comparison of stochastic model with the second order Markov chain and ARIMA(0,1,1) for wind speed when the sampling interval is 10 minute.

All the results in this step demonstrate that giving more weight to older data is the best option for stochastic modelling in terms of prediction accuracy of the wind speed. In Figure 12, the average of standard deviations are 0.35, 0.38, 0.35, 0.46 and 0.42 for K%D, DeMarker, ARIMA(0,1,1), Markov Chain and naive predictor, respectively. When the sampling interval increases to 10 minutes, it is not possible to select one stochastic model to achieve the best performance over every 3-day period. If, however, the result of one month is compared between stochastic models, then the K%D method still demonstrates the best results in terms of prediction accuracy. Time series stochastic modelling is suitable for short-term prediction of irregular patterns. For this group, the Markov chain method is less accurate when the sampling interval is less than an hour because of the lack of accuracy in the transition probability matrix of the Markov chain.

3.4. Stochastic model performance for regular patterns

Solar irradiance is selected from the group with regular patterns. There are 14 states to cover all the variations in solar irradiance. The final values of $B1 = 30$, $B2 = 70$, $\alpha = 4$ and $n = 14$ are selected for DeMarker and $D1 = 20$, $D2 = 80$, $\alpha = 6$ and $n = 25$ for K%D. Figures 13 and 14 show the comparison of the standard deviations for the one-step-ahead forecast produced by the models over one month. When the sampling interval is 1 minute (Figure 13), the DeMarker shows the best performance with the lowest standard deviation of one-step-ahead forecast with an average of 5.52 kWh error in average daily injection of 22.85 kWh over one month. When the sampling interval increases to 10 minutes (Figure 14), the DeMarker and ARIMA(0,1,1) show better performance in terms of lower standard deviations.

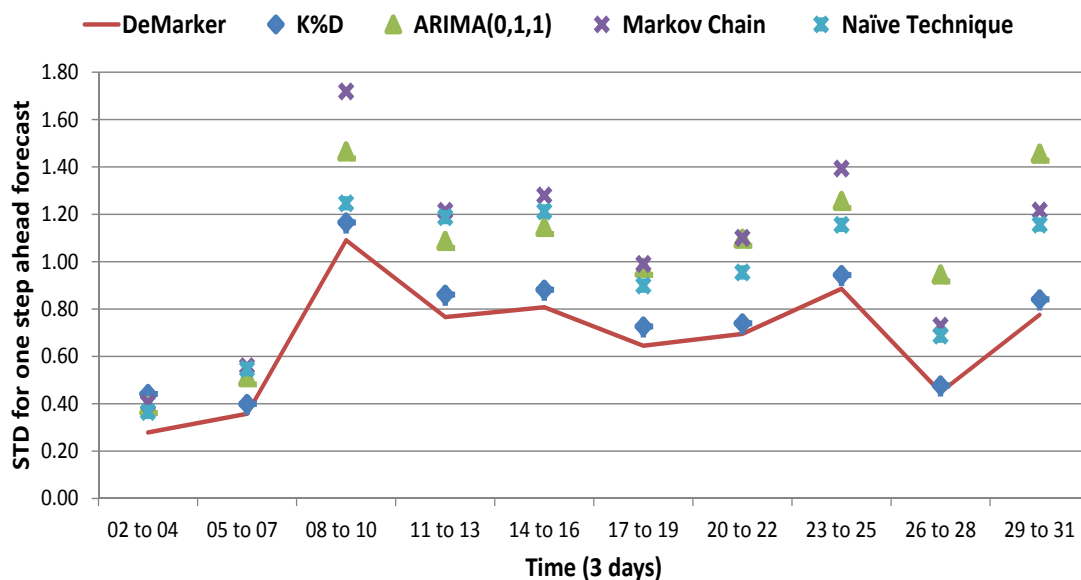


Figure 13. Comparison of stochastic models with Markov Chain and ARIMA (0,1,1) for PV for the sampling interval one minute.

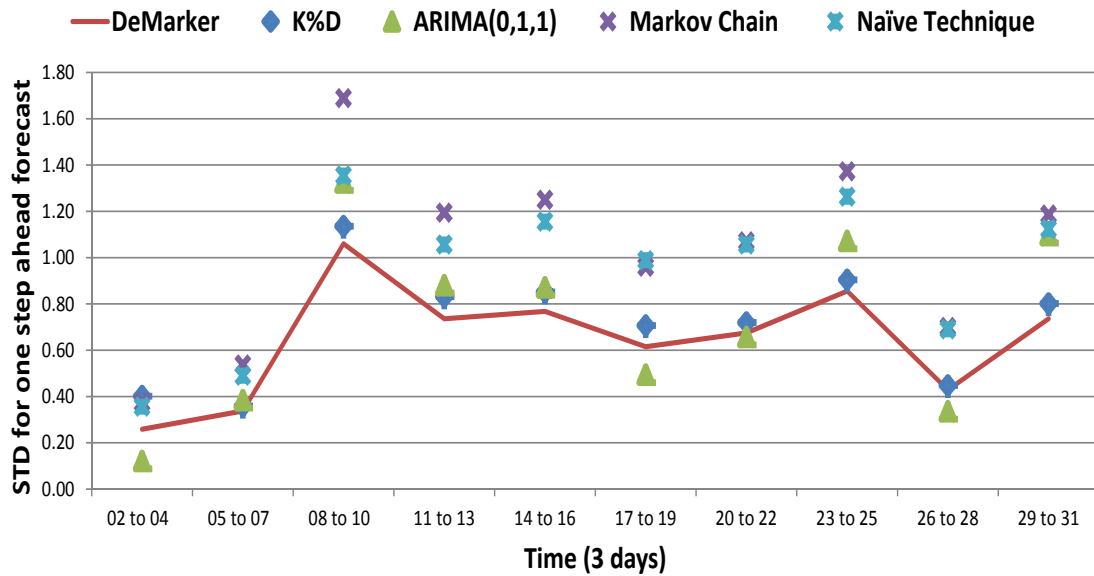


Figure 14. Comparison of stochastic models with Markov Chain and ARIMA (0,1,1) for PV for the sampling interval 10 minutes.

Taking recent history into consideration produces more accurate forecasts for this group because the stochastic model can reflect any minor change such as moving clouds or other temporary problems in the predicted result. In Figure 14, the averages of standard deviations are 0.72, 0.65, 0.72, 1.03 and 0.95 for K%D, DeMarker, ARIMA(0,1,1), Markov Chain and naive predictor, respectively. When the sampling interval increases to 10 minutes, there is no guarantee that one of the stochastic models achieves the best performance over every 3-day period but if the results of one month are compared, then DeMarker still demonstrates better results in terms of prediction accuracy.

3.5. Validation of SAO algorithm

The efficiency of the SAO algorithm in terms of valley filling/peak shaving is demonstrated by considering the energy consumed/injected by seven dwellings from the main grid during one day before and after using the algorithm. The size of the battery in this example is 1 kWh in each dwelling. In the first simulation, the power consumed from the main grid is calculated when there is no prediction strategy for the PV and wind turbine. In this scenario, the moving average of load consumption by seven households is calculated for 24 hours and the SAO algorithm attempts to minimise the gap relative to this moving average line. Figure 15 shows the result of a lack of strategy to store or consume energy when there is a considerable amount of power produced by PV or a wind turbine. The resulting power flow in Figure 15 is the sum of all consumption/injection power flow from the seven dwellings. The valley causes the reverse power flow on the main grid when the SAO algorithm is not running.

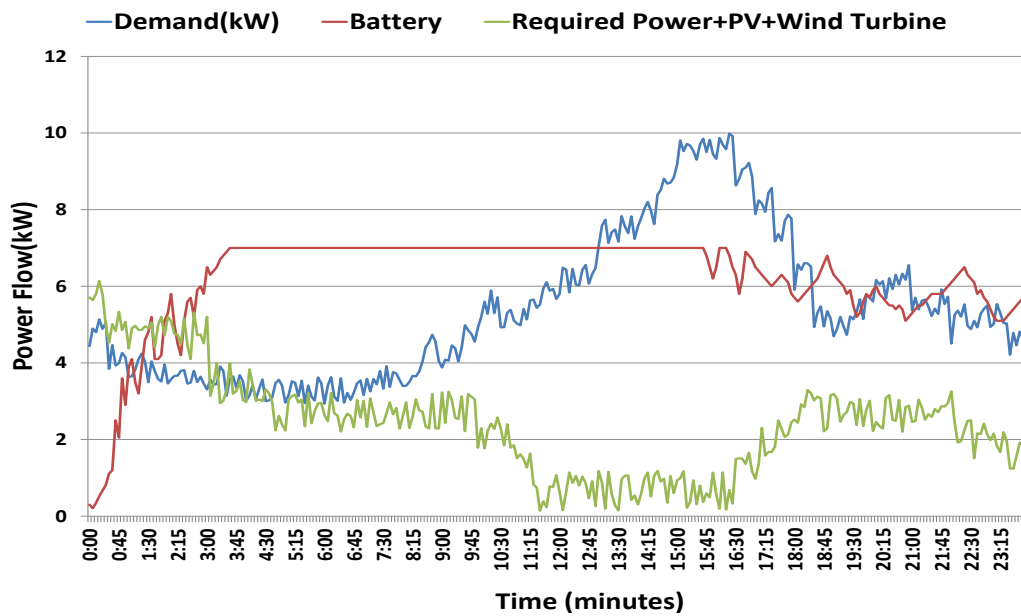


Figure 16. Power flow after implementing adaptive minimisation algorithm.

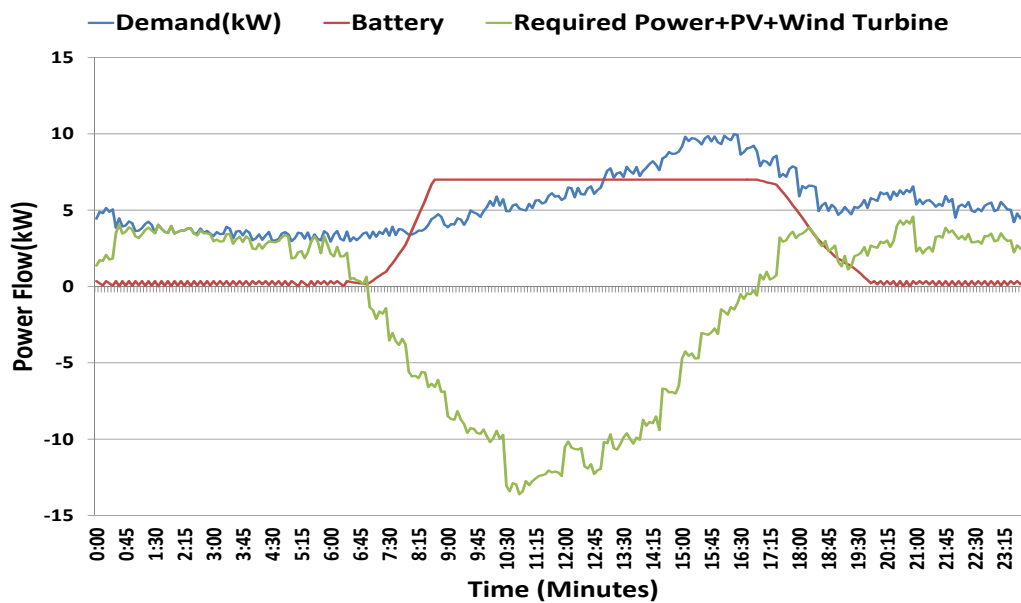


Figure 15. Power flow before using adaptive minimisation algorithm.

In the second simulation, the SAO algorithm is applied with the same size battery that lowers the moving average. Minimising the fluctuation of power flow is evaluated based on linear programming and stochastic modelling with a sampling interval of five minutes. Figure 16 shows that households can lower the gap between peak demand and off-peak hours 3 times during 24 hours by selecting a 1 kWh battery. Figure 16 also illustrates the smoothing action of the average line with no gap between peak demand hour and off peak hours.

4. Conclusion

The study reported in the current research represents a significant step forward in the stochastic modelling of energy flow in a smart grid. The proposed algorithm, developed based on moving average techniques, allows the network elements to minimise the fluctuation of the acquired power from the main grid between the off peak and peak demand hours. The work presented focused on the stochastic modelling of the *ad hoc* behaviour of renewable energy sources with or without a daily pattern. Stochastic modelling allows nodes to predict the value of injected/consumed power flow and to measure the required power to store in storage components such as batteries.

Simulation was performed based on the real time data collected from a weather station in Perth, Australia. The performance of the developed stochastic models was benchmarked against ARIMA and Markov chain models. Using linear programming, the amount of power required to be stored to match the future demand and supply was calculated. As shown, the low-level control of the nodes in terms of voltage and current adjustment was de-coupled from energy management. The scope of the linear programming carried out was limited by the assumption made that energy management was assigned to the higher layers of smart grid.

Acknowledgments

The authors wish to gratefully acknowledge the help of Dr. Madeleine Strong Cincotta in the final language editing of this paper.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Clean Energy Regulator, 2014. Available from: <http://www.cleanenergycouncil.org.au/policy-advocacy/reports/clean-energy-australia-report.html>.
2. Liu X (2010) Economic load dispatch constrained by wind power availability: A wait-and-see approach. *Smart Grid, IEEE Transactions on* 1: 347–355.
3. Parvania M, Fotuhi-Firuzabad M (2010) Demand response scheduling by stochastic SCUC. *Smart Grid, IEEE Transactions on* 1: 89–98.
4. Gong C, Wang X, Xu W, et al. (2013) Distributed real-time energy scheduling in smart grid: Stochastic model and fast optimization. *Smart Grid, IEEE Transactions on* 4: 1476–1489.
5. Su S, Lu C, Chang R, et al. (2011) Distributed generation interconnection planning: A wind power case study. *Smart Grid, IEEE Transactions on* 2: 181–189.
6. He M, Murugesan S, Zhang J (2013) A multi-timescale scheduling approach for stochastic reliability in smart grids with wind generation and opportunistic demand. *Smart Grid, IEEE Transactions on* 4: 521–529.

7. Mohd A, Ortjohann E, Schmelter A, et al. (2008) Challenges in integrating distributed energy storage systems into future smart grid. *Industrial Electronics. ISIE 2008. IEEE International Symposium on*: 1627–1632.
8. Dantzig G (1998) *Linear Programming and Extensions*, Princeton university Press.
9. Nfaoui H, Essiarab H, Sayigh A (2004) A stochastic Markov chain model for simulating wind speed time series at Tangiers, Morocco. *Renewable Energy* 29: 1407–1418.
10. Hocaoglu F (2011) Stochastic approach for daily solar radiation modeling. *Solar Energy* 85: 278–287.
11. Chen P, Pedersen T, Bak-Jensen B, et al. (2010) ARIMA-based time series model of stochastic wind power generation. *Power Systems, IEEE Transactions on* 25: 667–676.
12. Box G, Jenkins M, Reinsel G (2008) *Time Series Analysis, Forecasting and Control*, John Wiley & Sons Inc.
13. Bivona S, Bonanno G, Burlon r, et al. (2011) Stochastic models for wind speed forecasting. *Energy Conversion and Management* 52: 1157–1165.
14. A Case Study of Increasing Levels of PV Penetration in an Isolated Electricity Supply System, 2014. Available from: <http://apvi.org.au/wp-content/uploads/2014/05/Carnarvon-High-PV-Penetration-Case-Study.pdf>.
15. Nguyen C, Flueck A (2012) Agent based restoration with distributed energy storage support in smart grids. *Smart Grid, IEEE Transactions on* 3: 1029–1038.
16. Kodama J, Hamagami T, Shinji H, et al. (2009) Multi-agent-based autonomous power distribution network restoration using contract net protocol. *Electrical Engineering in Japan* 166: 56–63.
17. Lu B, Shahidehpour M (2005) Short-term scheduling of battery in a grid-connected PV/battery system. *Power Systems, IEEE Transactions on* 20: 1053–1061.
18. Sortomme E, El-Sharkawi M, (2009) Optimal Power Flow for a System of Microgrids with Controllable Loads and Battery Storage. *Power Systems Conference and Exposition, 2009*: 1–5.
19. Zong Y, Kullmann D, Thavlov A, et al. (2012) Application of model predictive control for active load management in a distributed power system with high wind penetration. *Smart Grid, IEEE Transactions on* 3: 1055–1062.
20. Zhao Z, Wu L (2013) Impacts of high penetration wind generation and demand response on LMPs in day-ahead market. *Smart Grid, IEEE Transactions on* 5: 220–229.
21. Li Y, Huang G, Xu Y, et al. (2010) Regional-scale electric power system planning under uncertainty a multistage interval-stochastic integer linear programming approach. *Energy Policy* 38: 475–490.
22. Molderink A, Bakker V, Bosman M, et al. (2010) Management and control of domestic smart grid technology. *Smart Grid, IEEE Transactions on* 1: 109–119.
23. Yu W, Liu D, Huang Y (2013) Operation Optimization Based on the Power Supply and Storage Capacity of an Active Distribution Network. *Energies* 6: 6423–6438.

24. McArthur S, Davidson E, Catterson V, et al. Multi-agent systems for power engineering applications, part ii: Technologies, standards, and tools for building multi-agent systems. *Power Systems, IEEE Transactions on* 22: 1753–1759.
25. Moslehi K, Kumar R (2010) A reliability perspective of the smart grid. *Smart Grid, IEEE Transactions on* 1: 57–64.
26. Garciaa-Ascanio C, Mate C (2010) Electric power demand forecasting using interval time series: A comparison between VAR and iMLP. *Energy Policy* 38: 715–725.
27. Shirzeh H, Naghdy F, Ciufu P, et al. (2015) Balancing energy in the smart grid using distributed value function (DVF). *Smart Grid, IEEE Transactions on* 6: 808–818.
28. Pawlowski A, Guzman J, Rodriguez F, et al. (2010) Application of time-series methods to disturbance estimation in predictive control problems. in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*: 409–414.
29. Taylor J, Menezes L, McSharry P (2006) A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting* 22: 1–16.
30. Munoz A, Sanchez-Ubeda E, Cruz A, et al. (2010) Short-term forecasting in power systems: A guided tour, *Handbook of Power Systems II*, Energy Systems, Springer Berlin Heidelberg, 129–160.
31. Pring M (2002) *Technical Analysis Explained*, McGraw-Hill, 2002.
32. Murphy J (1999) *Technical Analysis of the Financial Markets*, New York Institute of Finance.
33. Mooney C (1997) *Monte Carlo Simulation*, SAGE PUBLICATION.
34. Nguyen M, Nguyen D, Yoon Y (2012) A New Battery Energy Storage Charging/Discharging Scheme for Wind Power Producers in Real-Time Markets. *Energies*, 5: 5439-5452.
35. Endeavor Energy, 2013. Available from: www.endeavourenergy.com.au/.
36. Bellifemine F, Caire G, Greenwood D (2007) *Developing Multi Agent Systems with JADE*, England: Wiley.
37. Department of Agriculture and Food, WA, Australia, 2014. Available from: <https://www.agric.wa.gov.au/>.
38. University of York, 2009. Available from: <http://www.agentcontrol.co.uk/>.
39. Robinson C, Mendham P, Clarke T (2010) MACSIMJX: A Tool for Enabling Agent Modelling with Simulink Using JADE. *JoPha Journal of Physical Agents* 4: 1-7.
40. Shull F, Singer J, Sjoberg D (2008) *Guide to advanced empirical software engineering*, Springer.