2015-12-14

A Dynamic Adaptive Simulations Approach for Microgrid Control and Optimization

Aristotelis Emmanouil Thanos Filis
University of Miami, a.thanosfilis@umiami.edu

UNIVERSITY OF MIAMI

A DYNAMIC ADAPTIVE SIMULATIONS APPROACH FOR MICROGRID CONTROL AND OPTIMIZATION

By

Aristotelis E. Thanos

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

December 2015

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

A DYNAMIC ADAPTIVE SIMULATIONS APPROACH FOR MICROGRID CONTROL AND OPTIMIZATION

Aristotelis E. Thanos

Approved:

_____ Nurcin Celik, Ph.D. Assistant Professor of Industrial Engineering	_____ Shihab Asfour, Ph.D. Professor of Industrial Engineering
_____ Murat Erkoc, Ph.D. Associate Professor of Industrial Engineering	_____ Seok Gi Lee, Ph.D. Assistant Professor of Industrial Engineering
_____ Moataz Eltoukhy, Ph.D. Assistant Professor of Kinesiology and Sport Sciences	_____ Dean of the Graduate School

THANOS, ARISTOTELIS E. (Ph.D., Industrial Engineering)

A Dynamic Adaptive Simulations Approach for Microgrid Control and Optimization (December 2015)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Nurcin Celik.
No. of pages in text. (97)

During the past two decades, the power systems witnessed vital changes in terms of centralized paradigm versus more decentralized and market driven approaches; technical advances on communications and computation; and generation technologies which collectively lead to the advancement of microgrids (MGs). In this thesis, a novel dynamic adaptive simulations (DAS) approach is introduced for addressing major challenges in the operation and control of MGs, such as solving the economic and environmental load dispatch problem, achieving a sophisticated autonomous control of microgrids, and promoting the cooperation between individual microgrids to increase the power network reliability and energy surety. Initially, a first version of dynamic adaptive simulation was designed, namely DAS^{EELD}, for the efficient real-time economic and environmental load dispatching . The DAS^{EELD} framework was illustrated and validated via a modified IEEE-30 bus test system and as the experiments revealed, it is capable of reducing the computational resource usage for the reliable power dispatch without compromising the quality of the solution. Moreover, for the operation and control of MGs a second version of DAS was developed, namely $DAS^{CONTROL}$, in order to speed up significantly the real-time computation of the resource allocation and control decisions to optimize the operational cost, energy surety, and emissions. For validating the $DAS^{CONTROL}$

framework a realistic MG was utilized to prove that $DAS^{CONTROL}$ significantly reduces the computational burden of a considerably complex multi-objective problem. Finally, a third version of DAS was developed, namely DAS^{SH}, to provide distributed microgrids with a protocol of self-healing, both when they are operating collaboratively and competitively (in an isolated mode) while increasing the reliability of the network by pledging energy surety. DAS^{SH} framework was applied to a realistic case study that includes three microgrids and has been tested under four different emergency incidents. The results reveal that the cooperative collection of distributed microgrids were able to meet the critical and priority loads to a higher extent at all times while sacrificing from the less important non-critical loads. With the combination of the results from the different dynamic adaptive simulation versions that were created, this thesis reveals that DAS is a promising method to model microgrid systems as it provides means to find the most efficient method to optimize and enhance the microgrids' operation and control and attain several benefits.

ACKNOWLEDGMENTS

First of all, I would like to especially express my sincerest gratitude to my advisor Dr. Nurcin Celik, for her support and encouragement throughout the completion of my doctoral degree. I would also like to thank my committee members: Dr. Asfour, Dr. Erkoc, Dr. Lee, and Dr. Eltoukhy, for their invaluable input into my doctoral research. I would also like to state my appreciation to faculty and staff at the University of Miami, who have helped me a lot along this path.

Moreover, my appreciation also extends to all members and friends in our Simulation and Optimization Research Lab (SimLab): Mehrad Bastani, Haluk Damgacioglu, Duygu Yasar, Talal Alyamani, Delante Moore, Xiaoran Shi, and Juan Pablo Saenz. I have been greatly influenced by their great passion in research and pursuit of new knowledge. I will always remember my experience cooperating with them and will always cherish our friendship.

Finally, I cannot express enough thanks to my father Costas, my mother Eleni, my brother Aggelos, the rest of my family, and my friends in Greece, who always cheer me up, give me unconditional love, and stand by me through the hard times.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

Modern society has come to depend on the easy and reliable flow of electricity. When a major power outage occurs, damages and the inability to conduct normal life operations cause billions of dollars in losses per day, in addition to costing human lives. America's great wake-up call about the severe cost of insufficient investment and oversight in its electrical infrastructure came on August 14, 2003 when a widespread power outage hit many parts of the Northeastern and Midwestern United States as well as the Canadian province of Ontario. The US electrical system is so dynamic and interconnected that it's impossible for human operators to make the necessary changes (sometimes needed in a fraction of a second) to quarantine problems in the network. When the dust finally settled, over 50 million people were without power, and the majority were without power for the adjacent two days or more. It's estimated that this particular outage cost the North American economy 6 billion dollars.

As we come into an age of even more heightened electricity usage with the advent of electric/hybrid cars and the like, these inadequacies in the traditional grid structure seem highly unlikely to be solved. In fact, it is found by some measures that the frequency has actually increased slightly between 1984 and 2006. One of the concerns highlighted as being a potential contributory factor is the high usage of high-voltage transmission lines used for long distance transfers. The national grid study demonstrates the increased usage of Transmission Loading Relief over the years. This is used as a rough measure of system stress which is obviously heightened as these high-voltage lines take on added loads.

Having electricity travel such lengthy distances also comes with its share of inefficiencies. The share of power dissipated in transport as well as breaking down the giant voltages that come with high-voltage lines into domestically appropriate figures is

estimated to be 7% of the total power used in the US. It also makes it harder for smaller renewable sources to efficiently function as part of the grid leading to customers to prefer an "off the grid" approach in some cases. Traditional grid systems huge bulk also impede them from dynamically changing to meet demand as patterns of high demand develop in new areas. These substantial load variations lead to heightened inefficiencies in the entire system as energy consumption continues to evolve.

The traditional grid system also leaves areas of extreme importance which need unconditionally reliable power to operate properly (military bases, hospitals, certain private and state services) at risk. Energy surety is of absolute importance to these enterprises especially in a crisis situation. Traditional power services are too unwieldy to ensure power to certain priority sections for these customers and so, those customers must rely on back-up generators which can also fail.

The lack of strict regulation and allowing the controls to be in the hands of humans, who may prioritize politics or profits over integrity of the system, puts all of us at risk of a collapse. In an effort to cope with these unfortunate situations, microgrids, especially with the complementary use of the Dynamic Data Driven Application System paradigm, may provide part of the answer.

1.1. Overview of Microgrids (MGs)

Microgrids are autonomous electricity environments that function within a larger electrical system. They are able to react to a crisis (or advise a human operator on how to react) inside the necessary time frame in order to contain local problems when necessary. They have potential as a mechanism to increase reliability and efficiency while also offering new services which are unavailable under a traditional grid system [1]. These new

services may include dynamic pricing [2] and distributed energy resource management [3]. A typical microgrid can cater to many different types of consumers including customers with traditional demands, customers with sensitive loads, and customers who have micro-sources consisting of photovoltaic panels, wind turbines, fuel cells, micro-turbines, diesel generators, or battery storages [4].

More specifically, a microgrid is an energy distribution system consisting of both renewable and conventional power generation sources along with some form of energy storage. A microgrid system is capable of operating both alongside of a municipal power grid or as an "island" separate from the local utility grid. The purpose of a microgrid is to provide energy security as well as an uninterrupted source of energy for its prescribed customer. An added benefit of microgrids is that the use of renewable energy sources allows for significant cost savings in terms of utility bills. Microgrids can be conceived as a good example of a System of Systems. In order to meet the requirements of a System of Systems, a collection of systems must possess operational independence, managerial independence, geographic distribution, emergent behavior and evolutionary development. A microgrid combines software, hardware, policy, and technology to form a more reliable, secure, robust, and convenient alternative to its current traditional counterpart grids. Microgrids satisfy all of the properties of a system of systems as they include several operational and managerial independent component-systems such as distributed and renewable energy sources and energy storage that are geographically separate from each other. Additionally, the component-systems present emergent behavior due to the containment of renewable distributed energy resources that are driven by weather and are subject to random outages to which a microgrid must adjust. They are also highly complex

in that the control system must effectively manage all of the component systems, ensuring that each component works to the benefit of the overall system. Finally, a microgrid presents evolutionary development as it can accept new power generation as storages capabilities as they become available to either meet the increased demand needs of the customer or to further reduce their costs. As such, a microgrid fits perfectly into a system of systems construct.

The ever-increasing demands placed on the nation's power grid and local electrical utilities present significant challenges in the effective management and control of a microgrid. First, microgrids require sensored meters in order to measure the customer's consumption and patterns of demand. Without this, the microgrid would not be able to effectively detect faults in the system or provide secure power to critical assets in the event of an emergency. Secondly, microgrids require a much more robust means of communication than presently in use in the local utility transmission and distribution networks. The controls system in a microgrid uses a great deal of telecommunications resources in order to send price and status signals and receive information from consumers in real time. The next issue is that very few local utility grids can properly handle renewable resources. This is due to renewable energy resources that are by nature intermittent and non-dispatchable. Furthermore, the system operators for existing transmission networks do not possess ability to manage power generation equipment not owned by their specific utility company. For example, an orange farm that normally receives service from a local utility decides to install a diesel generator to supply power to their farm. The local utility company cannot control how much electricity the generator produces at any given time or when the owner of the farm decides to turns the generator on or off. This causes problems

for the utility company in that the generator on the orange farm in that it reduces their ability to forecast how much electricity they need to generate to meet the needs of their customers. Also, with the generator running, customers may not be able to utilize all of the generated electricity leading to waste for the utility. Finally, existing local utility grids cannot support new technical developments such as vehicle-to-grid technology. Such technologies allow electrical vehicles to store energy when the grid's demand is low and give back to the grid when the demand is high.

Viewing a microgrid in terms of levels of control helps to address the implementation challenges associated with installing a microgrid in parallel to a local utility. There are three main levels of control when referring to an electrical network. These are strategic, tactical, and operational. The strategic level provides a high level overview of the entire system and provides a long-term vision for the future of the overall system. This is where we address the problems associated with robustness of communications systems, sensored meters and operators' inability to control components outside of their network. At this level the microgrid controller ties into an existing electricity substation and monitors how much electricity the utility company is providing as well as the demand of the customers serviced by the microgrid. The most important aspect occurring at this level of control is the decision to isolate the microgrid from the local utility in the event of a power outage or national emergency. Component systems inside a microgrid that operate at the strategic level of control would be the computers mainframes and servers that oversee the overall electricity distribution system. Below the strategic level is the tactical level. This level is where we address problems stemming from the introduction of renewable energy resources into an electrical utility network. At this level, the microgrid determines how much electricity to

purchase from the local utility based upon whether or not the renewable energy resources meet the current consumer demand. It also determines when it is most cost effective to store excess energy or sell it back to the utility company using a form of net metering. The lowest level of control is the operational level. This is where we address the issues of integrating new technologies into an electrical distribution system. Component systems operating at the operational level would include the photovoltaic array, a wind farm, energy storage, and electric vehicles all of which require integration with a microgrid controller as well as some form of tie in to the local utility. Breaking down microgrid component systems into its different levels of control allow designers to effectively solve the major challenges associated with installing an electricity network operating both in parallel and separated from a local utility network.

Traditionally, modeling techniques for the strategic, tactical and operational control of microgrids involves a multi-pronged approach where the development of a theory, a model, and a set of experiments happen in succession.

1.2. Agent-based Simulation

A great way to simulate the behavior of a system of systems such as a microgrid is using an agent-based simulation model. Agent-based simulation is a cutting-edge technique for modeling systems composed several autonomous “agents” that interact with one another. An agent-based simulation must contain three components: A collection of agents, the relationships between the agents, and the environment in which the agents interact. The manner in which the individual agents interact with one another influences the overall behavior of the system. Modeling agent interactions with each other allows

patterns and behaviors to emerge that were not necessarily present or pre-programmed into the base model. The emergent and evolutionary behavior of these multiple independent agent interactions distinguishes agent-based simulations from other simulation techniques such as discrete-event and system dynamics. Agent-based modeling offers a novel approach to the simulation of electricity distribution systems. This is because multiple independent agents (power generation resources) that interact with and feed into a larger network create the structure of an electricity distribution system. This larger network learns from their capabilities and adapts the overall system behavior to better suit the needs of its customers.

Within this collection of agents, the most important characteristic of an agent is its ability to act without any external input. In general, agents act to achieve their own individually specified ends without regard for the actions of the agents in their environment. In terms of building a simulation model based on agents, each agent must possess certain characteristics. First, an agent must be individually distinguishable. This means that a person can determine whether a component in the simulation model is part of a specific agent. All agents in the environment have features and characteristics that distinguish them from but make them recognizable to all other agents. Next, agents are autonomous. Each agent is capable of performing its prescribed function independently in their given environment. Individual agents detect information and translate it into outputs (behaviors) that impact the entire system. Agents also possess states that vary over time. Similar to how the overall system has a state that is the collection of its state variables, an agent also has associated variables that determine its present behavior over time ([5] and [6]). Lastly, each agent has rules defining its interactions with other agents. These rules can

include but are not necessarily limited to how different agents communicate with each other and their ability to respond to changes in the environment or other agents. In addition to these requirements for all agent-based simulation models, particular models, such as those for a microgrid, requires that agents be heterogeneous. For an agent-based microgrid simulation to function properly, the model must consider the full spectrum of diverse agents operating within the environment. The simulation model gives each agent a different amount or resources and will in turn produce varying amounts of energy as a result of the different agent interactions.

1.3. Economic and Environmental Load Dispatch Problem

The main challenge in microgrids is to solve the economic and environmental load dispatch problem optimally while minimizing the computational resources. The goal of the economic and environmental load dispatch (EELD) is to produce electricity at the lowest cost and emissions to reliably serve customers, while recognizing the operational limits of generation plants and transmission lines ([7] and [8]). The dispatching of loads is performed to control and allocate the total energy generation amongst the available resources (including both conventional and renewable sources) within a microgrid.

The total demand of electricity is increasing day by day. For the load demand satisfaction a large number of thermal power plants have been utilized and the amount of the coal burnt keeps increasing. However, due to the aforementioned increasing amounts of coal being burnt, several toxic gases are emitted (such as carbon dioxide, sulphur dioxide, and nitrogen oxides) and significantly contribute to the pollution of the environment. Environmental pollution substantially impacts the global warming which in

turn damages the Ozone layer. To that end, it is highly desirable to produce power minimizing the costs while at the same time minimizing the amount of pollutants emitted. Hence, the study of EELD focuses on generating power on minimum costs and minimum emissions.

1.4. Dynamic Adaptive Simulation in MGs

In this thesis, we focus on creating framework of Dynamic Adaptive Simulations (DAS) by exploiting the Dynamic Data Driven Application Systems (DDDAS) paradigm in order to improve the operational control of microgrids, solve the EELD problem optimally and efficiently, and promote the cooperation between individual microgrids to increase the power network reliability and energy surety. DAS is a new modeling and control method which adaptively adjusts the fidelity of a simulation model. This paradigm steers the measurement process for selective data update and incorporates the real-time dynamic data into the executing model.

To this end, this work contributes to the advancement of DDDAS in achieving a significant reduction in the computational resources required to conduct real time simulation. We achieve this by only looking at the detailed specifics of the system whenever the microgrid detects an abnormality, otherwise having the simulation in a more overseer mode. Combining the concepts of DAS and microgrids produces the embodiment of a system of systems operating at the operational level of control as each component system within a microgrid can operate independently for its own individual purpose. Furthermore, the Dynamic Adaptive Simulations running within the microgrids uses sensors, computational resources, and various algorithms to link all of these separate

systems together with varying levels of control to effectively manage a complex power distribution system.

Chapter 2: Literature Review

The Dynamic Data Driven Application Systems (DDDAS) is a promising and evolving new technology with, in recent years, direct engineering and science applications [9]. Recently, many researchers ([10]-[15]) began to study new applications for adaptive simulations such as DDDAS. Park et al. develop a new method of data reduction to enhance data transmission for sensors used in structural health monitoring [16]. Khaleghi et al. propose a DDDAS framework for effective surveillance and crowd control using unmanned aerial and ground vehicles [17]. Blasch et al. explore the concept of an adaptive fine-tuning of secure communication trust analysis that aims to achieve a balance between standard static solutions versus dynamic data driven solutions [18]. A DDDAS assists in filling in the gaps of a complex (or even real world) model, while also improving itself semi-autonomously [19]. Each DDDAS however, must be constructed differently depending on the requirements of the current experiment. Electric power distribution networks, more specifically microgrids, are one of the application areas to make use of the decidedly effective measurement and control processes available by utilizing DDDAS modeling techniques. To this end, [20] defines a microgrid as a system of systems and apply the DDDAS paradigm to select the best simulation fidelity for a military microgrid. [21] and [22] apply a DDDAS framework to power network systems to address the economic load dispatch problem. Finally [23] proposes a DDDAS framework for automated control in microgrids.

One challenge in modeling a microgrid using Dynamic Adaptive Simulations is automatically adapting the simulations when experimental data indicates that a simulation must change. Carnahan and Reynolds draw attention to this challenge and determined that the goal of the first run of every simulation is to gain insight about a particular phenomenon

[12]. We then use this insight to determine what new observations to collect, and then adapt the simulation to reflect these observations. Here, attempting to generalize software so that it is able to anticipate each possible way to change will significantly impact its performance and make its underlying code unmanageably intricate [24]. Carnahan and Reynolds, therefore, propose a semi-automated adaptation approach that exploits the flexibility and constraints of model abstraction opportunities to automate simulation adaptation [12]. While their study does not involve manual modification of the code or application of optimization methods, which can make the software extremely complex to control, it is still in need of human intervention to determine the most likely places of the code that should be changed.

Modeling an electrical distribution system, especially one that utilizes renewable energy resources such as a microgrid requires the collection and processing of a substantial amount of sensor-based data between a variety of different devices and interface. In Thanos et al. and Celik et al. ([21] - [23]), changes in level of detail of data acquisition and the choice of certain parameters over others allow the automatic multi-fidelity adaptation in the simulation model. In addition to the development of interfaces to physical devices, they also address the creation of an infrastructure to support the communication and data requirements. Through a networked sensor-driven control system and integrated grid architecture for distributed computing, they proposed a methodology for the timely transfer of up to date data along the layers of the simulation model in real time.

Integrating all of the component systems of a microgrid into a managed electrical network capable of tying into a local utility requires a great deal of computing power. One of the significant benefits of a DDDAS is its ability to reduce the computational power

required to accomplish its tasks. Wang et al. proposed a wide-ranging planning and control structure for unmanned aerial and ground based vehicles based on adaptive simulations [13]. In their research, Wang's team proposed processes to update both intelligence and surveillance data dynamically. Additionally, their research allows operators of such vehicles to merge and interpret time and geographical data to facilitate improved use of unmanned systems for surveillance and crowd control. Their use of integrating real-time ever-changing data into a simulation model seamlessly ties into these methodologies for simulation in distributed electricity systems. This allowed them to reduce the computational power required for simulation without sacrificing accuracy. Moreover, Frew et al. applied a DDDAS to continuous detection in intricate atmospheric conditions [14]. Their research used novel on-board and remote wind monitoring techniques and capabilities. Furthermore, they developed a control framework that autonomously adapts their system to maximize efficiency of environmental, sensory, and computational resources. Their discoveries proved quite useful in efforts to combine multiple renewable energy sources into a dynamic model to reduce electricity costs to the customer.

The primary purpose of a microgrid is to combine multiple energy generation and storage capabilities and make them seamlessly work together to provide electricity to the customer. For a microgrid to be completely effective, it is critical to know a service interruption in one energy source impacts the performance overall microgrid performance (and potentially other energy sources). Han and DeLaurentis research into managing the complicated interdependencies between component-systems also proved extremely useful in mitigating such issues which may result is exceeding the cost and time constraints of the customer [15]. In relation to research with distributed energy systems, Han and

DeLaurentis highlight the need to understand the capabilities and limitations of each component-system that composes the overall system [15]. Additionally, they emphasize the need to develop protocols to mitigate the risks of disruption to the overall system propagating from deficiencies in one or more component. In other words, when designing and building a microgrid, it is critical to understand not only how component systems act when combined as a whole, but how each component system interacts with every other. The aforementioned research provides a great deal of insight into how to effectively model a system of systems as well as methods to reduce the computational resources needed to model real time systems data. Our DAS approach differs from previous research in that in addition to collecting sensory data at given intervals; it collects more detailed data when it detects an abnormality in the system (not meeting demand, not enough load on a bus, etc.). Furthermore, this approach is the first to look at different portions of the grid in different levels of detail. There is no need to devote significant computational resources to portions of the grid that previous computations have already determined are operating efficiently and effectively.

While the concept of a self-healing grid infrastructure is quite promising in improving the reliability of power distribution networks, there are several challenges faced during implementation into existing grid structures. The most crucial of these challenges is to identify potential vulnerabilities within the network and to decide on the appropriate system response. Addressing this problem, Hoffman et al. studied a symbiotic feedback control system to link sensing, visualizing, and modeling to reduce power system vulnerabilities that cause blackouts [25]. However, the scope of Hoffman's research was limited to legacy power grid structures, and included limited renewable energy penetration

with no self-healing characteristics. This work identifies numerous potential vulnerabilities that may lead to blackouts, but applies them to a self-healing microgrid network with extensive renewable energy penetration.

Identification and response to inherent vulnerabilities of the network in a microgrid environment also necessitate enhanced understanding of the complex interdependencies that exist within it. Electric power grids (including microgrids) contain strong interdependencies among their components. This results in a cascading effect on the remainder of the grid once an action or fault in one portion of the grid takes places. Amin and Giacomoni investigated the cause and effect relationships within an electrical grid in order to determine possible self-healing procedures that could be taken as a response to threats and component failures of the considered system [26]. Furthermore, Han and DeLaurentis stressed the importance of fully understanding the capabilities as well as limitations of every component and device within a network [15]. They emphasized the need to develop protocols to mitigate the risks of a fault in one component or portion of the network severely affecting the performance of the entire system. In other words, when designing and building a self-healing power grid, it is critical to understand how individual microgrids operate. However, it is just as critical to understand how they respond when interconnected with one another. Equally important to these two obligations in our process is the ability of the individual microgrids to react to major faults occurring within its neighboring microgrids.

Moslehi and Kumar looked into the impact of using renewable energy resources on the reliability of the power network and found that, due to their intermittency, a mix of several renewable energy generation resources is necessary to allay the reliability

challenges inherent in distributed generation networks [1]. However, to achieve self-healing among several independent microgrids there needs to be a switching method to prevent loss of power when the intermittency or other phenomena that may lead to a power outage occurs. Zhi devised a method to make power grids run continuously without losing load [27]. He developed a framework consisting of control circles, layers, and links to manage how different portions of a power grid communicate with each other.

Chapter 3: Dynamic Adaptive Simulation for Efficient and Optimal Power Dispatch in Microgrids (DAS^{EELD})

Environmental and economic load dispatching for microgrids is a challenging task due to several reasons. First, microgrids are highly dynamic and complex in their nature due to the variability in their status induced by different sources of energy generation and their associated generation capacities, environmental emissions, frequency of changes in load profiles, market policies and regulations, and revenues generated, amongst many others [28]. Second, microgrids may operate at various scales and scopes, causing the range for the solution space to be considerably large. Third, the inclusion of renewable energy into microgrids, has resulted in additional constraints on EELD such as more unpredictable ramp rates and the need for additional reserves to accommodate the intermittent nature of the output. Furthermore, this dynamicity and complexity inherent to these problems enforce significant burden on the available computational resource utilization while the developed solution procedures are being deployed, even if they are performed at specified intervals or offline. This burden further frustrates the monolithic implementation of the methodologies presented in the literature ([29] and [30]) in a realistic setting and necessitates a distributed framework for effective decision making within these systems.

In this chapter, we propose a dynamic adaptive simulation framework (DAS^{EELD}) for the EELD problem. The proposed framework is overviewed in Fig. 1. The overall scheme envisioned is a robust multi-scale federation of simulation models that enables efficient and optimal power dispatch in power networks.

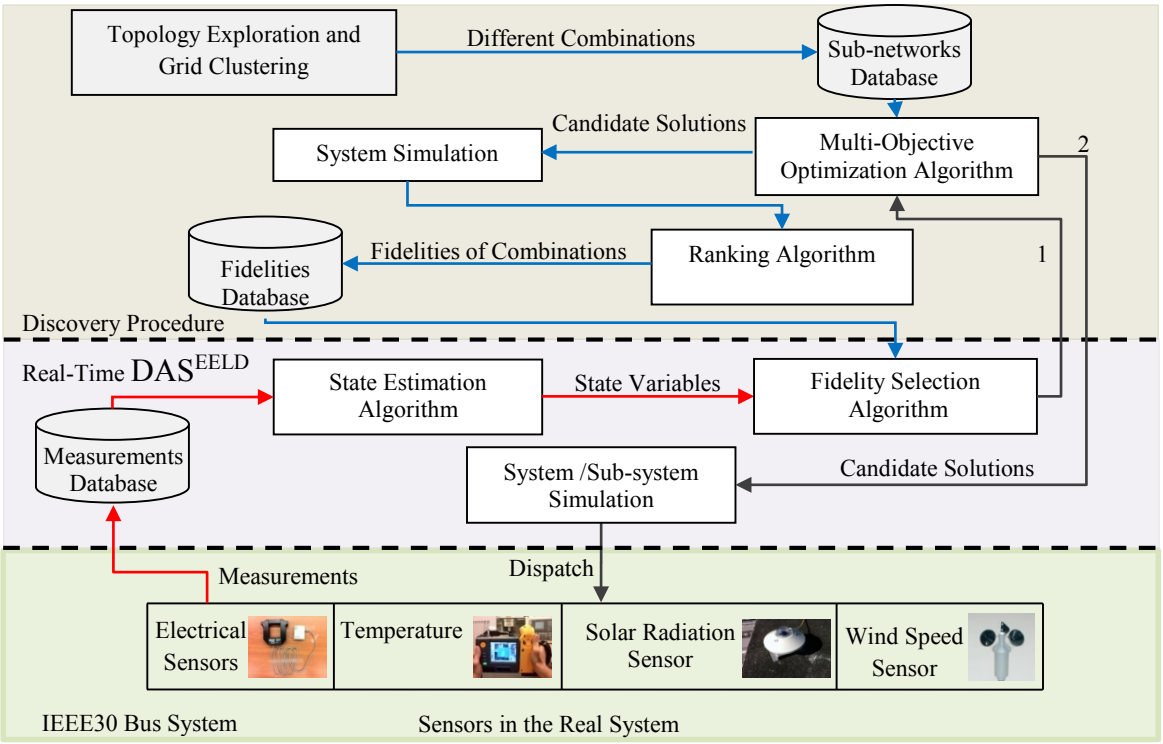


Fig. 1. Overview of the proposed DAS^{EELD} framework

The proposed DAS^{EELD} framework includes an offline discovery procedure and a real-time decision making procedure. The offline discovery procedure incorporates algorithms for grid topology and clustering, multi-objective optimization, fidelity ranking; and databases for sub-networks and fidelities. The grid topology and clustering algorithm examines the structure of the power network and determines the different possible sub-networks that may be built to compose the full power network.

Based on the results from the topology and clustering algorithm, the different sub-networks and their combinations are used to generate power dispatch solutions under various predetermined load scenarios. Performances of these power dispatches are evaluated based on the best-compromise solution generated for each of the sub-network

combinations in terms of their costs and emissions. Then, for each of the different load scenarios, the combinations are ranked using fuzzy logic.

The DAS^{EELD} framework embodies a measurements database that is fed using electrical and environmental sensors. Given the sensory data and available computational resources, a state estimation algorithm, and a fidelity selection algorithm are invoked to determine the state of the system and fidelity that the simulation should be performed at. Based on the estimated system status and the selected fidelities, a multi-objective optimization algorithm is employed to generate a non-dominated solution set in terms of costs and emissions. Then, a best compromise solution is selected and sent to the actual power network for realization. The details of the components embedded in the proposed framework are presented in the following sub-sections.

3.1. Multi-objective Optimization Algorithm

The proposed DAS^{EELD} framework is employed to provide the considered power network with the best possible solution, which is the environmental and economic load dispatch (EELD) in this case. Because of the multi-objective nature of the EELD problem, a multi-objective optimization algorithm is incorporated into the proposed framework. The details of the algorithm are presented below.

3.1.1.Formulation of the EELD Problem

The EELD problem has two distinct objectives, namely, minimizing the generation costs and minimizing the pollutant emissions of a power network's load dispatch while acknowledging the system's limitations. The problem is formulated in eq. (1) through (7),

where the decision variables are the real (P_G) and reactive (Q_G) power produced at each generation bus. Equations (1) and (2) present the cost and emissions objectives, where, a_i, b_i and c_i are cost coefficients, N_G is the number of generating units, P_{G_i} and Q_{G_i} are the real power and reactive power generated, and α_i, β_i, γ_i, ϵ_i and μ_i are the emissions coefficients.

$$Minimize\ F(P_G) = \sum_{i=1}^{N_G} a_i + b_i P_{G_i} + c_i P_{G_i}^2 \tag{1}$$

$$Minimize\ E(P_G) = \sum_{i=1}^{N_G} [10^{-1}(\alpha_i + \beta_i P_{G_i} + \gamma_i P_{G_i}^2) + \epsilon_i e^{\mu_i P_{G_i}}] \tag{2}$$

The constraints of the problem are presented in eq. (3)-(7). Equation (3) represents the generation capacity constraint which ensures that all energy generating plants operate within their capacity. Equations (4)-(7) represent the power balance constraints which ensure that the load provided to the system meets the demand while taking energy transmission losses into account.

$$P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max} \qquad \forall_i \tag{3}$$

$$\sum_{i=1}^{N_G} P_{G_i} - P_D = P_{loss} \tag{4}$$

$$P_{G_i} - P_{D_i} - V_i \sum_{j=1}^{N_B} V_j [G_{ij} cos(\delta_i - \delta_j) + B_{ij} sin(\delta_i - \delta_j)] = 0 \qquad \forall i \tag{5}$$

$$Q_{G_i} - Q_{D_i} - V_i \sum_{j=1}^{N_B} V_j [G_{ij} sin(\delta_i - \delta_j) + B_{ij} cos(\delta_i - \delta_j)] = 0 \qquad \forall i \tag{6}$$

$$P_{loss} = i \sum_{i=1}^{N_G} g_k [V_i^2 + V_j^2 - \cos(\delta_i - \delta_j)] \tag{7}$$

Here, $P_{G_i}^{min}$ and $P_{G_i}^{max}$ are the minimum and maximum operating output of unit i, respectively, N_B is the number of buses, P_{D_i} and Q_{D_i} represent the real and reactive loads at bus i, V_i is the voltage magnitude at bus i, G_{ij} is the transfer conductance between buses i and j, δ_i is the voltage angle at bus i, B_{ij} are the transfer conductance and susceptance between bus i and bus j, g_k is the conductance of the k^{th} line.

3.1.2.Multi-objective Optimization using Particle Filtering Algorithm

The EELD optimization problem described in Section 3.1.1, may be alternatively represented by (8) and (9), where x, m, and x^* are the decision vector, number of decision variables, and Pareto optimal solution set, respectively. Furthermore, $f_1(x)$ and $f_2(x)$ represent equations (1) and (2), respectively.

$$x^* = arg\,min\, f(x) = arg\,min\big(f_1(x), f_2(x)\big) \tag{8}$$

$$x = (x_1, x_2, \dots, x_m) \in R^m \tag{9}$$

The state space model defined for the particle filtering based multi-objective optimization is provided in (10) and (11), and the importance density function is defined in (12). In these equations, $x_k = \big(x_{k,1}, x_{k,2}, \dots, x_{k,m}\big)$ is the state of the system at time k, $y_k = \big(y_{k,1}, y_{k,2}, \dots, y_{k,n}\big)$ is the measurement taken at time k, $v_k = \big(v_{k,1}, v_{k,2}, \dots, v_{k,n}\big)$ is the measurement noise distributed with a pdf $\varphi(\cdot)$.

$$x_{k+1} = x_k, \tag{10}$$

$$y_k = f(x_k) - v_k, \tag{11}$$

$$q_k(x_k) = \frac{\varphi(f(x_k)-y_k)q_{k-1}(x_k)}{\int \varphi(f(x_k)-y_k)q_{k-1}(x_k)dx_k} \tag{12}$$

The algorithm is based on a particle filtering procedure that includes two-sampling stages. In the first stage, samples are taken from within the non-dominated solution set generated by the algorithm. In the second stage, a sampling distribution is generated using the solutions with the best performance in each of the separate objectives, and then samples are drawn from this distribution. The number of samples, an empty non-dominated set, and the number of iterations, are defined as the algorithm's input. Once initialization is completed, the data for buses, lines, and cost are used for random sampling. The admittance matrix is then updated to reflect the distributed generation levels from the random dispatch, and the resulting loads and the equivalent resistance are calculated. In the next step, the resultant power generation as well as the energy transmission losses is evaluated, and the dispatch at the swing bus is adjusted to ensure the power balance constraints are met. Then the non-dominated solution set is calculated and the resampling stage is triggered. The new samples obtained from the resampling at each distribution level are then used to update the admittance matrix sequentially. Once the predefined number of iterations is reached, the final non-dominated solution set is calculated and the corresponding objective values are exported.

3.2. Discovery Procedure

In the discovery procedure, the topology of the power network is explored, so that different potential sub-networks are identified, to guide the DAS^{EELD} with a predetermined set of simulation fidelities. This is achieved with a decomposition technique, through which the entire network under consideration is decomposed into n non-overlapping observable sub-networks [31]. Furthermore, each of these sub-networks must include at least one source of energy generation. Once the network is decomposed, the information of all the sub-networks and their combinations is stored into the sub-networks database. The number of items stored in the database is $2^n - 1$, where n is the number of sub-networks.

Once the combinations are defined, demand levels are selected in order to generate different scenarios with which different fidelities will be evaluated. In this step m levels of load variation are selected. Each of the sub-networks is mapped to a load with a variation corresponding to each of the m levels, so that the permutation of the levels within the sub-networks generates the number of different scenarios. At this point it is important to highlight that a 0% level is always included within the m levels of load variation, and the simulation is not performed when the demand in all the sub-networks is at this level; thus the total number of scenarios is $(m^n - 1)$. Based on the number of sub-network combinations and levels, the total number of simulations performed by the discovery procedure is $(2^n - 1)(m^n - 1)$.

It should be noted that the selection of the demand levels have a significant role in the accuracy of the proposed framework. On one hand, if few but very different levels of load variation are selected, the predetermined fidelities may not provide a good approximation for the demand variations of the real-time simulation. On the other hand, if

many but close levels are selected, the accuracy of the fidelity selection obtained for the power dispatch may be optimal. However, because of the permutation involved in the generation of scenarios, the discovery procedure becomes unrealistic in the latter.

A set of non-dominated solutions is generated for each of the $(2^n - 1)(m^n - 1)$ simulations based on cost and emissions. They are then ranked for each of the $(2^n - 1)$ sub-networks for each of the $(m^n - 1)$ scenarios, based on best compromise solution, using fuzzy logic. Finally, the rankings are saved into the fidelities database.

3.3. State Estimation Algorithm

The real-time state estimation algorithm for computing the electricity demand is triggered by measurements obtained from the sensors in the real system via the interaction with the sub-networks and fidelities databases. Efficient state estimation is crucial in this study since it significantly affects the control of the power flow, fidelity selection, security of the system, and performance of the load dispatch. To this end, in our proposed DAS^{EELD} paradigm, accurate estimates of real-time electricity demands are obtained via a smart sampling algorithm whose seeds were planted in their earlier work (i.e., [32]). The demand is estimated at the distribution level from smart sampling perspective using two sub-procedures whose operations are explained below.

3.3.1.Sub-procedure I

The goal of the first sub-procedure is to estimate the real and reactive power injections of the considered electricity network using environmental measurements (i.e., temperature readings), to incorporate the environmentally-driven impacts. To be specific, during cold days, temperature increments lead to a decrease in the electricity demand, due

to reduced heating demands, while on hot days, these temperature increments increase the electricity consumption due to higher cooling demands. To this end, the state-space model for electricity demand estimation in the first sub-procedure is given in eqs. (13) and (14).

$$D_{k+1} = \alpha D_k + U \tag{13}$$

$$D_k = \beta T_k + V \tag{14}$$

where D_{k+1} and D_k are the posterior state (i.e., demand), current state T_k is the current temperature, α and β are parameters related to state evolution and observation functions that are statistically calculated from historical data, and U and V represent the process noises and measurement errors, respectively.

3.3.2.Sub-procedure II

For the purpose of modifying the minor variation of the estimates and increasing the estimation accuracy, in the second sub-procedure, the available measurements of the electrical parameters (i.e., voltage magnitudes, power injections, power flow, etc.) are employed. Then, the refined state-space model for this sub-procedure is given in eqs. (15) and (16) as follows:

$$D_{k+1} = \gamma D_k + U \tag{15}$$

$$D_k = \mu(D_k) z_{k,j} + V, \quad \forall j \in \{1,2,\dots,J\} \tag{16}$$

Here γ is a parameter computed in a way analogous to that of α and β in the first sub-procedure, $\mu(\cdot)$ is a function relating the measurements to the power injection states, and J is the number of different measurements within any corresponding time interval t.

The states of the network are set as the real and reactive power injections of the buses. Data collection frequency (time interval estimation) is determined on the basis of load variation and response times of the available energy resources. The limits of these frequencies are governed by the fastest possible response time of energy resources, and the maximum duration in which the load variation is kept unchanged. Higher frequencies of data collection lead to a higher estimation accuracy and lower frequencies result in lighter computational burdens. Consequently, data collection frequency should be decided considering this trade-off. Once this frequency is determined, the algorithm generates four state variables corresponding to real and reactive power injections for either "weekday" or "weekend day" for each bus. Figure 2 presents the operation of the state estimation algorithm.

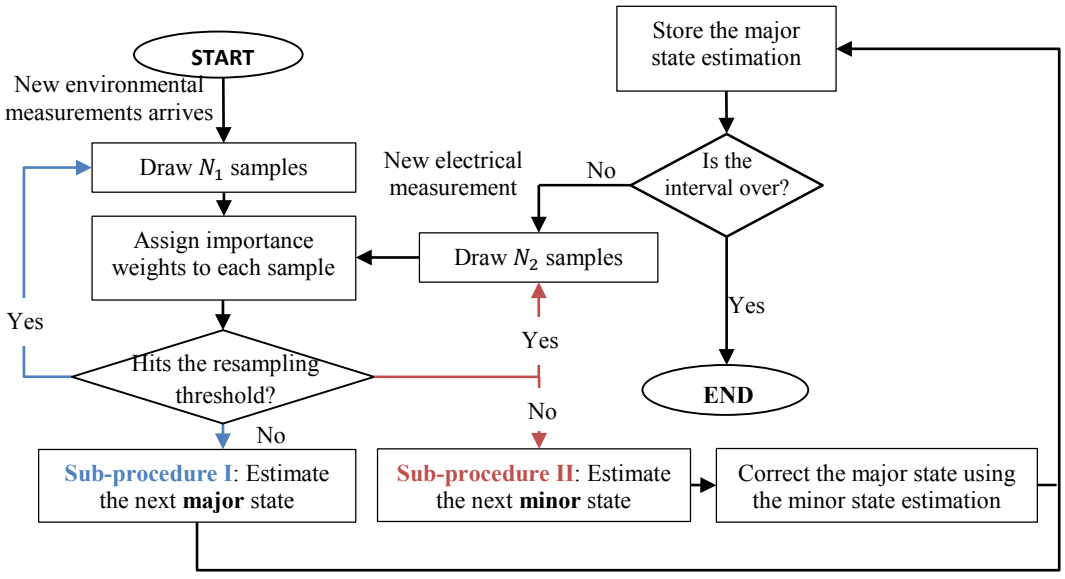


Fig. 2. Flowchart of the embedded stated estimation algorithm

3.4. Fidelity Selection Algorithm

The effective culling and fidelity selection algorithm is designed to determine which sub-networks should be included in the DAS^{EELD} simulation and which sub-networks' dispatch should remain unchanged. This way, a near optimal dispatch may be attained while ensuring an acceptable computational burden.

Whenever the dispatch of the system is to be updated, either because of periodic revision, or because a large change in the state of the system has been predicted, the fidelity selection algorithm is deployed. If the demand predicted by the state estimation algorithm suggests that the system continues to operate under normal conditions, dispatch results from the simulations running at earlier fidelities can be accepted. However, if a significant variation is detected in any of the loads, the fidelity algorithm is employed to select a new simulation fidelity. Here, for each of the different sub-networks, the algorithm determines load variations within the sub-networks using previous dispatch and current estimated loads. Based on these variations, the algorithm matches each of the sub-network variations to the closest corresponding demand level from the fidelities database. Once all of the sub-networks have been matched, the ranking for the corresponding fidelity is used to determine which sub-networks should be included in the simulation.

Two conditions are utilized to evaluate the ranking from the fidelities database. The first condition excludes combinations where 1) multiple sub-networks are included and 2) more than 90% of all of the networks' generation capacity is included. This condition is included to avoid the use of full system simulations which will incur a large computational burden because of the extensive search space for the optimization. The second condition excludes combinations where the total generation capacity of the sub-networks included is

inferior to 10% of the networks' generation capacity. This condition avoids the use of simulations in which diverse solutions meeting the power balance conditions cannot be obtained due to the narrow search space. The solutions that have been avoided by the second condition, not only would have a large computational burden, but also would provide a very limited non-dominated solution set.

3.5. Experiments and Results

3.5.1. Modified IEEE-30 Bus System

In order to demonstrate the validity of the proposed DAS^{EELD} framework in real-time load dispatching problems, a set of experiments are carried out based on a modified IEEE-30 bus system. The original IEEE-30 bus system consists of 30 buses and 41 lines; these buses consist of 6 generation buses, 19 load buses, and 5 buses that neither generate nor request electricity. As mentioned before, the network is divided into 3 sub-networks according to Rakpenthai et al. [32]. To this end, 5 sources of distributed generation are added, arbitrarily located at buses 7, 21, 22, 23 and 27 in the modified IEEE-30 bus system, as shown in Fig. 3. The data regarding the characteristics of this system is obtained from the Power Systems Test Case Archive of the Department of Electrical Engineering at the University of Washington (2012), and the cost data and generation capacities are obtained from [33].

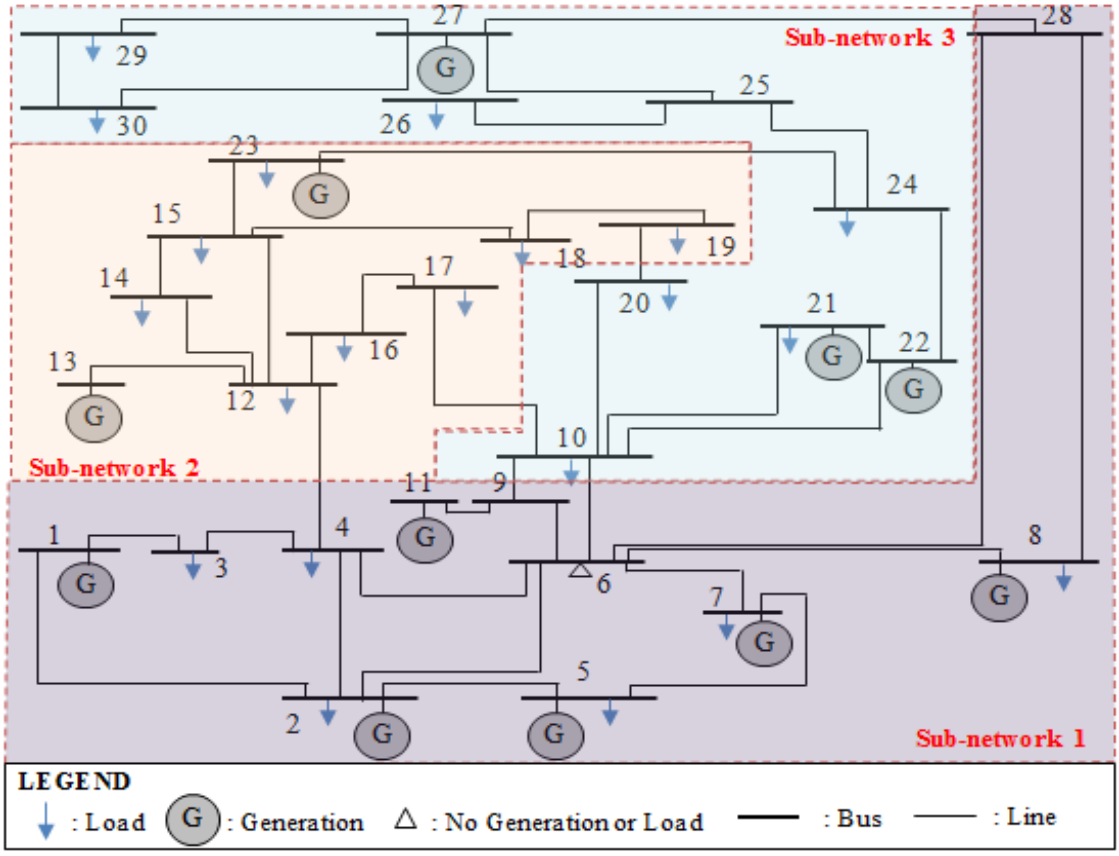


Fig. 3. Modified IEEE-30 bus system with three sub-networks

3.5.2.Discovery Procedure Simulation

The studied network is split into three sub-networks and three levels of load variation (0%, 5%, and 10%) are selected. Therefore, a total of 182 different simulations were carried out, for 26 scenarios with 7 combinations each. The performances of the simulations are shown in Table 1. The combinations within each scenario are ranked based on linear membership functions that give equal weight for both objectives

Table 2 presents the scenarios that correspond to different fidelities and ranks, as well as the probability that a certain scenario is given a certain rank. It shows that, on average, the best performing sub-network simulation fidelities are the combinations that include

only sub-network 2, only sub-network 3 and sub-networks 2 and 3. These three fidelities are ranked in the top three combinations in 52 of the 78 scenarios. Since the probability of their performances belonging to the top three among all the combinations is close to 70%, they are recommended as the simulation fidelity when the demand variations are difficult or impossible to achieve, or in extreme cases where the demand variations do not adjust to any of the predetermined levels of variation, and the burden of a full system simulation may be avoided. Except for these cases, the fidelities database obtained through the discovery procedure, provided in Table 3, is used as a reference, to search for the most suitable fidelities under different demand variation levels.

3.5.3. DAS^{EELD} Evaluation

In order to evaluate DAS^{EELD} framework, the state estimation algorithm has been used to generate 10 different cases where the environmental sensory data has been randomly generated. The fidelities selection database has been used by the DAS^{EELD} framework in the fidelity selection and culling algorithm. For each case, the DAS^{EELD} searches for the closest scenario by comparing the estimated demand changes with the three demand change levels. Once the corresponding scenario is identified, simulations are performed for the two top ranked sub-network combinations. This general rule bears two main exceptions that include the simulation of sub-network combinations with all of the sub-networks and the combination with only sub-network 3.

Table 1: Best compromise solutions from the discovery procedure

Demand Change (%)			Simulated Sub-Networks (All Potential Combinations of Sub-Networks)													
			1		2		3		1,2		1,3		2,3		1,2,3	
1	2	3	Cost	Emission	Cost	Emis.	Cost	Emis.	Cost	Emis.	Cost	Emis.	Cost	Emis.	Cost	Emis.
0	0	5	605.8	0.26730	595.4	0.26730	579.92	0.26733	569.6	0.26736	583.5	0.26732	520.5	0.26744	595.00	0.26732
0	0	10	636.6	0.26725	591.8	0.26731	592.77	0.26730	634.1	0.26726	635.7	0.26726	580.3	0.26733	592.11	0.26732
0	5	0	651.8	0.26724	591.8	0.26731	589.40	0.26731	655.0	0.26723	600.3	0.26729	611.5	0.26727	639.06	0.26726
0	5	5	590.2	0.26730	629.8	0.26724	611.98	0.26727	614.6	0.26728	636.0	0.26723	634.6	0.26723	628.68	0.26724
0	5	10	639.4	0.26723	613.4	0.26727	580.82	0.26733	606.9	0.26727	620.3	0.26725	612.7	0.26727	582.07	0.26733
0	10	0	642.2	0.26724	603.1	0.26728	577.22	0.26733	612.7	0.26728	659.9	0.26723	547.8	0.26739	645.95	0.26726
0	10	5	658.1	0.26723	587.3	0.26731	622.96	0.26725	643.1	0.26723	643.0	0.26724	600.6	0.26729	617.23	0.26726
0	10	10	646.0	0.26723	612.7	0.26727	638.92	0.26722	659.1	0.26722	645.0	0.26721	653.0	0.26719	636.89	0.26724
5	0	0	645.7	0.26722	591.7	0.26731	627.43	0.26724	611.0	0.26726	655.1	0.26722	595.9	0.26730	660.65	0.26722
5	0	5	664.3	0.26720	597.5	0.26729	626.06	0.26724	634.4	0.26724	627.8	0.26724	612.5	0.26727	708.32	0.26718
5	0	10	652.9	0.26720	653.3	0.26719	594.85	0.26730	651.8	0.26721	687.0	0.26718	630.0	0.26724	645.87	0.26722
5	5	0	633.3	0.26724	649.1	0.26720	644.07	0.26721	629.3	0.26724	648.1	0.26720	654.2	0.26719	632.48	0.26724
5	5	5	645.9	0.26721	653.2	0.26719	608.95	0.26727	638.0	0.26721	669.7	0.26717	649.7	0.26720	673.38	0.26719
5	5	10	696.1	0.26717	650.3	0.26720	665.07	0.26717	626.5	0.26725	705.2	0.26716	637.8	0.26722	657.77	0.26722
5	10	0	662.1	0.26719	651.9	0.26720	645.42	0.26721	661.3	0.26719	668.6	0.26719	581.3	0.26732	660.00	0.26718
5	10	5	669.0	0.26718	661.1	0.26718	613.88	0.26726	650.5	0.26719	687.4	0.26717	655.3	0.26719	647.11	0.26720
5	10	10	694.1	0.26715	619.3	0.26725	636.31	0.26722	694.2	0.26716	682.8	0.26716	667.5	0.26717	667.11	0.26718
10	0	0	677.5	0.26717	626.6	0.26724	648.91	0.26720	640.7	0.26721	655.9	0.26719	663.5	0.26718	692.85	0.26719
10	0	5	700.8	0.26715	621.5	0.26725	656.43	0.26719	677.7	0.26718	678.6	0.26717	661.8	0.26718	658.82	0.26718
10	0	10	684.7	0.26715	670.8	0.26716	655.30	0.26719	707.9	0.26715	702.5	0.26712	691.8	0.26713	699.08	0.26714
10	5	0	671.2	0.26717	658.2	0.26719	623.85	0.26725	681.9	0.26718	706.9	0.26715	673.5	0.26716	672.25	0.26717
10	5	5	685.2	0.26714	658.1	0.26719	657.63	0.26719	675.5	0.26717	659.2	0.26718	693.6	0.26712	695.22	0.26714
10	5	10	689.8	0.26714	679.1	0.26715	683.42	0.26714	661.5	0.26718	695.5	0.26714	692.1	0.26713	657.09	0.26718
10	10	0	679.6	0.26714	679.0	0.26715	635.80	0.26723	688.4	0.26714	699.1	0.26713	687.1	0.26714	690.79	0.26715
10	10	5	716.9	0.26710	676.7	0.26715	660.30	0.26718	694.2	0.26714	699.7	0.26711	665.8	0.26717	688.11	0.26713
10	10	10	732.5	0.26710	699.8	0.26711	687.80	0.26713	694.4	0.26712	717.9	0.26710	705.9	0.26710	667.06	0.26716

Table 2: Discovery procedure combination ranking

Combination \ Rank	1	2	3	4	5	6	7
1	1 - 3.8%	2 - 7.7%	2 - 7.7%	2 - 7.7%	5 - 19.2%	11 - 42.3%	3 - 11.5%
2	5 - 19.2%	5 - 19.2%	8 - 30.8%	7 - 26.9%	1 - 3.8%	0 - 0.0%	0 - 0.0%
3	4 - 15.4%	7 - 26.9%	5 - 19.2%	6 - 23.1%	4 - 15.4%	0 - 0.0%	0 - 0.0%
1,2	5 - 19.2%	1 - 3.8%	0 - 0.0%	3 - 11.5%	6 - 23.1%	4 - 15.4%	7 - 26.9%
1,3	5 - 19.2%	2 - 7.7%	1 - 3.8%	1 - 3.8%	4 - 15.4%	8 - 30.8%	5 - 19.2%
2,3	3 - 11.5%	5 - 19.2%	10 - 38.5%	4 - 15.4%	3 - 11.5%	0 - 0.0%	1 - 3.8%
1,2,3	3 - 11.5%	4 - 15.4%	0 - 0.0%	3 - 11.5%	3 - 11.5%	3 - 11.5%	10 - 38.5%

Table 3: Fidelities database

Demand Change (%)			Ranking							Demand Change (%)			Ranking						
1	2	3	1	2	3	4	5	6	7	1	2	3	1	2	3	4	5	6	7
0	0	5	2	1,3	1	3	1,2,3	1,2	2,3	5	5	10	3	2	2,3	1,2	1,2,3	1	1,3
0	0	10	3	2	2,3	1,2,3	1	1,3	1,2	5	10	0	1,2,3	2	3	1,2	2,3	1	1,3
0	5	0	1,3	2,3	3	2	1,2	1	1,2,3	5	10	5	1,2	1,2,3	2	2,3	3	1	1,3
0	5	5	2,3	1	2	1,2,3	3	1,3	1,2	5	10	10	2,3	3	2	1,2,3	1,3	1	1,2
0	5	10	1,2	1,3	2,3	2	3	1	1,2,3	10	0	0	1,3	1,2	3	2,3	2	1	1,2,3
0	10	0	2	3	2,3	1,2	1	1,3	1,2,3	10	0	5	1,2,3	3	2,3	2	1,3	1,2	1
0	10	5	3	1,2,3	2,3	2	1,2	1,3	1	10	0	10	2	3	2,3	1	1,3	1,2,3	1,2
0	10	10	2,3	3	1,3	2	1	1,2,3	1,2	10	5	0	2	3	2,3	1	1,3	1,2,3	1,2
5	0	0	1,2	3	2	2,3	1	1,3	1,2,3	10	5	5	1,3	1	2,3	2	3	1,2	1,2,3
5	0	5	3	2,3	2	1,3	1,2	1	1,2,3	10	5	10	1,2,3	2	2,3	3	1,2	1	1,3
5	0	10	2	2,3	1	3	1,2,3	1,2	1,3	10	10	0	1	2,3	2	3	1,2	1,3	1,2,3
5	5	0	1,3	2	3	2,3	1,2	1	1,2,3	10	10	5	1,3	1,2,3	2	3	2,3	1	1,2
5	5	5	1,2	2,3	2	3	1	1,3	1,2,3	10	10	10	1,2	1,2,3	3	2	2,3	1,3	1

In Table 4, the 7 combinations were run and ranked for each of the 10 cases, in order to further evaluate the effectiveness of the proposed DAS^{EELD} framework. T choice of the two fidelities chosen by the DAS^{EELD} can be benchmarked against this ranking.

Table 4: Fidelity ranking for the experimental cases

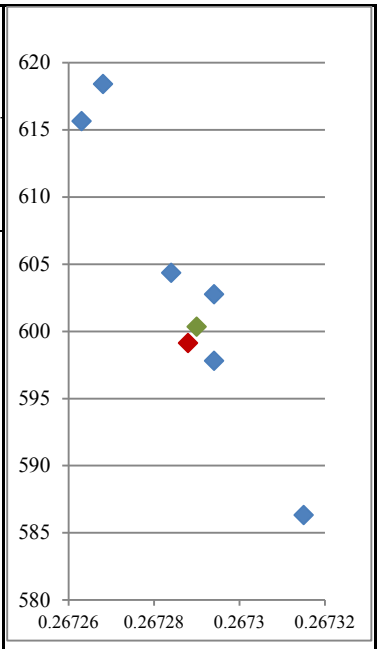
Case	Demand Change (%)			Ranking						
	1	2	3	1	2	3	4	5	6	7
1	0.39	1.50	1.32	1,2,3	2	2,3	1,3	1,2	3	1
2	0.99	0.54	3.77	2	2,3	3	1,2,3	1,3	1	1,2
3	0.32	2.74	2.27	1,2,3	1,3	2,3	1,2	2	3	1
4	3.90	1.69	0.39	1,3	1,2,3	1	2,3	2	3	1,2
5	2.26	0.10	2.60	1,3	1,2	2	1	2,3	3	1,2,3
6	2.55	4.56	6.70	1,3	2,3	2	1	3	1,2,3	1,2
7	5.74	3.48	1.38	1,2,3	2,3	1,3	1,2	3	2	1
8	7.40	0.45	2.28	1,3	2,3	1,2,3	1,2	2	3	1
9	4.50	3.45	12.39	1,2,3	1	2	1,3	3	1,2	2,3
10	10.36	6.24	4.26	2,3	1	2	3	1,2	1,2,3	1,3

Table 5 shows combinations selected by the DAS^{EELD} framework for each of the 10 cases, their rankings, and a comparison between the best sub-network combination obtained through experimental simulation and the suggested fidelity by the DAS^{EELD} for case 3. The table shows that the selected fidelities rank among the top two combinations in five of the cases and among the top three in eight of them. In the embedded figure, blue

dots represent experimental compromise solutions for case 3, red dot is the best solution found from the experimental simulations (i.e., combinations of sub-network 1, 2, and 3), and green dot is the result obtained via the suggested simulation 1 (i.e., combinations of 1 and 2). It is shown that the red and green dots are close to each other, meaning that there is no significant difference between the performances of these two combinations. Therefore, it can be concluded that the proposed DAS^{EELD} is able to provide a good compromise solution without utilizing great computational resources.

Table 5: Proposed sub-network simulation configuration from simulation culling

Case	Demand Change (%)			Discovery Case	Suggested Simulation 1		Suggested Simulation 2	
	1	2	3		Sub-Networks	Rank	Sub-Networks	Rank
1	0.39	1.50	1.32	0,0,0*	1,3	4	2,3	3
2	0.99	0.54	3.77	0,0,5	2	1	1,3	5
3	0.32	2.74	2.27	0,5,0	1,3	2	2,3	3
4	3.90	1.69	0.39	5,0,0	1,2	7	3	6
5	2.26	0.10	2.60	0,0,5	2	4	1,3	1
6	2.55	4.56	6.70	5,5,5	1,2	7	2,3	2
7	5.74	3.48	1.38	5,5,0	1,3	3	2	6
8	7.40	0.45	2.28	5,0,0	1,2	6	3	5
9	4.50	3.45	12.39	5,5,10	3	5	2	3
10	10.36	6.24	4.26	10,5,5	1,3	7	1	2



Chapter 4: Microgrid Automated Control and Optimization using Dynamic Adaptive Simulation DAS^{CONTROL}

A major challenge for microgrids with distributed generation is the technical difficulties related to control of a substantial number of micro-sources. Additionally, the demand for extensive advance in fast sensors and complex control from a central point creates bigger problems. The major concern about a complex control system is that a failure of a single component can bring the system down. Microgrids need to be capable to address events autonomously using only local data and information. In emergency circumstances where faults, blackouts or brownouts have occured, there is a requirement for an immediate change in the output power control of the micro-generators as well as an efficient management and isolation of regions within the microgrid to avoid further escalation of the problem.

To address the autonomous control in microgrids, in this chapter, we present a novel dynamic adaptive simulation framework ($DAS^{CONTROL}$) for the operation and control of MGs that speeds up the real-time computation of the resource allocation and significantly control decisions while optimizing the operational cost, energy surety and emissions. As illustrated in Fig. 4, our framework includes: 1) a database receiving data from electrical and environmental sensors, 2) a fault detection algorithm, 3) an agent-based simulation of the MG system that includes separate modules for energy storage, and energy generation from solar, wind, and diesel sources, 4) an optimal computing budget allocation (OCBA)-based control design selection algorithm, and 5) a multi-objective optimization algorithm for optimizing the decisions of the real MG.

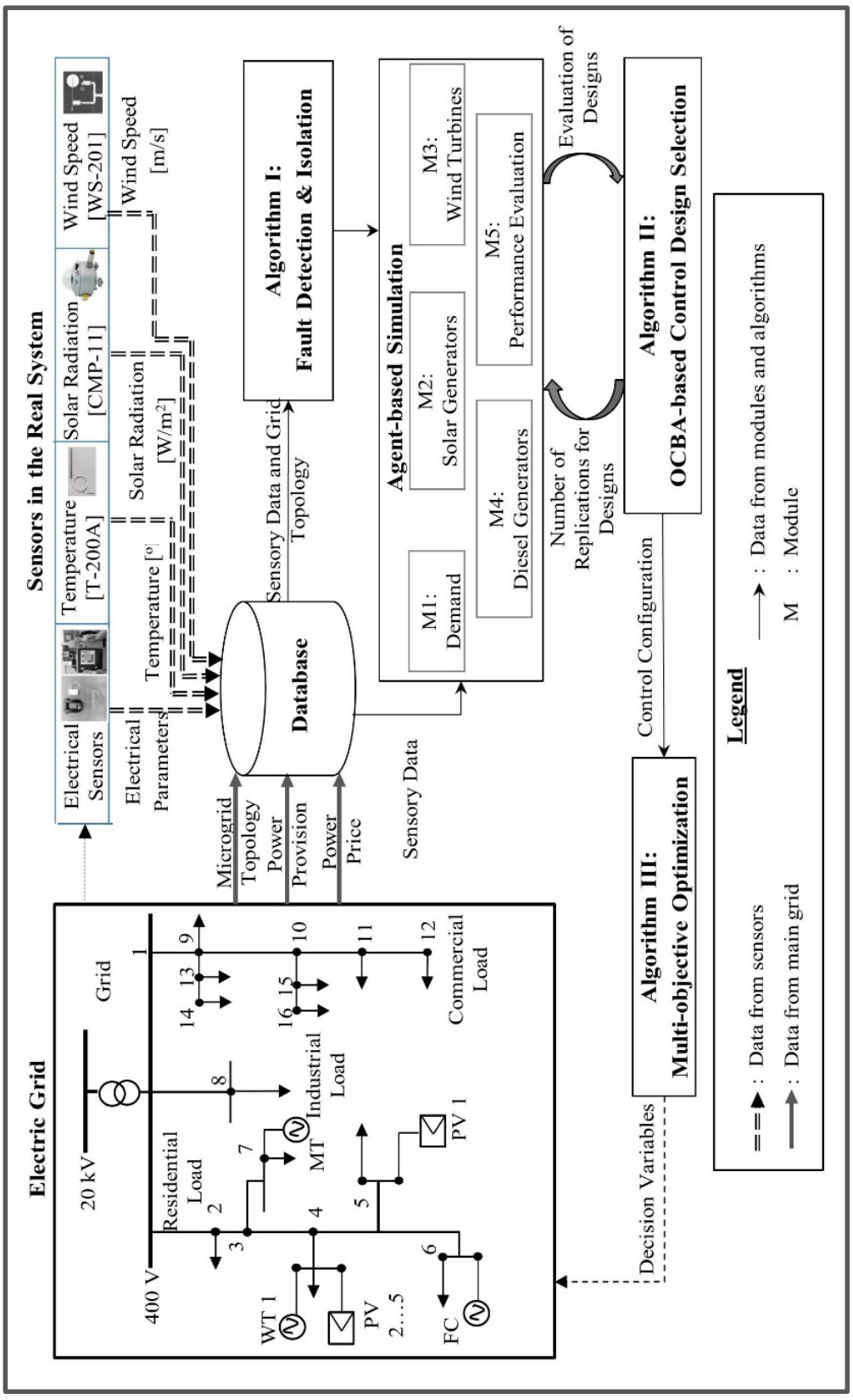


Fig. 4. A DAS framework for automated control in microgrids ($DAS^{CONTROL}$)

A high level description of the framework follows. The real system is equipped with sensors that feed the database of our framework with electrical and environmental data. More specifically, the database is collecting wind speed, solar irradiance, temperature, voltage, current, and frequency data from the real microgrid. The collected data is crucial for the framework since the environmental data is responsible for the forecasting of the energy generation from renewable sources and the electrical data is responsible for the proper operation of the microgrid and the smooth transition from the normal operation mode to the islanded mode. These data are then used by the fault detection and isolation algorithm (FDI) to discover liabilities and potential hazards within the MG. After the FDI algorithm scans the system, its results are provided to the agent-based simulation model. The agent-based simulation of the MG system imitates the real system under several different control designs. The simulation model approaches the MG as a system of systems and each agent simulates a sub-system within the overarching system. The goal of the simulation is to evaluate the performance of different control designs by isolating the most appropriate regions in the system and by finding an optimal resource allocation that minimizes the operational cost while also maximizing the energy surety of the system. To this end, a performance evaluation module is embedded in the model to swiftly evaluate the designs. The simulation model communicates with the OCBA-based control design selection algorithm that makes real-time decisions on which designs need to be simulated and on how many replications. The simulation is used to calculate the performance of various control designs (in an iterative manner) and, subsequently, the OCBA-based control design selection algorithm ranks these control designs based on their calculated

performance. Once the best control design is selected, the problem is solved using the proposed multi-objective optimization algorithm which is described in Section 4.3.

The connection amongst the four major components of our framework is demonstrated in Fig. 5 as the following.

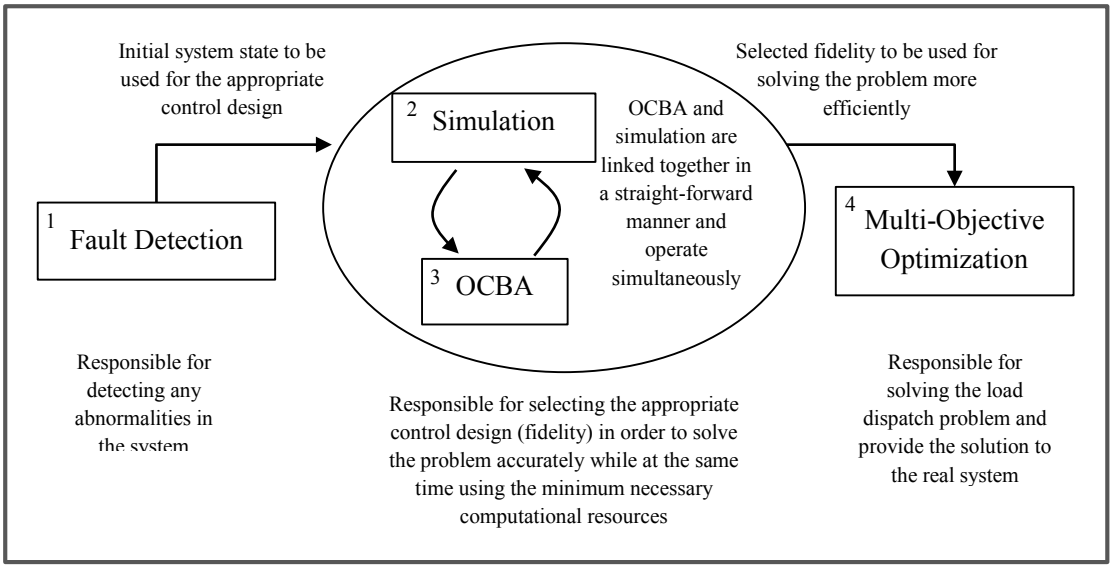


Fig. 5. Connection of the components of DAS^{CONTROL}

4.1. Algorithm I: Fault Detection and Isolation

Fault detection and isolation is a crucial element of numerous operations management automation systems. Fault detection is defined as the identification of a situation that a problem has occurred, even if the origin cause is not yet known. Fault diagnosis, also known as fault isolation, on the other hand, is responsible for locating the origin causes of problems, allowing the system to take the appropriate corrective actions. Automated fault detection and diagnosis prominently relies on the input of sensors or calculated performances of the system [34]. In application systems, the most common faults

encountered are sensor failures, and as a result, a major focus in fault detection and isolation is to identify these problems and address the challenge of distinguishing them from process problems [35].

Fault detection and isolation using equation-based analytical models involves and necessitates a static set of state variables and measurements. In matrix-based techniques particularly, every time that the fault detection and isolation analysis is performed, all state variables are recalculated, updated, and stored. This process does not scale well to large systems where an immense collection of state variables and sensors exists. Other approaches, such as the generalized discriminant analysis (GDA) or the kernel GDA [36], decrease the computational burden by only transmitting noteworthy new information, but even then, every state variable in the system still has to be defined and preserve space for storage.

On the other hand, event-driven systems do not require a predetermined storage size to represent the system's state. A specific event might occur several times, and events that do not occur necessitate no storage. Events contain some attributes including but not limited to an event category, an associated object that might reveal the root cause of the problem, and a time stamp that is tied with the exact time that the problem is observed. When the system is monitored in the highest fidelity, recent events are often stored in the RAM (rapid access memory) for brisk retrieval.

Table 6: List of events in fault detection and isolation

Measured Variable	Event	Possible Fault
Voltage	Increased Voltage	Load-shedding
	Decreased Voltage	Short-circuit/Motor Start-up
	Neutral Voltage	Ground-fault
	Negative Voltage	Sensor Fault
Current	Overcurrent	Overload/Short Circuit
	Differential Current	Short Circuit
	Negative Current	Sensor Fault
Impedance	Low Impedance	Short-circuit
	Abnormal Ratio X/R	Short-circuit/ Sensor Fault
Frequency	Low Frequency	Increased Load/Sensor Fault
	High Frequency	Loss of Load/Sensor Fault
Phase Angle	Phase Angle Change	Short-circuit/Sensor Fault
Temperature	Increased Temperature	Overload
	Decreased Temperature	Short-circuit
	Negative Temperature	Sensor Fault
Power	Active Power in Zero-Sequence Component	Short-circuit/Sensor Fault
	Change of Direction of Power Flow	Ground-fault/Sensor Fault
Speed	Change of Oil Flow in Transformer	Overload/Short-circuit
Solar Irradiance	Abnormal Solar Irradiance	Sensor Fault
Wind Speed	Abnormal Wind Speed	Sensor Fault

The possible events that may be reported in our model are listed in Table 6. When a measured variable gets a suspicious value then the corresponding event is created and a possible fault is associated with it. If there is a sensor error, the system stops acquiring data from this source and, concurrently, increases the details of the microgrid control design.

4.2. Algorithm II: OCBA-based Control Design Selection

This algorithm addresses the design selection problem from a finite set of control design alternatives with stochastic constraints in MG simulations for the allocation of the simulation replications. The real microgrid has several segregation points which are used to isolate specific areas or even individual buildings. Our framework automatically controls sets of these points and isolates regions accordingly when needed. For example, as shown in Fig. 6 for a group of 22 buildings, we may have a design where the buildings are partitioned into two subgroups, each controlled by a segregation point (Design A), or 4 subgroups and segregation points (Design B) or 22 subgroups and segregation points (Design C), etc. These different control designs are predetermined to fit the real system that is simulated. The goal of this algorithm is to select the most appropriate design and provide it to the multi-objective optimization algorithm as an input. Using this design, the multi-objective optimization algorithm provides a Pareto solution while saving significantly from the computational time.

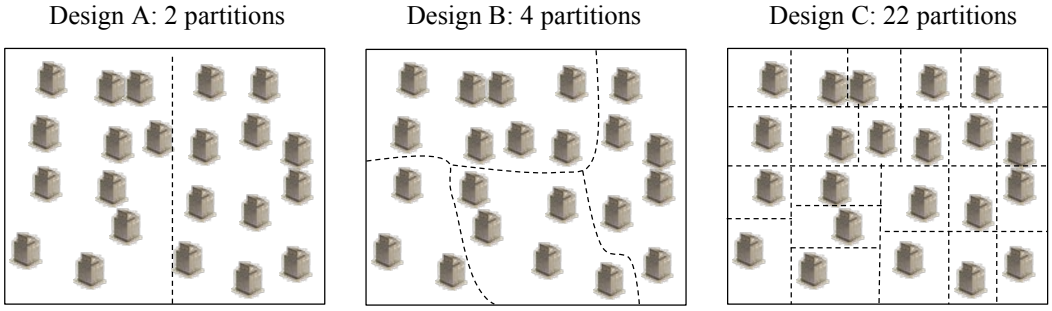


Fig. 6. Different designs for a group of buildings

Three demand categories, namely critical, priority, and non-critical are used to prioritize buildings within the microgrid. Critical demand areas are comprised of required areas for the administration and community control; priority demand areas cover the

buildings in which typical day-to-day functions take place; and non-critical demand buildings include the recreation, housing, and shopping facilities. There are also three different generation sources within the considered microgrid: diesel generators, photovoltaic arrays, and wind turbines.

For the proposed OCBA-based control design selection algorithm, the simulation performance of the system is formulated as a single maximization objective function as shown in eq. (17). This single objective function combines the average percentage of satisfaction for different load types and total operational cost of the MG during the simulation time.

$$f(X) = \alpha \cdot P_{crit} + \beta \cdot P_{pr} + \gamma \cdot P_{ncrit} + \delta \cdot \frac{c_{max} - c}{c_{max}} \tag{17}$$

where X is the design for which the performance is evaluated, P_{crit}, P_{pr}, and P_{ncrit} are the percentages of energy surety of the critical, priority and non-critical loads respectively, c_{max} is the maximum cost calculated so far between all replications and designs, c is the cost of the current replication for design X and α, β, γ, and δ are the coefficients of the percentages of energy surety/cost and correspond to the priority that is given to the different objectives.

We assign simulation replications to design alternatives utilizing the aforementioned performance function and the OCBA technique [37]-[41]. As the simulation proceeds, we compute the mean and variance of the MG's performance using the data which are collected up to that stage. The algorithm is presented in Fig. 7.

[Step 1]: Input data

(1) Define k (number of alternative designs), N (required probability of correct selection), Δ (available budget for one iteration of the algorithm), n_0 (initial replications for each design), and l (current iteration in OCBA)

(2) Set $l \leftarrow 0$;

(3) Set $n_1^l = n_2^l = \cdots = n_k^l = n_0$

While ($P\{CS\} <$ N) repeat Steps [2]-[5]

[Step 2]: Performance calculation

(1) Calculate the mean of MG's performance for each design:

$$\bar{f}_i = \frac{1}{n_i^l} \sum_{j=1}^{n_i^l} f_{ij} , \qquad i = 1,2, \dots, k$$

(2) Calculate the corresponding standard deviations:

$$s_{f_i} = \sqrt{\sum_{j=1}^{n_i^l} (f_{ij} - \bar{f}_i)^2 \Big/ n_i^l - 1} , \qquad i = 1,2, \dots, k$$

[Step 3]: Replication Calculation

Calculate the new replications for MG designs (n_1^{l+1}, n_2^{l+1}, … , n_k^{l+1}) using the following equations and rounding them to the nearest integer (b represents design with best performance):

$$\frac{n_i^{l+1}}{n_j^{l+1}} = \left(s_{f_i} (\bar{f}_b - \bar{f}_j) / s_{f_j} (\bar{f}_b - \bar{f}_i) \right)^2 , \text{for } i > j, \qquad i, j \neq b$$

$$n_b^{l+1} = s_{f_b} \sqrt{\sum_{i=1, i \neq b}^{k} \left(n_i^{l+1} / s_{f_i} \right)^2}$$

[Step 4]: Simulation

(1) Perform $\left[\max\left(n_i^{l+1} - n_i^l , 0\right)\right]$ replications for all design i

(2) Set $l \leftarrow l + 1$.

[Step 5]: Probability of Correct Selection Calculation

(1) Calculation of the approximate probability of correct selection using the Bonferroni inequality ($APCS - B$):

$$APCS - B \equiv 1 - \sum_{i=1, i \neq b}^{k} P\{f_b < f_i\}$$

(2) Calculation of the approximate probability of correct selection in a product form:

$$APCS - P \equiv \prod_{i=1, i \neq b}^{k} P\{f_b > f_i\}$$

(3) Both $APCS - B$ and $APCS - P$ result in the approximate lower bound of $P\{CS\}$. Therefore,

$$P\{CS\} = \max(APCS - B , APCS - P)$$

*In order to obtain $P\{f_b < f_i\}$ pairwise Student's t-test is used

Fig. 7. Operations of the isolated MG control design algorithm

In the OCBA-based control design selection algorithm, the best design (b) may change in each iteration (l). As the simulation proceeds by increasing the number of iterations, the best design in most cases converges to the optimal design as l goes to infinity (the probability of correct selection approaches very close to 1). In order to conduct OCBA appropriately, the initial number of replications (n_0), and one-time increment (Δ) should not be too small to avoid a poor estimation of the mean and variance.

4.3. Algorithm III: Multi-objective Optimization

The multi-objective optimization algorithm aims to determine the decision variables of the MG real system with accuracy. Thus, given a set of segregation points, demand in each building, and renewable generation, the multi-objective optimization algorithm defines the region that will be disconnected as well as the generation sources that will be initiated to satisfy the demand. The goal of this algorithm is to maximize energy surety while at the same time minimize the total operational cost. Our model also considers the environmental factors by minimizing the emissions of carbon oxides (CO_2), oxides of nitrogen (NO_x), and sulfur oxides (SO_2). The notation of the problem is provided in Table 7.

Table 7: Summary of Notation and Formulation of the Problem

Parameters	Explanation
d_i	Demand in building i
p_W	Production from wind turbines
p_S	Production from photovoltaic solar panels
n	Number of buildings in MG
m	Number of segregation points
l_j	Number of buildings controlled by point j
s_j	Indices of buildings associated with point j
k	Number of diesel generators
P_z^{max}	Maximum operating power of generator z
o_z	Operation and maintenance cost of generator z
a_y	Cost of emission type y
e_{zy}	Emission factor of generator z for type y
f_c	Fuel cost per liter
t_1	Weight of critical demand
t_2	Weight of priority demand
t_3	Weight of non-critical demand
u_z, v_z, w_z	Parameters of diesel generator z
cr_i	$\begin{cases} 1, & \text{if bulding i is critical} \\ 0, & \text{otherwise} \end{cases}$
pr_i	$\begin{cases} 1, & \text{if bulding i is priority} \\ 0, & \text{otherwise} \end{cases}$
Binary variables	
x_i	$\begin{cases} 1, & \text{if bulding i is connected} \\ 0, & \text{otherwise} \end{cases}$
g_j	$\begin{cases} 1, & \text{if switch in point j is on} \\ 0, & \text{otherwise} \end{cases}$
Continuous variables	
P_z	Production from diesel generator z

The mathematical formulation of the objectives of the problem is given in eqs. (18)-(20). Z_1 represents the operational cost of the microgrid, Z_2 the emissions and Z_3 the energy surety.

$$Min\ Z_1 = \sum_{z=1}^{k} f_c(u_z + v_z P_z + w_z P_z^2) \quad (18)$$

$$Min\ Z_2 = \sum_{z=1}^{k} \sum_{y=1}^{3} a_y e_{zy} P_z \quad (19)$$

$$Max\ Z_3 = \sum_{i=1}^{n} t_1 cr_i x_i d_i + t_2\, pr_i x_i d_i + t_3(1 - cr_i - pr_i) x_i d_i \qquad (20)$$

The constraints of the problem are shown in eqs. (21)-(25).

$$p_W + p_S + \sum_{z=1}^{k} P_z = \sum_{i=1}^{n} d_i x_i \qquad (21)$$

$$P_z^{min} < P_z < P_z^{max} \qquad \forall z = 1, \dots, k \qquad (22)$$

$$g_j = \frac{\sum_{i \in s_j} x_i}{l_j} \qquad \forall j = 1, \dots, m \qquad (23)$$

$$x_i \in \{0,1\} \qquad \forall\, i = 1, \dots, n \qquad (24)$$

$$g_j \in \{0,1\} \qquad \forall\, j = 1, \dots, m \qquad (25)$$

In our framework, each segregation point is associated with a specific set of buildings depending on the topology of the microgrid. When the fidelity level of the control design becomes higher, so does the number of segregation points. Therefore, the number of buildings that is associated with each point becomes smaller. In particular, in the highest possible fidelity, each segregation point controls a single building. As each segregation point is represented in the problem's mathematical formulation with a binary variable, when the fidelity level is higher, the multi-objective optimization algorithm has to deal with a much more complex problem. This is because the solution space of the problem is exponential to the number of the segregation points. Hence, if there are N segregation points, the solution space contains O(2^N) different solutions. As illustrated in Fig. 8, if a small group of 3 buildings is controlled by a single segregation point, then the solution space contains only 2 potential solutions. However, if each building is controlled by a unique segregation point, then the solution space contains 8 potential solutions.

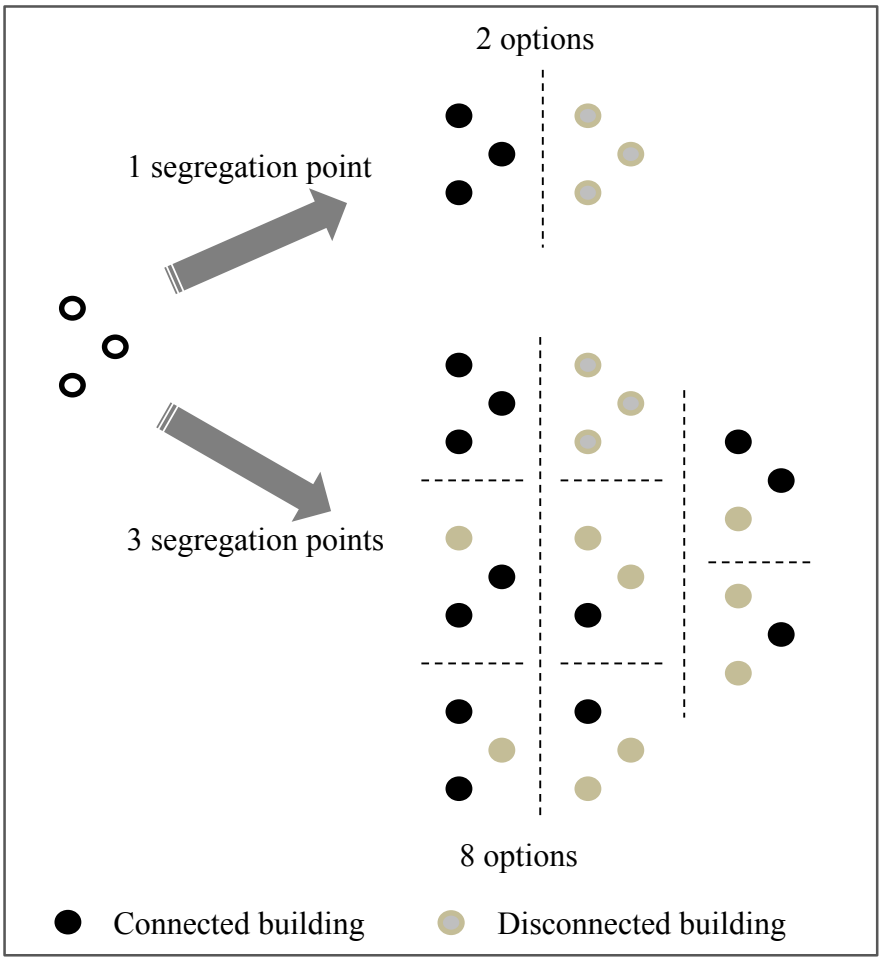


Fig. 8. Segregation points and solution space complexity

It must be noted here that when the control of the isolation mechanism within the MG becomes less intensive, the algorithm may end up with decisions that are sub-optimal sacrificing from energy surety, cost, and emissions. To this end, it is crucial to effectively select the fidelity level for the control mechanism in order to speed up the computational time while also producing near-optimal decisions for the system.

Our multi-objective optimization algorithm utilizes the ϵ-constraint method which has multiple advantages over the weighting method in cases where there exist multiple objectives:

- It can produce unsupported efficient solutions for multi-objective integer programming problems.
- There is no need to scale the objective functions in order to obtain a weighted sum.
- The total number of efficient solutions can be controlled by adjusting the number of grid points [42]-[44].

The appropriate ϵ for each constraint can be acquired by a parametrical variation of the right-hand-side part of each constraint. In order to prevent redundant or dominated solutions, a range for each objective is calculated using the payoff table. The payoff table is created using the results from the individual optimization of each objective function using the other objectives as constraints. After the calculation of the payoff table, the ranges of the values of the objectives can be divided in an arbitrary number of intervals.

4.4. Agent Based Simulation

One of the most promising ways to simulate the behavior of a microgrid is using an agent-based simulation model. Agent-based simulation is a pioneering technique for modeling systems that contains several autonomous components that interact with one another. Agent-based modeling is a novel approach to the simulation of power network systems due to the multiple independent agents (power generation resources) that interact with and feed into a larger network that can realistically represent the structure of an electricity distribution system. This larger network learns from their ever-increasing data-collection and adapts the overall system behavior to better suit the needs of its customers.

In our case, when creating the microgrid simulation model, agents are designed for the loads, wind turbines, photovoltaic arrays, and diesel generators. Our simulation model

is built using a Java based software that brings together system dynamics, discrete event and agent based methods within a single model development environment. Figure 9 illustrates the main class of the microgrid simulation model in the aforementioned software.

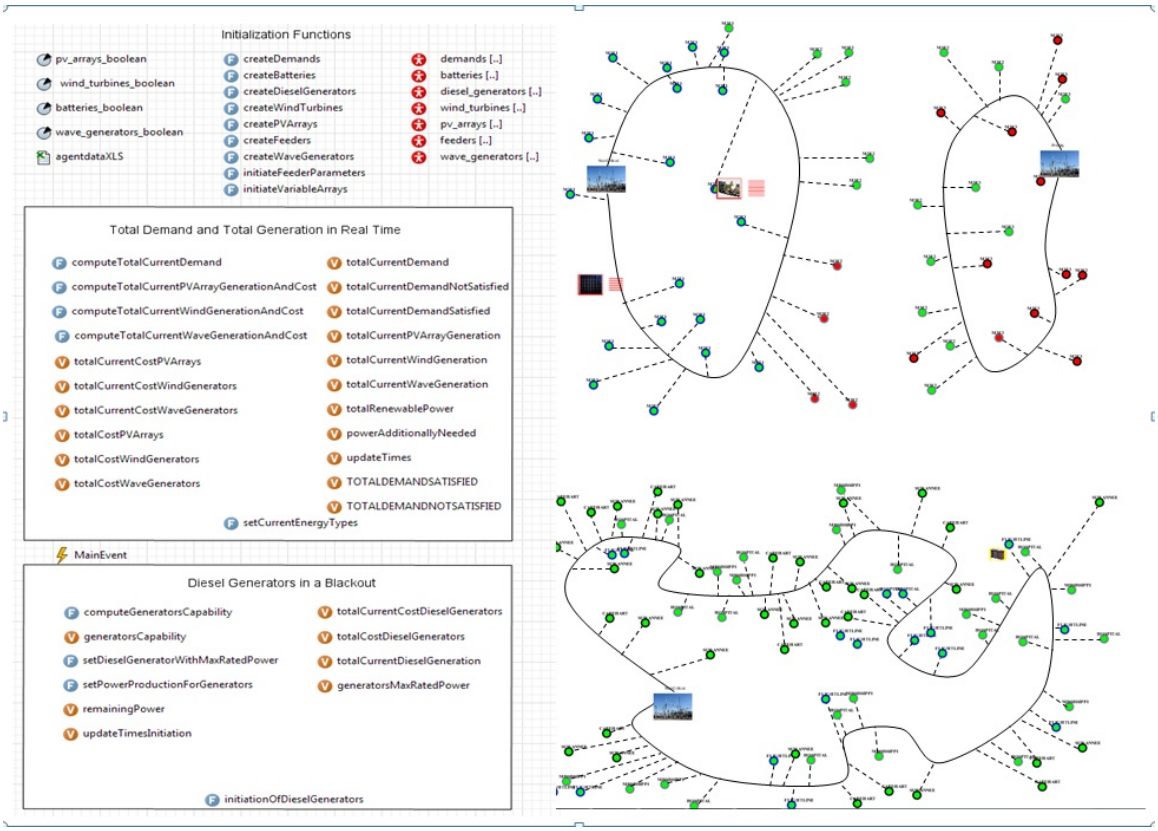


Fig. 9. Main class of agent based simulation model of a microgrid

Details of the major agents are described in the following sections.

4.4.1.Demand Agent

For each building on the microgrid area, a demand agent is created. The agent reads its parameters from the database's corresponding tables. To this end, it reads the type of demand (critical, priority, non-critical), the type of load (residential, commercial, or

industrial), the peak demand of the building, as well as the transformer that feeds the building. By using a power factor plot for the load, the current demand is calculated per second. More specifically, following the values of the power factor in the schedule, the current demand is calculated by also adding a Gaussian noise. This agent contains an event that is triggered each second and calculates the new demand and total power.

4.4.2.Solar Agent

A photovoltaic cell (PV) transforms the solar irradiance into electricity. In our simulation model, the solar generator agent is designed to simulate each solar farm in the system each consisting of several PV cells. To this end, the agent utilizes the area and efficiency of each cell along with the total number of cells. It then uses the embedded functions to calculate the power output. Each second, the simulation triggers an event and calculates the power output for the solar farm using the solar radiation measurements. Equation (26) allows for the calculation of the power output from PV system with an area $A\ (m^2)$ when the total solar irradiance of $Ir\left(\frac{kWh}{m^2}\right)$ is incident on the PV surface [45],[46]:

$$P = Ir \cdot \eta \cdot A \tag{26}$$

where η denoted the system efficiency and η_m the module efficiency and are defined in eqs. (27) and (28).

$$\eta = \eta_m \cdot \eta_{pc} \cdot P_f \tag{27}$$

$$\eta_m = \eta_r \cdot \left(1 - \beta(T_c - T_r)\right) \tag{28}$$

where η_r is the manufacturer's reported efficiency, η_{pc} is the power conditioning efficiency, P_f is the packing factor, β is the array temperature coefficient, T_r is the reference temperature for the cell efficiency, and T_c is the current temperature. Figure 10 is the state chart for the solar agent used in this simulation.

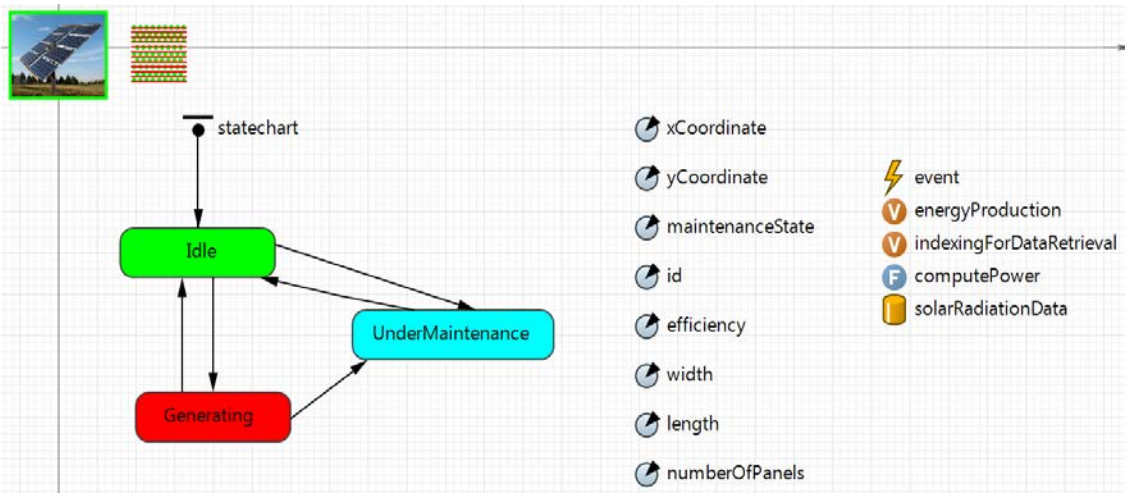


Fig. 10. State chart and considered parameters in the solar generator agent

4.4.3.Wind Generator Agent

Wind power is increasing in popularity primarily due to the fact that it does not produce greenhouse gas emissions while it uses a relatively insignificant amount of space. Wind turbines are responsible for transforming wind energy into electricity. In the proposed simulation model, a wind generator agent is designed to compute the power output of the wind farms over short periods of time (on the order of seconds). The agent takes into consideration the real time measurements of the wind speed, the number of wind

turbines, their efficiency, and their rated power output to calculate the overall power output. Equation (29) is used for the actual power output of a wind turbine [46]:

$$P = P_W \cdot A_W \cdot \eta \tag{29}$$

where P_W is the power output, A_W is the total swept area, and η is the efficiency of the wind turbine generator.

In the simulation model, for each second, the simulation triggers an event and calculates the power output of the wind farm using the wind speed measurements and the given parameters of the wind turbines.

4.4.4.Diesel Generator Agent

Diesel backup generators offer an uninterruptible source of electricity when all other resources of electricity fail. In our simulation model, the diesel generator agent acts as a stand-by system that activates automatically whenever the system detects a lack of power. When a power loss occurs, the microgrid controller signals the generator or generators needed depending on the estimated amount of electricity needed. When power is restored or renewables are again able to meet the demand, the controller automatically transfers the electrical load back to the utility or the renewables respectively. The microgrid then sends a signal to the generators to power down and to return to standby mode to wait for the next power outage.

4.5. Validation of the Individual Components of DAS^{CONTROL}

In order to validate the performance of the proposed framework, we first test the performance of its crucial components in isolation (before they are all embedded into our proposed decision framework). To this end, we first validate the OCBA-based control design selection algorithm by testing the total number of replications needed from the simulation to reach the desired probability of correct selection of the best design in the considered case study microgrid. We compare the total number of replications acquired by the proposed algorithm against those obtained from traditional approaches, including equal allocation (EA) and proportional to variance (PTV) algorithms, with 3 different predetermined control design sets. The first, second, and third control design sets includes 5, 25, and 125 different designs (i.e., fidelities), respectively. In EA, the replications of the simulation are equally allocated among all of the design candidates [37]. PTV can be regarded as a sequential version of the two-stage procedure of Rinott [47] where the computing budget is allocated proportionally to the estimated sample variances. The details of the experiments and the results obtained from the proposed approach in comparison with those obtained from the traditional approaches are provided in Figs. 11-13.

MG structure with 5 fidelities: The results of this experiment show that the OCBA-based control design selection algorithm reaches a probability of correct selection of 98.5% after 35 replications while the PTV and EA algorithms need 55 and 65 replications to reach the same P{CS} (see Fig. 11). Therefore, the OCBA-based Control Design Selection Algorithm performs approximately 60% and 85% better than PTV and EA algorithms respectively.

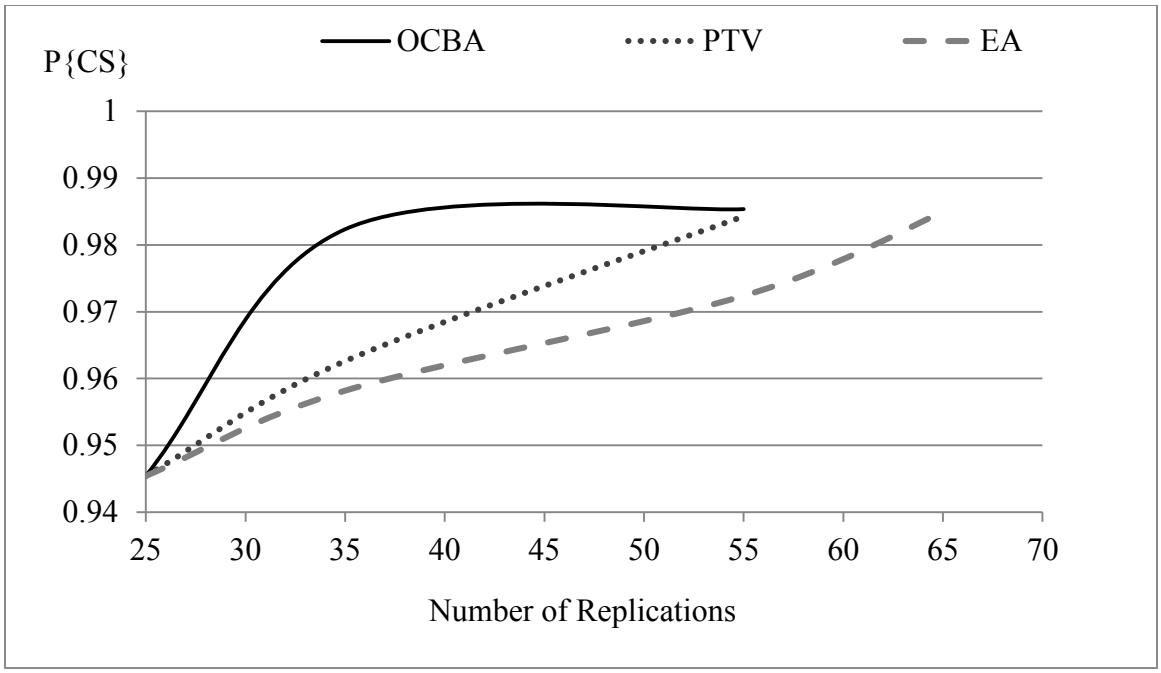


Fig. 11. Comparison of OCBA, PTV and EA in a 5-fidelity microgrid

MG structure with 25 fidelities: The results shown in Fig. 12 reveal that 500 and 750 replications of the OCBA-based control design selection algorithm lead to a 97.9% and 99.8% probability of correct selection, respectively. On the other hand, using PTV and EA algorithms results in the same P{CS} after 2250 and 2500 replications. These results illustrate that using the proposed algorithm is approximately four times faster than PTV and EA.

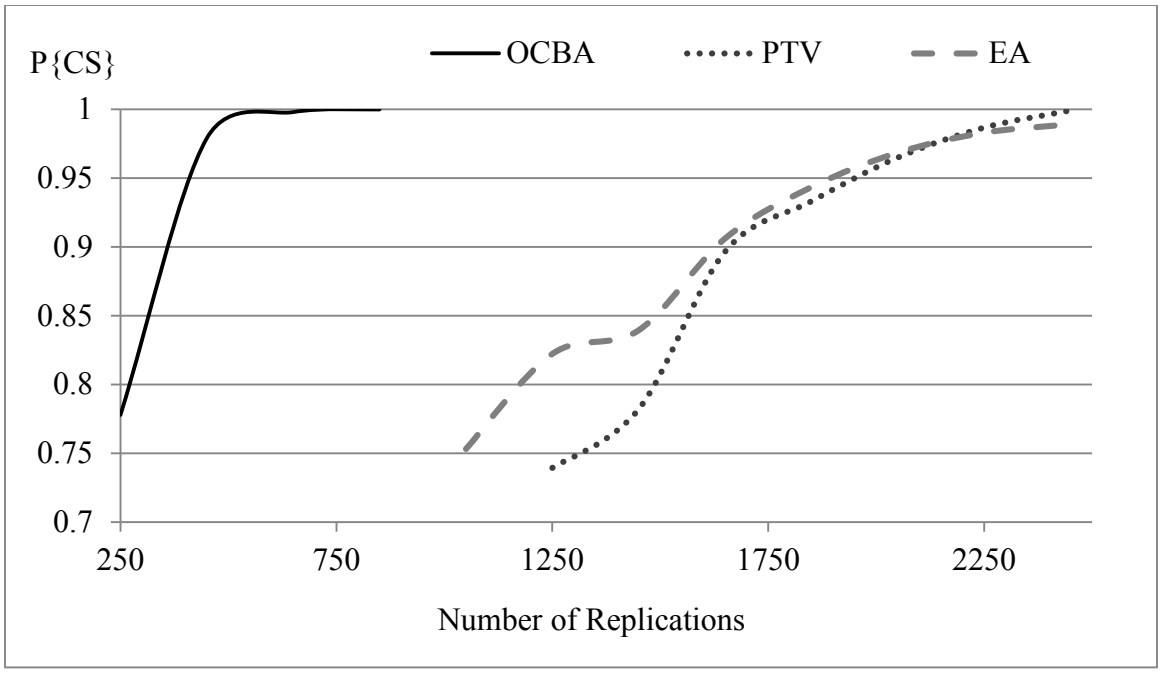


Fig. 12. Comparison of OCBA, PTV and EA in a 25-fidelity microgrid

MG structure with 125 fidelities: Figure 13 shows that the OCBA-based control design selection algorithm reaches the 94.5% and 98.45% probability of correct selection after 4500 and 6500 replications respectively. Conversely, using EA and PTV algorithms only resulted in 93% probability of correct selection after 36500, and 40500 replications, respectively.

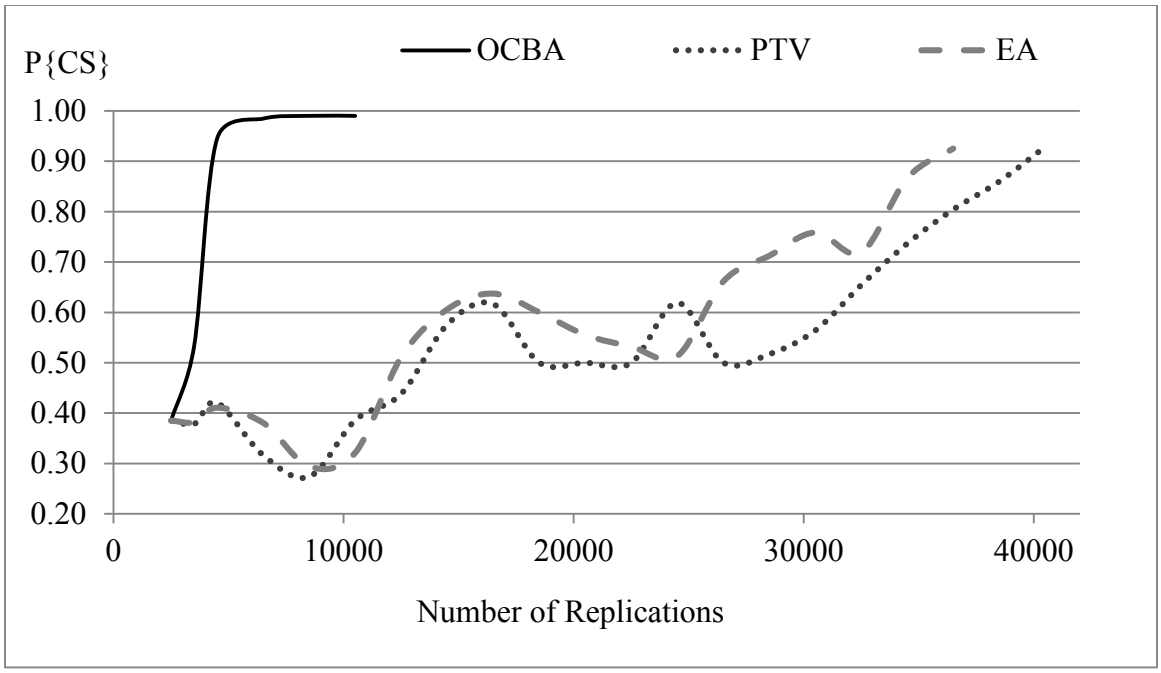


Fig. 13. Comparison of OCBA, PTV and EA in a 125-fidelity microgrid

To validate the effectiveness of the proposed multi-objective optimization algorithm, we compare the quality of the solutions by solving the problem of (2)-(9) within the considered microgrid when the control fidelity is 3 and 5, with 25 and 186 segregation points, respectively.

The objectives considered in the multi-objective optimization algorithm are to maximize the total energy surety by giving priority to the critical and priority loads, minimize the operational cost, and minimize the amount of emissions by allocating the generation resources and controlling the segregation points. All solutions were obtained using GAMS (General Algebraic Modeling System) with SCIP solver version 2.1.1. The experiments were conducted on a computer with 3.40 GHz processor and 8.00 GB memory (RAM).

In order to use the ϵ-constraint method in the optimization problem described in

Section III.C., the optimization problem was first solved for each objective function separately by considering the rest of the objective functions as constraints. The results obtained from that procedure led to a minimum and maximum value for each objective function. These maximum and minimum values were then used for the creation of 10 intervals for both the cost and emissions objective functions. The minimum and maximum of every interval represents values of ϵ_i for the objective functions. In the last step, the objective function with the highest priority (i.e. energy surety objective) was considered as the only objective function of the problem, and the cost/emission objectives were added as constraints allowing them to take values only within the intervals described above. Since for both the cost and emissions objectives 10 intervals were created, their combination resulted in 100 different optimization problems and, respectively, in 100 distinct Pareto solutions. However, some of the obtained solutions were non-feasible while some others were not in the Pareto frontier. In Fig. 14 we illustrate the obtained Pareto frontier solutions solving the problem in fidelity 5 (left) and in fidelity 3 (right), respectively.

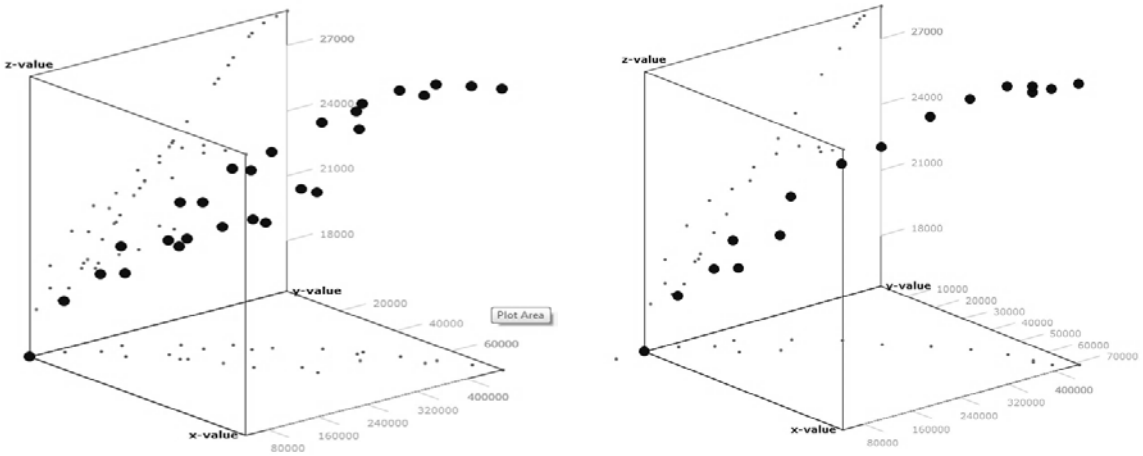


Fig. 14. Set of Pareto solutions for fidelity 5 (left) and fidelity 3 (right).

The large (and bold) and small dots represent the Pareto solutions, and their projections on the axis-planes, respectively. The x-, y-, and z-axis represent the operational cost of the microgrid, the emissions, and the amount of satisfied demand, respectively.

The challenge in comparing the solutions here is that the solutions in each case correspond to a different optimization model, while both models are using the same input data and are optimizing the same objectives. While the highest fidelity model includes substantially more binary variables and hence, acquires more solutions in the Pareto frontier, considering the fact that we have to choose only one solution (for the microgrid operation and control) and that the energy surety is the most important objective, we should focus on the solutions that maximize the z-axis (energy surety). We can see that in both cases of fidelity 3 and 5, we acquire several Pareto solutions, but when the control of the segregation points is limited, we acquire less Pareto solutions. This is expected since in the second case the number of the binary decision variables is significantly limited. However, for the Pareto solutions that are acquired in the space region where the most important objective is optimized (z-axis for energy surety objective), in both cases, the ϵ-constraint method leads to the same efficiency.

4.6. Validation of DAS$^{\text{CONTROL}}$

Our proposed framework is applied to the considered microgrid for a period of 24 hours starting at 12 am for validation purposes. The microgrid under investigation suffers from a blackout and is isolated using only its own generation resources to satisfy the demand.

The main goal of the framework is to significantly speed up the computational time of solving the aforementioned problem. To this end, the simulation model is used for the selection of the most appropriate control/design mechanism, and then, the problem is solved with the multi-objective optimization algorithm of the framework using the best selected control design as an input.

In order to validate the framework, we compare the computational time needed for solving the problem using the proposed framework to the computational time needed for solving the problem as a plain multi-objective optimization problem without considering any smart control design selection that would reduce the complexity. In Table 8, the time in seconds is shown for both of these considered cases. Even though the proposed framework utilizes the simulation and the OCBA-based control design selection algorithm, the total time of solving the framework is significantly reduced. This occurs since with the proposed framework the complexity of the multi-objective optimization problem substantially drops. As discussed in Section II.C, the solution space is exponential to the number of segregation points. To this end, with the proposed framework the number of segregation points is much smaller, and thus, the Multi-objective Optimization Algorithm solves the problem much faster.

Table 8: Computational time comparison for the framework validation

Hour	Time (s) (Plain Optimization)	Time (s) (Framework)		
		Simulation& OCBA	Optimization	Total
1	105	13	1	105
2	118	15	2	118
3	80	13	3	80
4	82	12	4	82
5	124	12	5	124
6	81	14	6	81
7	81	12	7	81
8	122	13	8	122
9	119	13	9	119
10	76	15	10	76
11	136	16	11	136
12	75	13	12	75
13	76	13	13	76
14	124	14	14	124
15	119	15	15	119
16	78	16	16	78
17	77	16	17	77
18	77	14	18	77
19	83	14	19	83
20	76	12	20	76
21	77	12	21	77
22	73	16	22	73
23	75	13	23	75

The total computational time needed to solve the problem using the proposed framework in comparison to solve it as a plain multi-objective optimization problem is shown in Fig. 15. Here, the dashed black line illustrates the amount of time it takes to solve the problem as a plain optimization problem. This time also corresponds to the computational time of solving the considered problem using our framework in the maximum possible control fidelity. The solid black line is the computational time it takes to solve the problem using our proposed DAS$^{\text{CONTROL}}$ framework, and this time includes the

time required for the simulation to acquire the best possible control design.

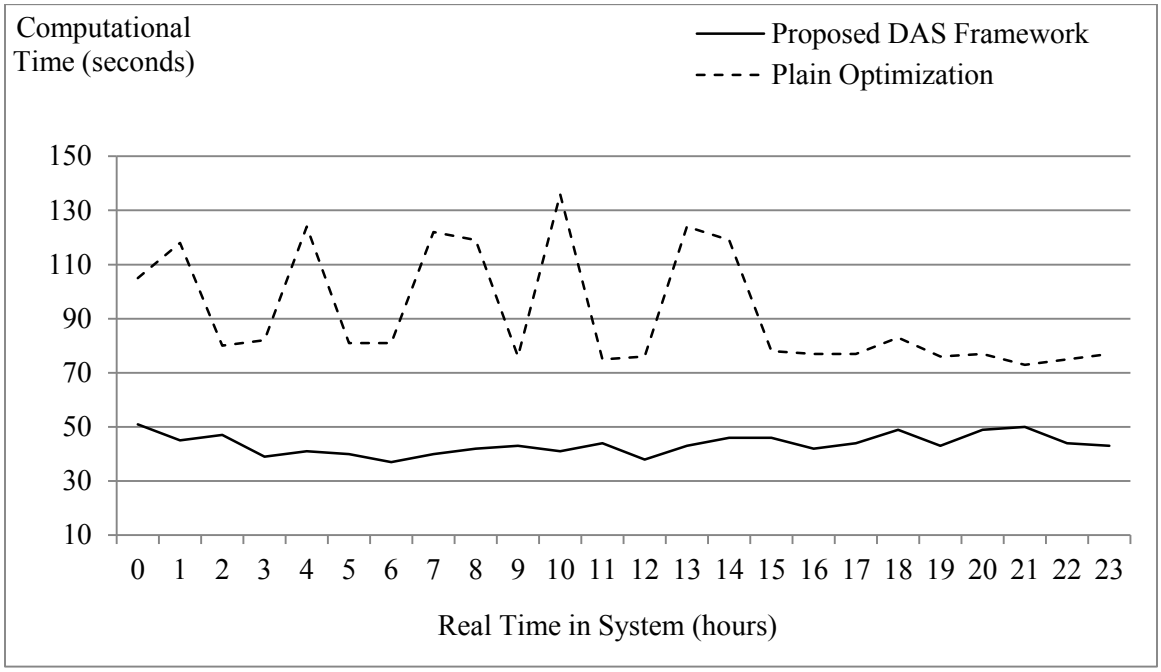


Fig. 15. Comparison of computational time for solving the problem

The time required to solve the considered problem using our proposed framework is significantly less than that of solving the problem as a plain optimization problem. The 95% confidence interval reduction in the computational time is 50.38% $\pm$ 11.09%; where in some cases even a reduction of up to 70% is observed.

Chapter 5: Self-healing of Distributed Microgrids using Dynamic Adaptive Simulations (DAS^{SH})

During severe weather and other disasters, some damages to physical infrastructure is unavoidable but a microgrid with the ability to respond to faults and isolate regions could alleviate the impact of the fault and improve the speed of recovery. Moving toward a smarter, self-healing power grid serves to improve the reliability of an electrical network by reducing the frequency and duration that a power event or outage affects its customers. In order to be able to mitigate the impact of such power events, self-healing power grids must be able to dynamically diagnose and assess the status of each sub-grid within the overall topology in a timely and efficient manner. Such a timely status assessment and system control could be facilitated through the selection of the best control strategy that enables the communication between the different segments of the considered smart grid. This is necessary so that, when a fault occurs in one portion of the grid, that area can be isolated to break the cascading events' series and reallocate resources to the affected area in order to minimize its negative impacts on its customers. Smarter grid technologies use several interaction strategies to achieve the needs of their specific function, namely, isolated self-healing strategy, centralized self-healing strategy, and cooperative self-healing strategy. Figure 16 demonstrates the self-healing microgrid environment utilizing different system interaction strategies.

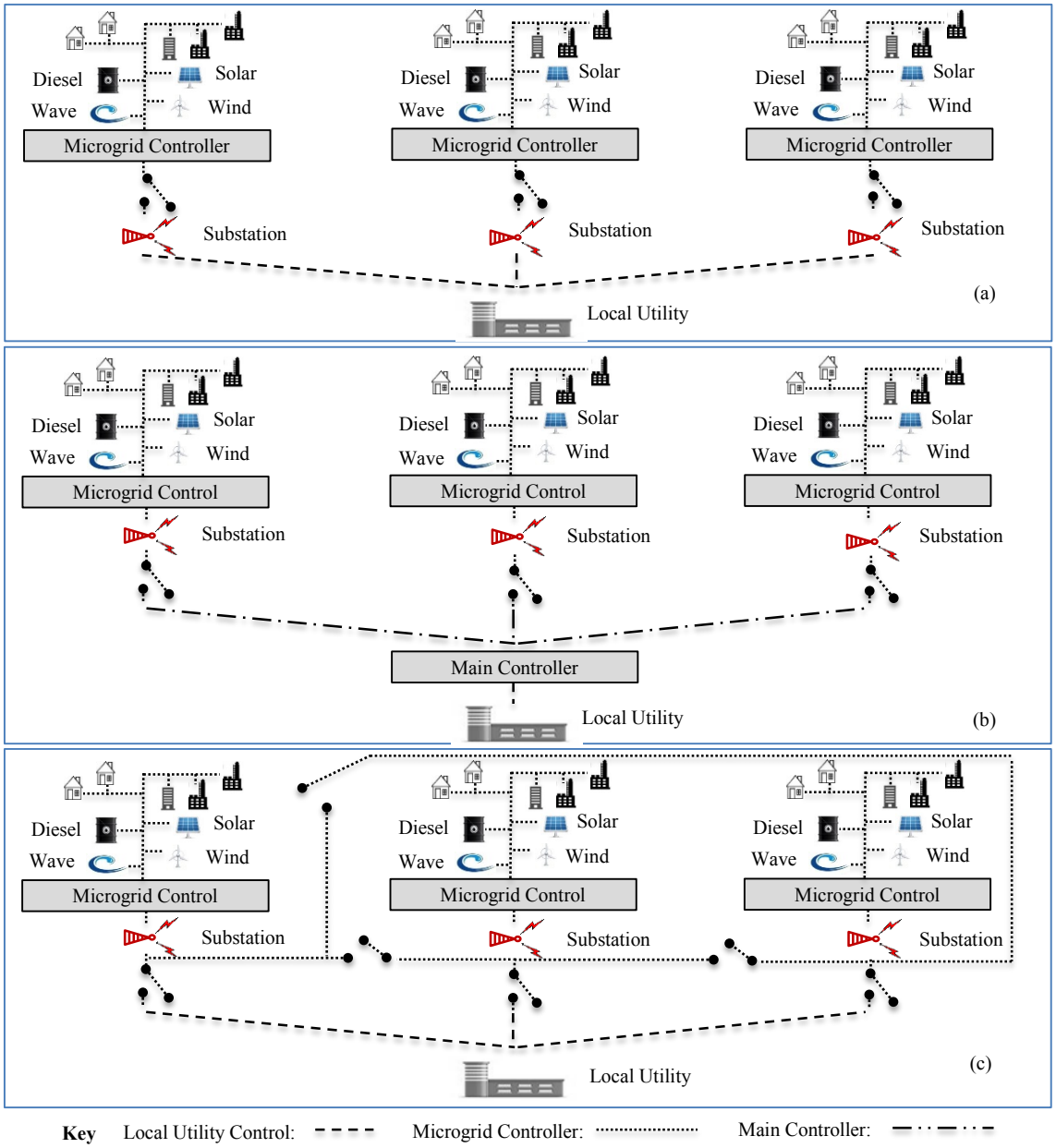


Fig. 16. Self-healing microgrids under different interaction strategies ((a) isolated self-healing; (b) centralized self-healing; (c) cooperative self-healing)

Interconnected microgrid systems use an isolated interaction strategy to detect, isolate, and recover from faults or other issues within its boundaries. An isolated microgrid controller can identify the amount of energy produced by its distributed generation resources as well as the status of electricity coming in from its servicin1g local utility.

However, a drawback of employing isolated system control in microgrids is that it does not allow self-healing as energy resources are limited to what the local utility and distributed generation resources within its boundaries can provide. Conceptually, this type of interaction strategy would be best utilized in microgrids that are housed on campus-like facilities such as universities, military bases, and business centers. Additionally, since there is no external connection to detect faults or transfer resources outside of its designated substations, microgrids operating under isolated control are not suitable for large-scale smart grid topologies. They may, however, serve as the ideal building blocks.

In Fig. 16 (a), each microgrid controller solely manages all loads within their purview. Nonetheless, they cannot interact with the other microgrids although they share a local utility. Hence, for the purposes of real-time decision making and control, each microgrid is modeled as a separate simulation that is capable of running their own DAS^{SH}. The sensors attached to each of the renewable generation resources enable the microgrid to dynamically determine how much, if any, energy to purchase from the local utility. Here, each microgrid is able to dynamically detect abnormalities (i.e., power events) with the local utility. Once such a condition is detected, each individual microgrid controller can isolate its site from the local utility and rely solely on its distributed generation to meet the demand needs of its customers.

On the other hand, a centralized method of interaction in microgrids utilizes an external controller to monitor and regulate the interactions of each microgrid within the self-healing grid topology. Each microgrid acts almost exactly as it would in an isolated interaction strategy with the exception that the external grid controller detects abnormalities or faults occurring either from the local utility or from a particular microgrid.

It then uses DAS^{SH} to determine which microgrids provide support to the microgrid/grids experiencing the fault. The central controller can also detect abnormalities occurring from the feed to the local utility and isolate all the microgrids from the local utility to prevent the fault from propagating to the equipment in each microgrid.

A centralized interaction strategy (Fig. 16 (b)) takes the DAS^{SH} concept of the local interaction strategy and then adds a central controller that dynamically monitors and collects data from the individual microgrids as well as the local utility. The DAS^{SH} controller then uses that data to determine whether or not to isolate the microgrids from the local utility as well as to select which microgrids share energy with each other at any given time. A centralized interaction strategy's main advantage is that is allows a single controller to view the entire smart grid network at one time. The primary disadvantage of a centralized control strategy is that as the network of microgrids grows, so does the time needed to respond to abnormalities within the network. Also, adding separate controllers outside of the microgrids adds considerable equipment and labor costs to the network.

The third possible interaction strategy that can be applied between microgrids is the cooperative. In many ways, a cooperative interaction strategy (Fig. 16 (c)) is similar to the cases of isolated and centralized interaction strategies. Each microgrid manages its own distributed generation and optimizes the amount of energy purchased from the local utility to reduce costs under normal conditions. There is one thing about the cooperative interaction strategy however, that makes it more useful than the other two in a number of tasks. This is its ability to communicate with each of the other microgrids within the topology.

A cooperative interaction strategy fits into the DAS^{SH} framework as each of the microgrid controllers receives real time status updates from each of the other microgrids in the topology. It then uses that information to determine whether or not to share its energy with the microgrid experiencing the anomaly. The main advantage of a cooperative interaction is that it allows a grid to self-heal by sharing energy without the added cost of building a separate control system. However, this system uses a lot more computational resources than the local interaction strategy and, similarly to a centralized strategy, necessitates some form of sophisticated way to reduce the computational burden.

5.1. Adaptive Energy Routing Mechanism using DAS^{SH}

In order to effectively address the real time monitoring and control of self-healing power grids, a powerful Dynamic Adaptive Simulation technique (DAS^{SH}) has been applied. Within the framework, a vast amount of data is collected, several procedures and protocols are utilized, and a complex multi-objective problem is addressed. Initially, we provide the formulation of this multi-objective problem, namely economic emission load dispatch (EELD). The EELD is solved within every microgrid in order to acquire the optimal resource allocation (i.e. simultaneous minimization of the cost as well as total output of emissions). Then, we present the details of the investigated DAS^{SH} framework for self-healing microgrids. This framework includes a cooperative system and control. Finally, we describe the elements of the considered case study, which demonstrates the structure of the realistic model created and all of the feeder and agent therein. To this end, we explore a realistic, notional microgrid network consisting of three separate microgrids

(University, Civic, and Military) with different features, priority sets, and renewable power sources.

5.2. Self-healing in Microgrids

The primary goal in the self-healing networking structure in this work is to ensure that the most important electrical needs are satisfied while; the total cost incurred is minimized. Cooperative system interaction is identified as the backbone of the described structure. In an abnormal situation where one or more microgrids suffer from power events, the cooperative communication and control facilitates the meeting of requested demands within each individual microgrid by their very own generation sources first, and then searches for neighboring microgrids or the enclosed utility service for back-up. This is performed in order to avoid jeopardizing the microgrid's stability and security starting with its critical loads.

In order to improve the efficiency of the self-healing structure, each individual microgrid is assigned a priority number which determines the importance of satisfying the loads within this system. For instance, a microgrid with priority 1 is given higher attention in serving the loads than a microgrid with priorities 2 or 3. Here, we also define energy excess in a microgrid as the renewable energy that remains after satisfying all the loads within the same microgrid, without taking into account the energy bought from the utility provider. While it is quite rare for the renewables to exceed the energy needs in a microgrid, sharing that amount of energy reduces the total cost in the interconnected microgrid system significantly. In the event that no power event is observed in any of the interconnected microgrids and a microgrid has excess energy, it sends a signal sequentially to the rest of

the other connected microgrids. If there are multiple simultaneous requests to utilize this excess energy, then it is sent to the requester microgrid that is closest to the microgrid that hosts this excess energy so that the transmission losses are minimized.

When a microgrid is suffering a blackout, it sends a signal for aid. Then, the microgrids around it check the current frequency value (i.e. the oscillations of the alternating current (AC) of the power grid). We note here that different microgrids may be connected to different utility providers, and thus, the frequencies might vary. Examining the frequency of a power network can produce valuable information about its state. The normal value of the frequency in the USA is 60 Hz, and values that differ significantly from that value indicate problems in the main grid. The system frequency is actually illustrating the dissimilarity between generation and consumption and thus, is a crucial factor for load control. The trend dictates that if an unanticipated increase between the ratios of the energy consumption over the energy generation occurs caused by either a failure in the transmission system or of the main generators, the frequency drops. On the other hand, the system frequency rises if the total system generation is steady, but an interconnected system (e.g. microgrid) disconnects.

The microgrids check the frequency in order to determine in which way they may aid the microgrid that suffers the blackout. If the frequency value is in its normal range (59.6 Hz - 60.4 Hz) the closest microgrid acts as an intermediary between its utility provider and the microgrid that suffers the blackout. On the other hand, if the frequencies are in a critical range (i.e. outside of the normal range) then the utility providers cannot be used for buying additional energy without risking the security of the power network. Thus, in this case, the microgrids with the lower priorities consecutively disconnect their non-critical loads and

share that amount of energy with the microgrid that suffers the blackout. This ensures that the critical and priority loads are satisfied first. In this way, the amount of energy bought from the utility provider does not increase, and the system as a whole is not put into jeopardy.

Finally, in cases of extreme crisis when multiple microgrids are suffering blackouts, the system manages the renewable energy generation and the diesel generators within their boundaries in order to satisfy only critical and priority demands of the microgrids with the higher priority. The protocol for the actions of the interconnected microgrids is illustrated in Fig. 17.

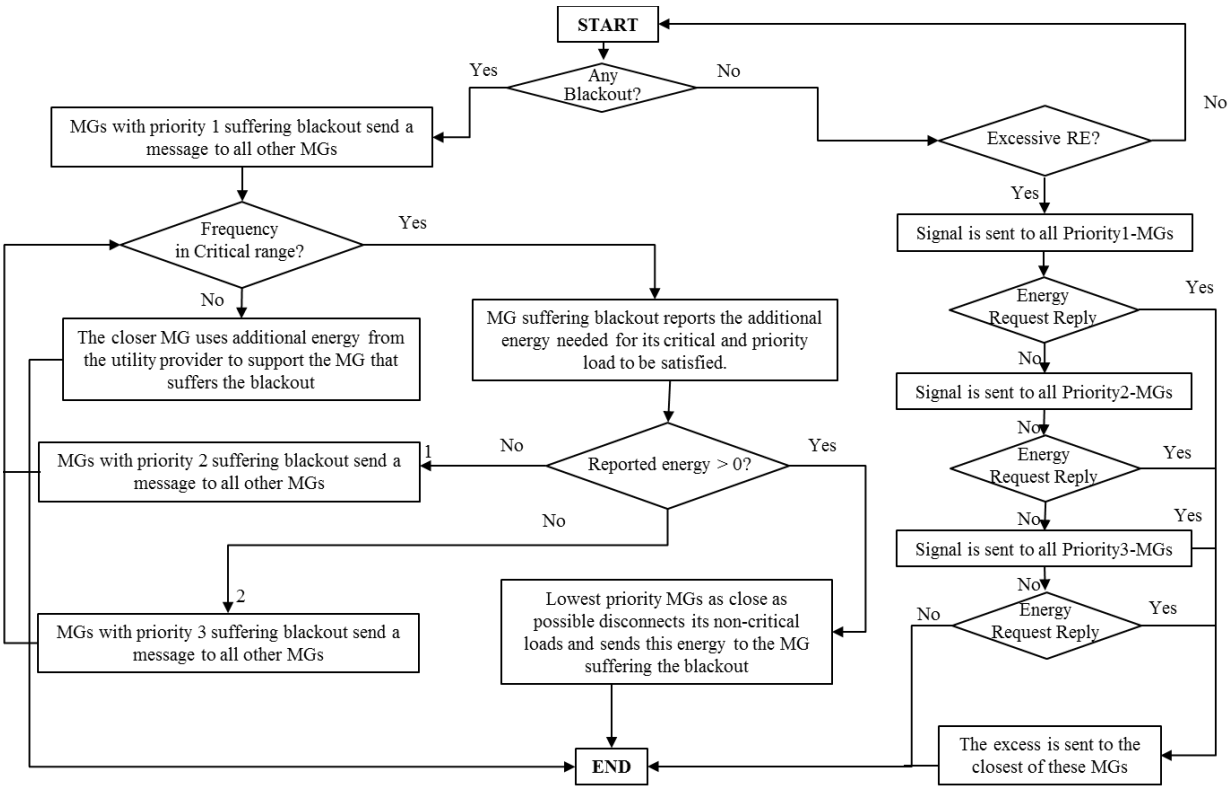


Fig. 17. Self-healing procedure for an interconnected microgrid topology

5.3. Details of the Considered Case Study

To further illustrate the concept of a self-healing power grid, this case study utilizes a notional network of three interconnected microgrids employing a distributed communications strategy The three notional microgrids represent a military base (MBM), a local university (LU), and a government center (GCM). The entire compilation includes 308 buildings and 11 feeders. The average annual demand for the MBM is 130.6 GWh; the average annual demand for the LU is 67.8 GWh; and the average annual demand for the GCM is 84.8 GWh. In addition, the buildings within each microgrid are also prioritized into critical, priority, and non-critical demand categories. Critical demand areas include areas that are required for the administration and control of the community within the microgrid. Examples of these types of areas are city hall for the GCM, the school administration building for the LU, and major headquarters for the MBM. Also included in the critical demand areas are buildings that house first responders and medical treatment facilities. Priority demand includes the buildings in which typical day-to-day functions take place. Examples of these facilities include drivers' license offices for the municipal microgrid, research laboratories for the university and logistics facilities for the military base. Finally, non-critical demand buildings for all three microgrid types include the recreation, housing, and shopping facilities. The MBM being the most critical has the largest diversity of distributed generation resources. These resources include solar, wind, wave, and backup diesel generation. The GCM is next highest in priority but is also the most limited in terms of land area. Because of this the GCM only contains solar power generation. Lastly, the LU is third in terms of priority, but because of its location it is able

to support both solar and wind generation. Table 9 shows the average annualized demand for each demand type of the 3 microgrids appearing in the simulation.

Table 9: Average Annualized Demand by Type (in GWh)

	MBM	**LU**	**GCM**
Critical	31.7	16.1	37.4
Priority	40.1	23.6	26.2
Non-Critical	58.8	28.1	21.2
Total	130.6	67.8	84.8

In the aforementioned notional network, agent based modeling and simulation provides an outstanding method to apply the DAS^{SH} framework to self-healing power grids. There are three distinct areas that define an agent-based simulation: a series of agents, the relationships between the agents, and the environment in which the agents interact ([5] and [6]). In this case study, each building, distributed generation resource, and local utility are represented as agents. The agents interact with one another by giving and receiving electrical power. The distributed generation resources and the local utility agents meet the demand requirements of the building agents. Additionally, each microgrid works as an independent system that contains several components that are independent as well. All these components interact with each other within the same self-healing smart grid environment.

In order to simulate every singular component of the system, its behavior, and to fully capture its interactions with all other units, having an agent for each node is ideal. To this

end, there is an individual agent to represent the electricity demand for each of the 308 buildings and another agent to represent each of the 11 power distribution feeders in the self-healing power grid network. Additionally, the simulation uses an agent for the Photovoltaic arrays located at each microgrid, an agent representing the wind farm at the MBM and the LU, and finally, an agent to represent the wave generator at the MBM.

In each microgrid, electricity is delivered to the end users by way of the feeders that are connected to the substation. The electricity channeled through those feeders is then distributed to the different types of users receiving electricity from that particular substation. The MBM has 5 feeders delivering power to 186 buildings from the substation to its customers. The LU has 3 feeders delivering power to 64 buildings from the substation to its customers. The GCM has 3 feeders delivering power to 58 buildings from the substation to its customers. Figure 18 shows the self-healing smart grid framework developed for this case study.

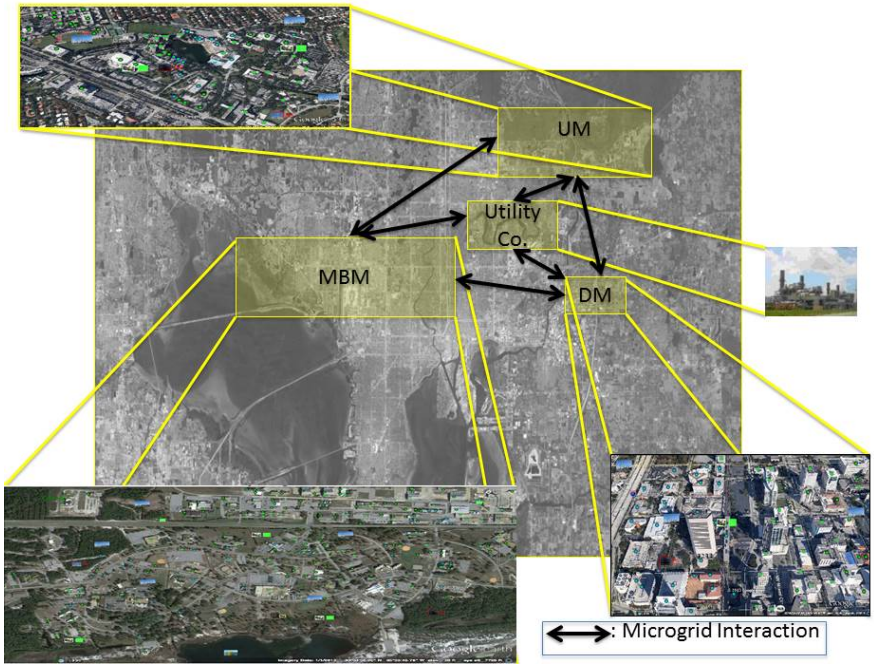


Fig. 18. An exemplary energy routing framework

The data collected for this study consists of real data collected over a period of 2 years by the National Renewable Energy Laboratory from sensors located approximately 60 miles from Panama City, Florida for solar irradiance and wind speed. Additionally, a buoy located along the coast of Destin, FL provided data for the wave generator. As the model received this data, it filtered it using a Fault Detection, Fault Isolation, and Recovery (FDIR) algorithm to check the system at different levels of detail in order to limit the computational burden while solving the EELD problem for each microgrid. Finally, if the model detected a fault in the system, the microgrids are programmed to share energy in order to self-heal the power network as a whole and satisfy the highest possible amount of critical loads.

5.4. Experiments and Results

One of the purposes of implementing a self-healing grid is to reduce costs of energy. For this reason, the added material and labor costs of a separate controller make the centralized control strategy unfeasible. The DAS^{SH} model developed as part of this work simulates a local and distributed control strategy. Furthermore, given the current state of the existing main power grid, the DAS^{SH} seeks to determine the percentage of each demand type able to be satisfied under the following scenarios:

- Scenario A: A deliberate cyber-attack is directed toward the local utility, which renders it inoperable for a period of 24 hours.

- Scenario B: A major hurricane completely wipes out power to the GCM for 48 hours. However during that time frame, workers in critical buildings must come to work to facilitate the clean-up effort.

- Scenario C: A terrorist attack within the borders of the LU forces the MBM to isolate from the local utility for a period of 2 hours until the threat has passed. However, the attack damaged the LU link to the local utility and it will require 6 hours to repair.

- Scenario D: While building a new research facility, construction crews cut the LU's link to the local utility. The link to the local utility will require 24 hours to fully restore services.

The results obtained from the simulation model executing based self-healing of a local and distributed smart grid configuration under the aforementioned scenarios are presented below: The results show what percentage of the demand the DAS^{SH} is able to meet for the local and distributed control strategies. Table 10 shows the percentage of self-healing that can be achieved when using either a local control strategy or distributed control.

Table 10: Self-healing Percentages under Local Control (Left) and Distributed Control (Right)

Scenario	MBM Loads			LU Loads			GCM Loads		
	Critical	Priority	Non-critical	Critical	Priority	Non-critical	Critical	Priority	Non-critical
A	96.5/97.2	48/47.8	0/0	57.3/57.4	4.1/4	0.2/0	29.7/29.6	0/0	0/0
B	100/100	100/100	100/95.7	100/100	100/93.2	100/27.9	46/100	0/0	0/0
C	97.6/98.6	79/94	66.4/66.4	45.2/97	4/41.1	0/6	100/99	100/52.1	100/26
D	100/100	100/100	100/75.9	57.5/100	4/90.1	0.2/8.6	100/98.4	100/81.3	100/28.2

From Table 10 we can see that in scenario A, the local and distributed control strategies yield the same result. This is because when the local utility is taken out of the picture, each microgrid is forced to rely solely on their distributed generation resources to supply power to meet their demands. When this occurs there is no excess energy as each microgrid is using its full capability to meet its own critical demands. One resolution to this would be to install enough renewable energy resources in combination with backup diesel generation to satisfy their critical demand needs. All microgrids in this simulation model lack the required land area to add additional renewable resources, leaving additional diesel generation as the only viable option. Table 11 shows updated results for scenario A when an additional 7 MW diesel generator is added to the MBM or LU.

Table 11: Self-healing Percentages under Local Control/Distributed Control with additional 7MW of Diesel Generation at the MBM/LU

		MBM Loads			LU Loads			GCM Loads		
		Critical	Priority	Non-critical	Critical	Priority	Non-critical	Critical	Priority	Non-critical
MBM	Local Control	100	100	50.1	51.7	4.1	0	31.6	0	0
	Distributed Control	100	99.8	0	41.9	4.1	0	70.4	26.8	7.9
LU	Local Control	93.9	57.9	0.5	100	95.6	58.6	35.3	0	0
	Distributed Control	94.4	82.4	7.4	100	83	9.6	40.9	8.9	1.7

Based on prioritization, when the extra 7MW diesel generator is added to the MBM, under distributed control, the GCM is able to see a significant increase in its ability to meet its critical and priority demands even though the GCM is still experiencing a significant energy deficit for all its demand categories. When the same 7MW diesel generator is instead installed at the LU, the MBM experiences a significant increase in meeting its priority demand requirements. The GCM, while seeing some increase in the percentage of the demand it is able to meet, is still experiencing significant outages.

In scenario B, when the hurricane destroyed the local utilities connection to GCM, it was only able to meet 46 % of its demand requirements with its onsite renewable generation under the local control strategy. However, in the distributed control strategy, the LU and

MBM were able to disconnect some of their less critical buildings to ensure the needed facilities within the GCM were fully functional.

With scenario C, after the terrorist attack the MBM was able to meet over 97% of its critical demands and a large majority or its priority and non-critical demand under the local control strategy. However, the LU was only able to satisfy 45.2% of its critical demands and almost none of its others. Also, the GCM was largely unaffected under local control. Conversely, when utilizing the distributed control strategy the MBM was able to achieve a greater percentage of its priority loads and the LU was able to meet 97% of its critical loads. On the other hand, the GCM was only able to satisfy approximately half of its priority and one fourth of its non-critical loads.

Finally, in scenario D the LU suffered significantly utilizing the local control strategy as only 57.5% of its critical loads and almost none of its other loads were able to be satisfied. However, under the distributed control strategy, the MBM and GCM were able to disconnect non-critical and priority loads to assist in fully restoring the critical loads and partially restoring the priority loads of the LU.

Chapter 6: Conclusions and Future Work

Microgrids are pioneering systems that offer managerial independence, operational independence, emergent behavior, geographic distribution, and evolutionary development. This dissertation reveals that Dynamic Adaptive Simulation is a promising method to model such systems as it provides means to find the most efficient method to optimize and enhance the microgrids' operation and control and attain several benefits. The focus on this thesis is to solve the economic and environmental load dispatch problem optimally and efficiently, to achieve a sophisticated autonomous control of microgrids, and to promote the cooperation between individual microgrids to increase the power network reliability and energy surety.

For addressing the challenge of solving the EELD problem in power networks efficiently, a DAS^{EELD} framework has been designed. The decision making capability of the framework resides in its algorithms developed for grid topology exploration and clustering, multi-objective optimization, state estimation, and fidelity selection. The framework also encompasses three databases for storing information related to sub-networks, fidelities, and measurements. The proposed DAS^{EELD} framework has been demonstrated on a modified version of the IEEE-30 bus system. The presented results consistently reveal that the proposed framework is able to assess the system status and determine a simulation fidelity leading to a compromise solution which ranks among the global best possible solutions, while saving significantly from the computational resource utilization. Future ventures of this research include both methodological and technological extensions to the proposed framework. Methodological extensions can be performed in the development of the fidelity selection and culling algorithm, combining the current selection procedure with an advanced optimal computing budget allocation (OCBA) algorithm to

improve the quality of the solutions, while ensuring that a realistic computational threshold is respected. Furthermore, the effect of the variation in demand levels within the discovery procedure may be investigated to determine an optimal and feasible range. Technologically, the impact of integrating the DAS^{EELD} framework with high-speed sensor networks on the system's overall performance may be studied.

In order to achieve autonomous control of the operation of microgrids, a novel dynamic adaptive simulation framework $DAS^{CONTROL}$ has been created that significantly accelerates the real-time computation of the resource allocation and controls decisions to optimize the operational cost, energy surety, as well as emissions per MW. This version of the DAS framework includes a database responsible for collecting data from electrical and environmental sensors; a fault detection/isolation algorithm that identifies liabilities as well as potential threats within the microgrid; an agent-based simulation of the microgrid system that includes modules for energy generation from renewable and fuel sources, and a heuristic algorithm for fast optimization in terms of cost and demand satisfaction; an optimal computing budget allocation-based control design selection algorithm that uses the agent-based simulation to select the best control design of the microgrid; and a multi-objective algorithm for optimizing the decisions of the microgrid given the best control design. The $DAS^{CONTROL}$ framework is quite promising in determining the optimal level of detail to analyze a system while minimizing the simulation time and required computational resources involved therein. For validation purposes a realistic case study microgrid has been utilized. The results show that the computational time is significantly reduced (50.3% on average) while the quality of the solutions is not compromised. Nowadays, with the available parallel computing and super computing capabilities, a

decrease in the computational complexity of the magnitude of 50% might seem negligible. However, the more computational capabilities that are available, the more complex and detailed the simulated systems become and hence, a decrease in the computational complexity is always significant. In the future, this work can expand to apply these concepts to multiple microgrids simultaneously. As new renewable energy sources become available and more efficient, the simulation can add a robust storage element to store excess electricity for future use. Additionally the $DAS^{CONTROL}$ framework can be applied to an even more complex microgrid system where there are more sources of renewable generation as well as electric vehicles. Finally in our future venues is the idea of embedding an optimization genetic algorithm within the simulation that will use the Optimal Costing Budget Allocation technique to rank the produced solutions in every iteration.

Finally, a version of the DAS framework has been developed to provide distributed microgrids with a protocol of self-healing, both when they are operating collaboratively and competitively (in an isolated mode) while increasing the reliability of the network by pledging energy surety (DAS^{SH}). In DAS^{SH} framework, an analytical protocol is built to guide the considered microgrids' responses in the event of emergencies such as severe weather phenomena, damaged infrastructure, or cyber-attacks. This framework has been applied to a realistic case study that includes three microgrids and has been tested under four different emergency incidents. The results reveal that the cooperative collection of distributed microgrids were able to meet the critical and priority loads to a higher extent at all times while sacrificing from the less important non-critical loads, hence accomplishing a significantly improved management of energy resources. While this protocol is not able to meet 100 percent of all demands in the modeled scenarios, it does ensure that the most

critical facilities of an organizational structure have the electricity to perform their most essential tasks in times of crisis. While this framework focused on self-healing of electrical infrastructure based on the predicted total demand from buildings in the microgrid construct, it does not have a way to segment the demands within particular buildings. Future work in this area would determine a method to dynamically model the demands for individual rooms or sectors in a building. Similar to how each building is assigned a priority, each room within a building can also be assigned a priority. One can safely assume that not all the demand requirements located within a critical building are indeed critical. Similarly some priority buildings may house some critical assets. Furthermore, dynamic modeling of different demand priorities within a building and adjust those priorities in real-time given factors such as time of day/week/month/year, is amongst the future venues of this research in order to fully satisfy the critical demands and a much larger proportion of the priority demand requirements.

References

[1] K. Moslehi, and R. Kumar, 2010. "A reliability perspective of the smart grid," *IEEE Transactions on Smart Grid*, 1, pp 57–64.

[2] D. T. Nguyen, and L.B. Le, 2014. "Optimal bidding strategy for microgrids considering renewable energy and building thermal dynamics," *IEEE Tranactions on Smart Grid*, 5, pp. 1608–1620.

[3] C. Ahn, and H. Peng, 2013. "Decentralized voltage control to minimize distribution power loss of microgrids," *IEEE Transactions on Smart Grid*, 4, pp. 1297–1304.

[4] F. A. Mohamed, and H. N. Koivo, 2007. "System modelling and online optimal management of microgrid with battery storage," *In Proceedings of the 2007 ICREPQ*, pp. 5.

[5] Macal, C., North, M., 2005. "Tutorial on agent-based modeling and simulation," *In Proceedings of the 2005 Winter Simulation Conference*, pp. 2–15.

[6] Macal, C., North, M., 2006. "Tutorial on agent-based modeling and simulation, Part 2: How to model with agents," *In Proceedings of the 2006 Winter Simulation Conference*, pp. 73–83.

[7] Energy Policy Act, 2005. Pub. L. No. 109-58, 109th Cong., Accessed August 8. http://www.gpo.gov/fdsys/pkg/PLAW-109publ58/pdf/PLAW-109publ58.pdf.

[8] Zhang G.L., Lu H.Y., Li G. Y., and Zhang G. Q., 2005. "Dynamic economic load dispatch using hybrid genetic algorithm and the method of fuzzy number ranking," *In Proceedings of the Fourth International Conference on Machine Learning and Cybernetics,* 4: pp. 2472–2477.

[9] Y. Ouyang, J. E. Zhang and S. M. Luo, 2007. "Dynamic data driven application system: Recent development and future perspective," *Ecological Modelling*, 204, pp. 1–8.

[10] Darema, F., 2004. "Dynamic data driven applications system: A new paradigm for application simulations and measurements," *In Proceedings of the 2004 International Conference on Computational Science*, pp. 662–669.

[11] J. Mandel, M. Chen, L. Franca, C. Johns, A. Pulhalskii, J. Coen, C. Douglas, R. Kremens, A. Vodacek and W. Zhao, 2004. "A note on dynamic data driven wildfire modeling," *In Fourth International Conference on Computational Science*, pp. 725–731.

[12] Carnahan, J.C. and Reynolds, P.F., 2006. "Requirements for DDDAS flexible point support," *In Proceedings of the 2006 Winter Simulation Conference*, pp. 2101–2108.

[13] Wang, Z., Li, M. Khaleghi, A. M., Xu, D., Lobos, A. Vo, C. Lien, J.M., Liu, J., Son., Y.J., 2013. "DDDAMS-based crowd control via UAVs and UGVs," *In Proceedings of the 2013 International Conference on Computational Science*, pp. 2028–2035.

[14] Frew, E., Argrow, B., Houston, A., Weiss, C., Elston., J., 2013. "An energy-aware airborne dynamic data-driven application system for persistent sampling and surveillance," *In Proceedings of the 2013 International Conference on Computational Science*, pp. 2008–2017.

[15] Han, S.Y., DeLaurentis, D.A., 2013. "Development interdependency modeling for system-of-systems (SoS) using Bayesian networks: SoS management strategy planning," *In Proceedings of the 2013 Conference on Systems Engineering Research*, pp. 698–707.

[16] C. Park, J. Tang, and Y. Ding, 2010. "Aggressive data reduction for damage detection in structural health monitoring," *Structural Health Monitoring*, 9, pp. 59–74.

[17] A. Khaleghi, D. Xu, Z. Wang, M. Li, A. Lobos, J. Liu, and Y. Son, 2013. "A DDDAMS-based planning and control framework for surveillance and crowd control via UAVs and UGVs," *Expert Systems with Applications*, 40, pp. 7168–7183.

[18] E. Blasch, Y. Al-Nashif, and S. Hariri, 2014. "Static versus dynamic data information fusion analysis using DDDAS for cyber security trust," *In Proceedings of the 2014 International Conference on Computational Science*, pp. 1299–1313.

[19] D. Allaire, and K. Willcox, 2014. "A mathematical and computational framework for multifidelity design and analysis with computer models," *International Journal for Uncertainty Quantification*, 4, pp. 1–20.

[20] A. E. Thanos, D. Moore, X. Shi, and N. Celik, 2015. "System of systems modeling and simulation for microgrids using DDDAMS," *A Handbook for Modeling and Simulation for System of Systems*, Wiley and Sons.

[21] A. E. Thanos, X. Shi, J. P. Saenz, and N. Celik, 2013. "A DDDAMS framework for real-time load dispatching in power networks," *In Proceedings of the 2013 Winter Simulation Conference*, pp. 1893–1904.

[22] N. Celik, A. E. Thanos, and J. P. Saenz, 2013. "DDDAMS-based dispatch control in power networks," I*n Proceedings of the 2013 International Conference on Computational Science*, pp. 1899–1908.

[23] A. E. Thanos, M. Bastani, N. Celik, and C. H. Chen, 2015. "Dynamic data driven adaptive simulation framework for automated control in microgrids," *IEEE Transactions on Smart Grid.*

[24] Parnas, D.L., 1979. "Designing software for ease of extension and contraction," *IEEE Transactions on Software Engineering*, 5, pp. 128–138.

[25] Hoffman C., Swain E., Xu Y., Downar T., Tsoukalas L., Top P., Senel M., Bell M., Coyle E., Loop B., Aliprantis D., Wasynczuk O., and Meliopoulos S., 2007. "DDDAS for autonomous interconnected systems: National energy infrastructure," *In Proceedings of the 2007 International Conference on Computational Science*, pp. 1074–1082.

[26] Amin S., Giacomoni A., 2012. "Smart grid - safe, secure, self-healing: Challenges and opportunities in power system security, resiliency, and privacy," *in IEEE Power and Energy Magazine*, pp 33–40.

[27] Zhi G., 2005. "Current status and prospects of wide area protection (dynamic stability control)," *Journal of Power System Technology*, pp. 2004–2008.

[28] Sáenz J.P., Celik N., Xi H., Son Y., and Asfour S., 2013. "Two-stage economic and environmental load dispatching framework using particle filtering," *International Journal of Electrical Power & Energy Systems,* 48*,* pp. 93–110.

[29] Abido M.A., 2009. "Multi-objective particle swarm optimization for environmental/economic dispatch problem," *Journal of Electric Power Systems Research*, 79, pp. 1105–1113.

[30] Colson C.M., Nehrir M.H., 2009. "A review of challenges to real-time power management of microgrids," *In Proceedings of the 2009 IEEE Power and Energy Society General Meeting*, pp. 1–8.

[31] Rakpenthai, C., Premrudeepreechacharn, S., Uatrongjit, S, Watson, N. R., 2005. "Measurement placement for power system state estimation using decomposition technique," *Electric Power Systems Research,* 75(1), pp. 41–49.

[32] Thanos A.E. and Celik N., 2013. "Online state estimation of a microgrid using particle filtering," *In Proceedings of the 2013 Annual Industrial and Systems Engineering Research Conference*, pp. 316–325.

[33] Phonrattanasak P., 2010. "Optimal placement of DG using multiobjective particle swarm optimization," *In Proceedings of 2nd International Conference on Mechanical and Electrical Technology*, pp. 342–346.

[34] H. Jiang, J. J. Zhang, W. Gao, and Z. Wu, 2014. "Fault detection, identification, and location in smart grid based on data-driven computational methods," *IEEE Transactions on Smart Grid*, 5, pp. 2947–2956.

[35] Z. Zhang, S. Gong, A. D. Dimitrovski, and H. Li, 2013. "Time synchronization attack in smart grid: Impact and analysis," *IEEE Transactions on Smart Grid*, 4, pp. 87–98.

[36] X. Zhang, S. Chen, Y. Zhu, and W. Yan, 2011. "Fault detection and diagnosis for steam turbine based on kernel GDA," *In Proceedings of the 2011 ICMIC*, pp. 58–62.

[37] C. H. Chen, and L. H. Lee, 2010. "Stochastic simulation optimization: An optimal computing budget allocation," New Jersey: World Scientific.

[38] W. Chen, S. Gao, C. H. Chen, and L. Shi, 2014 "An optimal sample allocation strategy for partition-based random search," *IEEE Transactions on Automation Science and Engineering*, 11, pp.177–186.

[39] B. W. Hsieh, C. H. Chen, and S. C. Chang, 2007. "Efficient simulation-based composition of scheduling policies by integrating ordinal optimization with design of experiment," *IEEE Transactions on Automation Science and Engineering*, 4, pp. 553–568.

[40] S. Yan, E. Zhou, and C. H. Chen, 2012. "Efficient selection of a set of good enough designs with complexity preference," *IEEE Transactions on Automation Science and Engineering*, 9, pp. 596–606.

[41] M. Bastani, A.E. Thanos, N. Celik, and C. H. Chen,2014. "Efficient design selection in microgrid simulations," *In Proceedings of the 2014 Winter Simulation Conference*, pp. 2762–2773.

[42] G. Mavrotas, 2009, "Effective implementation of the ε-constraint method in multi-objective mathematical programming problems," *Applied Mathematics and Computation*, 213, pp. 455–465.

[43] R. E. Steuer, 1986. "Multiple criteria optimization: Theory, computation and application," New York: Wiley.

[44] K. M. Miettinen, 1999. "Nonlinear multiobjective optimization," Boston: Kluwer Academic Publishers.

[45] J. Zhao, E. Mazhari, N. Celik and Y. J. Son, 2011. "Hybrid agent-based simulation for policy evaluation of solar power generation systems," *Simulation Modelling Practice and Theory*, 19, pp. 2189–2205.

[46] J. P. Sáenz, N. Celik, S. Asfour and Y. J. Son, 2012. "Electric utility resource planning using continuous-discrete modular simulation and optimization (CoDiMoSO)," *Computers & Industrial Engineering*, 63, pp. 671–694.

[47] Y. Rinott, 1978. "On two-stage selection procedures and related probability inequalities," *Communications in Statistics*, A7, pp. 799–811.

APPENDIX I: OCBA Java Code

```
import java.util.*;
import java.io.*;

public class MicrogridOCBA
{
private static int ND;
private static int ADD_BUDGET;
private static int[] n;
private static int[] an;
private static int b;

public static void ocba(double[] mean,double[] var, int nd,int[] n,int add_budget,int type)
/*
This subroutine determines how many additional runs each design will should have for
next iteration of simulation.
s_mean[i]: sample mean of design i, i=0,1,...,ND-1
s_var[i]: sample variance of design i, i=0,1,...,ND-1
nd: the number of designs
n[i]: number of simulation replications of design i, i=0,1,...,ND-1
add_budget; the additional simulation budget
an[i]: additional number of simulation replications assigned to design i, i=0,1,...,ND-1
type: type of opitmization problem. type=1, MIN problem; type=2, MAX problem
*/
{
 int i,s,budget,remainingBudget;
 double[] ratio=new double[nd];  /* Ni/Ns */
 double totalRatio;
 double temp=0;
 boolean[] moreRun=new boolean[nd];
 boolean moreAllocation=true;

 if(type==2) /*MAX problem*/
 {
  for(i=0;i<nd;i++) mean[i]=-mean[i];
 }

 b=best(mean,nd);
 s=second_best(mean,nd,b);
 ratio[s]=1;

 for(i=0;i<nd;i++)
 {

  if(i!=s && i!=b)
     {
      ratio[i]=Math.pow((mean[b]-mean[s])/(mean[b]-mean[i]),2)*var[i]/var[s];
```

```
    } /* calculate ratio of Ni/Ns */
  if(i!=b) temp+=ratio[i]*ratio[i]/var[i];
}
  ratio[b]=Math.sqrt(var[b]*temp);

  /* calculate NB */
  budget=add_budget;
for(i=0;i<nd;i++)
{
    budget+=n[i];
    moreRun[i]=true;
  }
  remainingBudget=budget;

  while(moreAllocation)
  {
      moreAllocation=false;
      totalRatio=0;
      for(i=0;i<nd;i++)
      {
         if(moreRun[i]) totalRatio+=ratio[i];
      }

      for(i=0;i<nd;i++)
      {
         if(moreRun[i]) an[i]=(int)(remainingBudget*ratio[i]/totalRatio);
         /*disable those design which have been run too much */
         if(an[i]<n[i])
         {
            an[i]=n[i];
            moreRun[i]=false;
            moreAllocation=true;
         }
      }
      if(moreAllocation)
      {
         remainingBudget=budget;
         for(i=0;i<nd;i++)
         {
            if(moreRun[i]==false) remainingBudget-=an[i];
         }
      }
  }  /*end of WHILE */
       /*calculate the difference */
  remainingBudget=an[0];
  for(i=1;i<nd;i++) remainingBudget+=an[i];
```

```
  an[b]+=(budget-remainingBudget); /* give the difference to design b */
  for (i=0;i<nd;i++) an[i]-=n[i];
}
public static int best(double[] mean, int nd)

 /* This function determines the best design based on current simulation results */
 /* t_s_mean[i]: temporary array for sample mean of design i, i=0,1,...,ND-1
 nd: the number of designs */
  {
  int min_index=0;
  for(int i=0;i<nd;i++)
    {
     if(mean[i]<mean[min_index])
       {
        min_index=i;
       }
     }
  return min_index;
  }

 public static int second_best(double[] t_s_mean,int nd,int b)
 /* This function determines the second best design based on current simulation results */
 /* t_s_mean[i]: temporary array for sample mean of design i, i=0,1,...,ND-1
  nd: the number of designs.
  b: current best design determined by function best() */
  {
  int i, second_index;
  if(b==0) second_index=1;
  else second_index=0;
  for(i=0;i<nd;i++)
    {
       if(t_s_mean[i]<t_s_mean[second_index] && i!=b)
       {
        second_index=i;
       }
     }
  return second_index;
  }
```