Battery Energy Storage System Sizing Using PSO Algorithm in DIgSILENT PowerFactory

Wei Hown Tee* , Khaldon Ahmed Qaid* , Chin Kim Gan*‡ , Pi Hua Tan*,** ,

* Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka

** AFRY Malaysia Sdn. Bhd., Kuala Lumpur, Malaysia

(weihowntee@gmail.com, khaldon603@gmail.com, ckgan@utem.edu.my, tan.pi-hua@afry.com)

‡ Corresponding Author; Chin Kim Gan, Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Jalan Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Tel: +60 62701310, ckgan@utem.edu.my

Abstract- The usage of battery energy storage system (BESS) can be a significant technology to improve the performance of power systems. Optimal sizing of BESS can reduce power losses, improve voltage profile and relieve peak demand in power systems. This paper aims to establish a simulation-based optimization in DIgSILENT Programming Language (DPL) script in DIgSILENT PowerFactory environment, which is more efficient for data exchange between operations. Problem for optimization of BESS sizing and placement are formulated in DPL script with analytic algorithm and particle swarm optimization (PSO) with loadflow simulation. The IEEE 9-bus system is the test case used to demonstrate and discuss the application of algorithms in DPL script. The placement of BESS is identified to be optimal with lowest power losses of 4.962 MW at bus 5 and the optimal BESS size of 47.168 MW with operation cost of 4852.65 \$/h are determined. The output of the study concludes that optimal sizing of BESS can be applied in solving power system problems effectively with DPL operations. This paper serves as a reference for researchers to study and implement optimization approach in DIgSILENT DPL script.

Keywords DIgSILENT PowerFactory, analytic algorithm, particle swarm optimization, battery energy storage system.

Nomenclature

AI	Artificial Intelligent
BESS	Battery Energy Storage System
CE	Cross-Entropy
ComDPL	DPL Command Object
DICOPT	Discrete and Continuous Optimizer
DPL	DIgSILENT Programming Language
GA	Genetic Algorithm
PSO	Particle Swarm Optimization

1. Introduction

In the recent years, optimization techniques have been implemented into power system analysis, operations and controls worldwide. The main contributions of optimization implemented are involved in terms of improved operational reliability, techno economic analysis, system stability and security [1]. Power systems are growing larger and more complicated due to the rising of load demand and integration of renewable energy sources [2][3]. Hence optimization has become important to ensure the development of the power systems in this era. Various optimization problems have been concerned and solved in power system operations such as optimal power flow, economic dispatch and so on [4][5]. Optimization approaches including conventional and artificial intelligent (AI) techniques have been applied to solve problems in power systems [6][7]. The main advantages of AI techniques are the ability to deal with

complex problems. These include mathematical optimization strategies and nature inspired methods [8][9].

Battery energy storage system (BESS) can be applied in different aspects of the power systems. It is currently applied as one of the key factors for sustainable energy in country such as United States (US), Japan, China and so on [10]. It is also proven by recent studies that BESS can be used for ancillary service [11], energy arbitrage [12], improve reliability [13][14] and power quality [15] as well as renewable energy integration [16][17]. Hence it is attracting more attention of the researchers to determine the optimal sizing of BESS integrated with power systems. Authors in [18] proposed BESS sizing method in a smart household by taking into account the electricity price mechanisms which combines the cataclysmic genetic algorithm and discrete and continuous optimizer (DICOPT) solver. A strategy is proposed in [19] to optimize BESS size and placement using genetic algorithm (GA) with linear programming method through linking DIgSILENT with MATLAB. It is time consuming for data exchange between two applications. An optimization approach for BESS placement to enhance oscillation damping in power system is proposed in [20]. The proposed method is applied by connecting MATLAB with DIgSILENT where PSO runs in MATLAB environment while constraints checking is performed in DIgSILENT simulation. Communication between two applications may consume time to exchange time-varying data between the operations [21]. Authors in [22] proposed an optimal placement of storage with Cross-Entropy (CE) optimization. The method is applied in MATLAB and DIgSILENT where the optimization is solved in MATLAB while potential fitness value of each iteration is delivered to DIgSILENT for simulation. The voltage waveforms from simulation will then be returned to MATLAB for next function operation. However, these papers did not consider implementing the optimization in DPL script in DIgSILENT environment. According to [23][21], it will be time consuming to run the operations which requires data exchange between two software such as MATLAB and DIgSILENT. This process has inherent delays that result in slower software interface. The complex operation of data transfer between two software will burden the memory arrays of both software and lead to slow responding of certain complex operations. Larger memory arrays are needed for both software to perform complex operation such as optimization where more data information and transfer are required between the software for every iteration runs. The delay can be reduced by implementing the optimization operation in DPL script where the data transfer for operations are within DIgSILENT software. Besides, there are less studies on DPL script programming for optimization available online as well.

In comparison to previous works, the contribution of this paper is to establish a simulation-based optimization in DPL script in DIgSILENT environment, which is less time consuming for data exchange between operations. This paper serves as a reference source for researchers to study and implement optimization coding in DPL script. BESS optimal sizing and placement with particle swarm optimization (PSO) and analytic algorithm in DPL script are presented in

Pseudo code. The algorithms are tested in IEEE 9-bus system in DIgSILENT environment.

2. Methodology

The sizing methodology is developed in DIgSILENT PowerFactory where it is a strong technical engineering software for the analysis of power system. DIgSILENT Programming Language (DPL) in the program is used to develop automated programmes that link with power system components in DIgSILENT PowerFactory. This language's structure is comparable to that of the C++ programming language [24]. The principle structure of a DPL script is presented in Figure 1. The DPL command object, ComDpl is the core element that links all of the different objects, variables and parameters modelled or defined to different functions or internal elements and then provides results parameters. As a result, the DPL script will perform a sequence of operations or functions within the script. Based on the operation conducted, it interfaces with the database to access and record any modified parameters or results directly into the objects database. Unlike other studies that developed interconnection link between DIgSILENT PowerFactory and optimization algorithm scripting in other programming software such as MATLAB which may have high computation time to transmit and receive data information between two software. In this study, BESS optimal sizing method is developed by DPL script in DIgSILENT.
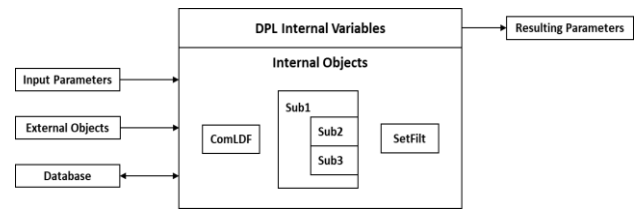

Fig. 1. Structure of DPL script.

2.1. Objective Function

The aim of this study is to reduce power losses in distribution system. An objective optimization is then formulated by minimizing power losses. The objective of optimal BESS sizing is mathematically stated as

$$Objective\ Function = Minimize(Power_{loss}) \tag{1}$$

$$Minimize(Power_{loss}) = \sum_{i=1}^{N} Power_{loss,i} \tag{2}$$

where $Power_{loss}$ is the power loss in MW at line i, N is total line number of the system. The losses for each hour are calculated by using the backward and forward sweep in DIgSILENT powerfactory. The equation can be written as

$$\min f(x) = \Delta Ploss \tag{3}$$

$$\Delta Ploss = \sum_{i=1}^{N} \left[Ploss_i - Ploss(P_i^{BESS}) \right] \tag{4}$$

where $\Delta Ploss$ is the total active power loss in MW. P_i^{BESS} is the active power of BESS in MW. Constraints on the system are given to determine the necessary capabilities of the

system. The constraints of the objective function are presented as

$$0 \leq P_{Loss(BESS)} \leq P_{Loss(old)} \tag{5}$$

where $P_{Loss(old)}$ is the initial power loss of the system, $P_{Loss(BESS)}$ is the new power loss with the integration of BESS in the system.

$$V_i^{\min} \leq V_i \leq V_i^{\max} \qquad i = 1, 2, ..., N \tag{6}$$

where V_i is the voltage at bus i, $V_i^{\min}$ and $V_i^{\max}$ are 0.9 and 1.1 p.u. respectively.

$$0 \leq P_{BESS} \leq P_{genref} \tag{7}$$

where P_{BESS} is the battery power in MW, P_{genref} is the power of reference generator of the system in MW.

2.2. PSO for Optimum Sizing of BESS

PSO is a strategy to find optimal values in complex search spaces that is inspired based on social interactions exhibited by animals. It begins with random particles to investigate optimal values by updating generations. The particles move around in the search space according to the position and velocity of the particles. At each iteration, each particle remembers its own best position found in the search space and is updated by as personal best solution, P_{best}. G_{best} is the best global solution found by any particle in all previous iterations [25]. Position and velocity of each particle are updated by equation (8) and (9)

$$v_j^{i+1} = \omega v_j^i + c_1 r_1 (P_{best}^i - x_j^i) + c_2 r_2 (G_{best}^i - x_j^i) \tag{8}$$

$$x_j^{i+1} = v_j^{i+1} + x_j^i \tag{9}$$

There are more PSO information can be found in [26][27].

The number of iteration (N_i) in this study is 50 followed by the number of particle (N_p) of 30. The inertia weight ω is 0.4. Learning factors, c_1 and c_2 are 0.5 and 0.6 respectively. The process of the BESS sizing is followed as:

1. Declare the constraints of the objective function and initialize the parameters (N_i, N_p, ω, c_1, c_2) and iteration $i = 1$ with random position (x_i) and velocity (v_i).

2. Initialize particle $j = 1$ in the swarm.

3. Run the objective function for particle j^{th} of iteration i^{th}.

4. Explore and determine P_{best} and G_{best} for particle j^{th} of iteration i^{th}. If the fitness function value of jth is less than or equal to the personal best value, the program stores $P_{best} = f(x_j)$.

5. Particle j^{th} is increased by 1. Then condition is checked: If particle j^{th} + 1 less than or equal to N_p, go back to Step 3.

6. Adjust G_{best} of $i^{th} = P_{best}$ at i^{th}.

7. Iteration i^{th} is increased by 1. Then condition is checked: If iteration i^{th} + 1 less than or equal to N_i, update the new position (x_{i+1}) and new velocity (v_{i+1}) for iteration i^{th} +1 by going back to Step 2.

8. If iteration i^{th} + 1 more than or equal to N_i, the process is completed and the optimal size of BESS is obtained.

The steps of BESS sizing in DIgSILENT DPL script based on PSO have been illustrated in Figure 2. The Pseudo code of PSO is shown in Algorithm 1.

2.3. Analytic Algorithm for Optimum Sizing of BESS

Analytic algorithm is a simple and easy method for sizing calculations. The process of the BESS sizing is followed as:

1. Initialize the system parameters (Generators, loads, buses, lines).

2. Read the input data of the system in DIgSILENT PowerFactory program and obtain the values of the elements modelled.

3. Run loadflow simulation, read and store initial power losses of the system.

4. Set initial BESS power = reference generator of the system.

5. Run loadflow simulation, read and store the new power losses of the system.

6. Check the condition if the $Ploss_{new} > Ploss_{old}$, the program will decrease $Pbess$ by 0.5 and go back to step 5. The step will end if the condition is not satisfied.

7. End of process.

The steps of BESS sizing in DIgSILENT DPL script based on analytic algorithm are shown in Figure 3. The Pseudo code of analytic algorithm is shown in Algorithm 2.
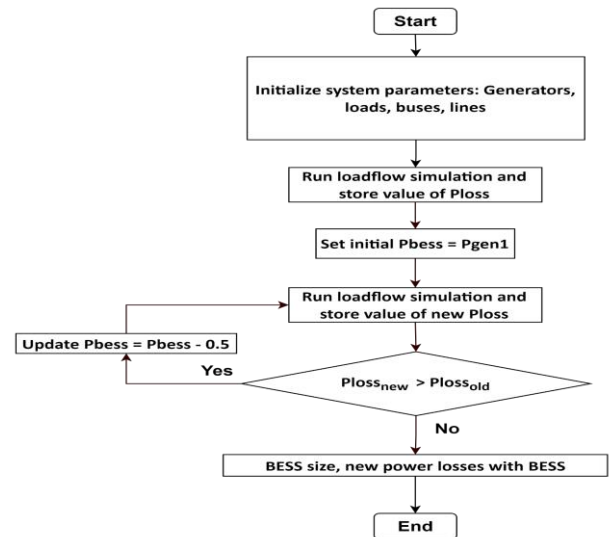

Fig. 3. The flowchart of analytic algorithm for optimum sizing of BESS in DIgSILENT.

2144

	Algorithm 1: Particle swarm optimization (PSO)
01:	**Begin**, initialize the parameters $N_i, N_p, \omega, c_1, c_2$
02:	Initialize DPL object and set of the system *Gen, Load, Bus, Line, BESS, Ldf*
03:	Execute loadflow in DIgSILENT PowerFactory to calculate initial power losses
04:	Set Pbess and evaluate fitness of particles, *f(x)*
05:	Set initial Pbest and Gbest
06:	**for** $i = 1$ to maximum number of iterations, N_i
07:	**for** $j = 1$ to maximum number of particles, N_p
08:	Update the velocity of the particles v_j^{i+1} using equation (8)
09:	Update the position of the particles x_j^{i+1} using equation (9)
10:	Evaluate the fitness of the updated position x_j^{i+1}
11:	Execute loadflow in DIgSILENT PowerFactory
12:	**if** $f(x_j^{i+1})$ satisfies constraints **then**,
13:	Save Pbess and $f(x_j^{i+1})$
14:	**else** $f(x_j^{i+1}) = $ penalty
15:	**end if**
16:	Update the personal best by:
17:	**if** $f(x_j^{i+1}) < f\left[P_{best}(x_{j,Np}^i)\right]$ **then**,
18:	$P_{best\,j,Np}^i = x_j^{i+1}$
19:	**end if**
20:	Update the global best by:
21:	**if** $f(P_{best\,j,Np}^i) < f(G_{best})$ **then**,
22:	$Gbest = P_{best\,j,Np}^i$
23:	**end if**
24:	**end** j
25:	Check stopping criteria, if $i < N_i$ go to step6, else go to next step.
26:	**end** i

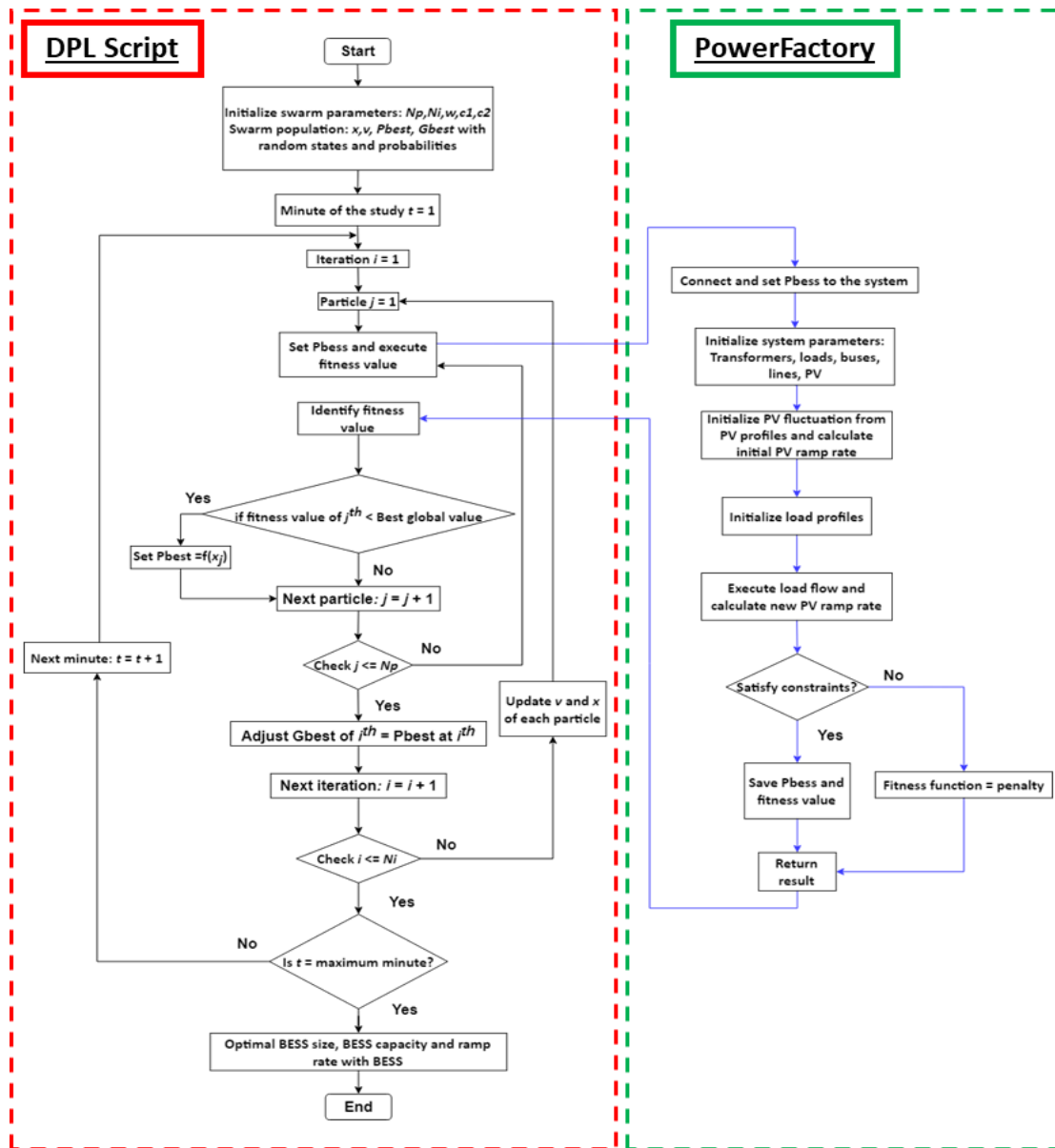|	**Algorithm 2**: Analytic algorithm
01:	**Begin**, initialize DPL object and set of the system *Gen, Load, Bus, Line, BESS, Ldf*
02:	Execute loadflow in DIgSILENT PowerFactory to calculate initial power losses
03:	Set initial power losses = $Ploss_{old}$
04:	Set initial $Pbess = Pgen1$
05:	Execute loadflow in DIgSILENT PowerFactory
06:	Set $i = Ploss_{new}$
07:	**while** $i > Ploss_{old}$
08:	$Pbess = Pbess - 0.5$
09:	Execute loadflow
10:	Update i
11:	Checking stopping criteria, if $i > Ploss_{old}$ go to step8, else stop.
12:	**end while**

Fig. 2. The flowchart of PSO for optimum sizing of BESS in DIgSILENT.

2.4. Operation Cost of Generators After Installing BESS

Analytic algorithm is a simple and easy method for sizing calculations. The cost function of each generator unite is related to the actual power output injected into the system and is usually modeled with a smooth quadratic function. The total generating fuel cost for the system can be calculated by

$$Total\ fuel\ \cos t = \sum_{k}^{M} F_k(P_k) \tag{10}$$

$$\sum_{k=1}^{M} F_k(P_k) = \sum_{k=1}^{M} \left[a_k P_k^2 + b_k P_k + c_k \right] \tag{11}$$

where $F_k(P_k)$ is the fuel cost of kth generating unit in \$/h; M is the total number of generating units in the system; P_k is the power generated by the kth generating unit in MW which calculated from the analytic algorithm or PSO algorithm after installing the BESS in the system; and a_k, b_k, c_k, are the fuel cost correlation coefficients of kth generator [28]. In the operation cost calculation, the BESS cost is assumed to be zero. The operation cost is calculated after installing BESS in each bus. The power output P_k of each generator is taken from the load flow simulation results of DIgSILENT after the BESS installed in each bus of the system.

3. Results and Discussions

The DPL scripts are applied and tested for the effectiveness of BESS sizing using an IEEE 9-bus system. The IEEE 9-bus system is modelled in DIgSILENT PowerFactory as shown in Figure 4. The system consists of 3 load buses, 3 generator buses with 6 branches. The tests were carried out by solving the BESS sizing using both PSO and analytic algorithm scripts in DPL scripts. The total generators, loads and initial power losses for the IEEE 9-bus system are shown in Table 1 [29].

Table 1. IEEE 9-bus system parameters used in the study.

Variable	Value
G1 (reference generator), G2, G3	71.6 MW, 163 MW, 85 MW
Load 1, Load 2, Load 3	125 MW, 90 MW, 100 MW
Initial power losses	5.190 MW
Power max G1, G2, G3	250 MW, 300 MW, 270 MW
Power min G1, G2, G3	10 MW, 10 MW, 10 MW
Coefficient a G1, G2, G3	0.11 \$/(MWh)^2, 0.085 \$/(MWh)^2, 0.1225 \$/(MWh)^2
Coefficient b G1, G2, G3	5 \$/MWh, 1.2 \$/MWh, 1 \$/MWh
Coefficient c G1, G2, G3	150 \$/h, 600 \$/h, 335 \$/h

Table 2. Comparison between analytic algorithm and PSO BESS Sizing.

BESS at Bus	Analytic Algorithm				PSO			
	BESS Size (MW)	G1 after BESS installation (MW)	Losses after BESS installation (MW)	Operation Cost \$/h	BESS Size (MW)	G1 after BESS installation (MW)	Losses after BESS installation (MW)	Operation Cost \$/h
4	71.6	0.005	5.188	4509.03	71.6	0	5.185	4509.03
5	71.6	0.000	5.007	4509.47	47.168	24.275	4.962	4697.16
6	71.6	0.000	5.139	4533.96	36.106	35.494	4.986	4852.65
7	0	71.6	5.190	5434.66	0	71.6	5.190	5434.66
8	0	71.6	5.190	5434.66	0	71.6	5.190	5434.66
9	0	71.6	5.190	5434.66	0	71.6	5.190	5434.66

By simulating the IEEE 9-bus system using BESS, the algorithms are used to determine the location of BESS placement. This was repeated for all the buses and the simulation results by both PSO and analytic algorithm were compared as shown in Table 2. The optimum sizing of BESS is designed based on reduction of power losses of the system in DPL scripts. It can be observed that in analytic method, the system has lowest power losses of 5.007 MW with BESS of size 71.6 MW at bus 5, which has the highest load of 125 MW. In PSO, the system has lowest power losses of 4.962 MW with BESS size of 47.168 MW in bus 5 as well. The power losses before the implementation of algorithms were 5.190 MW and reduced to 5.007 MW and 4.962 MW respectively after the algorithms were used by BESS placement at bus 5. Hence the optimal placement of BESS is at bus 5 with lowest power losses achieved.

In analytic algorithm, the BESS size determined are similar to reference generator due to the condition set in the script where P_{BESS} is in between 0 and $P_{grenref}$. As mentioned in methodology where analytic algorithm is a simple calculation method, the BESS size will be obtained nearest to active power of reference generator in order to quit the condition checking loop. PSO determines the fitness value by obeying to the constraints declared. Each particle will explore and find best solution which results to obtaining minimal power losses and active power reduction of reference generator with optimal BESS size. Figure 5 presents the convergence rate of the PSO BESS sizing. The algorithm is converged to final fitness value (power losses) approximately after 6 iterations. It is shown that attainment is obtained with Gbest after 6th iterations. As compared to analytic algorithm, PSO can improve fast to minimize the power losses of the system and achieve optimal BESS size.
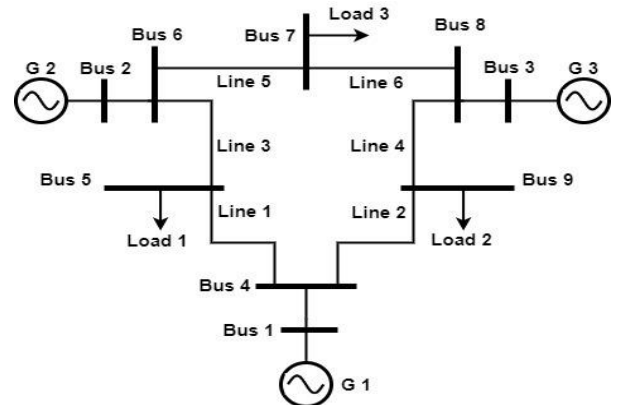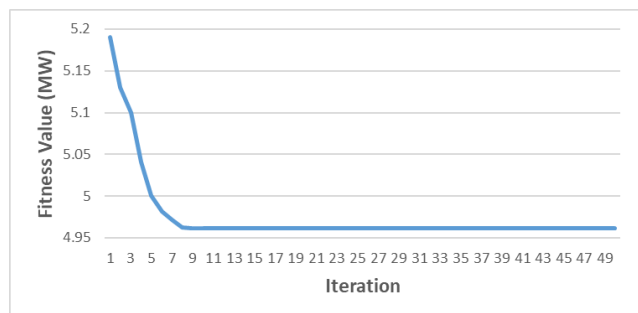

Fig. 4. IEEE 9-bus system single line diagram.

Fig. 5. Convergence rate of the PSO BESS sizing at bus 5.

Table 3. Buses voltage in p.u without BESS and with BESS installed each bus.

Bus Number	Without BESS (p. u.)	Voltage with BESS at Bus 4 (p. u.)	Voltage with BESS at Bus 5 (p. u.)	Voltage with BESS at Bus 6 (p. u.)	Voltage with BESS at Bus 7 (p. u.)	Voltage with BESS at Bus 8 (p. u.)	Voltage with BESS at Bus 9 (p. u.)
1	1.040	1.040	1.040	1.040	1.04	1.04	1.04
2	1.025	1.025	1.025	1.025	1.025	1.025	1.025
3	1.025	1.025	1.025	1.025	1.025	1.025	1.025
4	1.025	1.00	1.018	1.015	1.025	1.025	1.025
5	1.012	0.993	1.00	0.991	1.0126	1.0126	1.0126
6	1.032	1.027	1.029	1.00	1.032	1.032	1.032
7	1.015	1.010	1.013	0.996	1.015	1.015	1.015
8	1.025	1.020	1.023	1.015	1.025	1.025	1.025
9	0.995	0.975	0.999	0.982	0.995	0.995	0.995

The operation cost in both algorithms is affected by the G1 generator where it is the reference generator for the study while the remaining generators are acted as constant power supply. It can be seen that the operation cost is almost the same in both analytic algorithm and PSO except when the BESS is installed in bus 4 and bus 5. This is because G1 is still delivering some portion of power to the system when the BESS is installed in those buses. The operating cost when BESS is installed in bus 4 is 4509.47 $/h for the analytic algorithm and 4697.16 $/h when PSO is applied. However, when the BESS is installed in bus 5 the operating cost is 4533.96 $/h after applying the analytical method and 4852.65 $/h after the PSO method is used.

The voltage in p.u of each bus in the system after installing the BESS is show in Table 3. The voltage of the buses 1, 2 and 3 remains unchanged since those buses are the generators buses. The other buses voltages are changed after the injection of the BESS to the system. It can be observed that when the BESS is installed in bus 5 the voltage of other buses in the system is considered better then when the BESS installed in other buses.

4. Conclusion

This paper presents BESS optimal sizing approaches in DIgSILENT PowerFactory DPL script with PSO and analytic algorithm. The optimization functions and various operations of the approach can be performed with the communication between DPL script and elements modelled in DIgSILENT PowerFactory. The simulations show that the optimal sizing of BESS with PSO gives better results than analytic algorithm. Besides, the PSO algorithm in DPL script can achieve and guarantee optimal size of BESS in DIgSILENT PowerFactory. It is evident that the PSO algorithm in DPL script provides faster and better results than the analytic algorithm. From the presented case study, the optimal placement of BESS is identified at bus 5 with the lowest power losses of 4.962 MW optimal BESS size of 47.168 MW. The operation cost for both algorithms is calculated in the DPL script. The operation cost for analytic and PSO algorithms with optimal BESS installed at bus 5 are 4533.96 $/h and 4852.65 $/h, respectively. Finally, this paper serves as a reference source for researchers to study and implement optimization coding in DPL script.

Acknowledgements

References

[1] M. P. Hajiabbas and B. Mohammadi-Ivatloo, Optimization of power system problems: methods, algorithms and MATLAB codes, vol. 262. Springer Nature, 2020.

[2] J. Wang, X. Qi, F. Ren, G. Zhang, and J. Wang, "Optimal design of hybrid combined cooling, heating and power systems considering the uncertainties of load demands and renewable energy sources," J. Clean. Prod., vol. 281, p. 125357, 2021.

[3] L. Gan, P. Jiang, B. Lev, and X. Zhou, "Balancing of supply and demand of renewable energy power system: A review and bibliometric analysis," Sustain. Futur., vol. 2, p. 100013, 2020.

[4] E. Naderi, M. Pourakbari-Kasmaei, and H. Abdi, "An efficient particle swarm optimization algorithm to solve optimal power flow problem integrated with FACTS devices," Appl. Soft Comput., vol. 80, pp. 243–262, 2019.

[5] Z. Ullah, S. Wang, G. Wu, H. M. Hasanien, M. W. Jabbar, H. S. Qazi, M. Tostado-Véliz, R. A. Turky, M. R. Elkadeem, "Advanced studies for probabilistic optimal power flow in active distribution networks: A scientometric review," IET Gener. Transm. Distrib., vol. 16, no. 18, pp. 3579-3604, 2022.

[6] S. Twaha and M. A. M. Ramli, "A review of optimization approaches for hybrid distributed energy generation systems: Off-grid and grid-connected systems," Sustain. Cities Soc., vol. 41, pp. 320–331, 2018.

[7] A. Oymak and M. R. Tur, "A Short Review on the Optimization Methods Using for Distributed Generation Planning," Int. J. Smart Grid-ijSmartGrid, vol. 6, no. 3, pp. 54–64, 2022.

[8] K. Anoune, M. Bouya, A. Astito, and A. Ben Abdellah, "Sizing methods and optimization techniques for PV-wind based hybrid renewable energy system: A review," Renew. Sustain. Energy Rev., vol. 93, pp. 652–673, 2018.

[9] A. Swarnkar and A. Swarnkar, "Artificial intelligence based optimization techniques: A review," Intell. Comput. Tech. Smart Energy Syst., pp. 95–103, 2020.

[10] T. Kerdphol, Y. Qudaih, and Y. Mitani, "Battery energy storage system size optimization in microgrid using particle swarm optimization," in IEEE PES Innovative Smart Grid Technologies, Europe, pp. 1–6, 2014.

[11] P. Zamani-Dehkordi, H. Chitsaz, L. Rakai, and H. Zareipour, "A price signal prediction method for energy arbitrage scheduling of energy storage systems," Int. J. Electr. Power Energy Syst., vol. 122, p. 106122, 2020.

[12] T. Terlouw, T. AlSkaif, C. Bauer, and W. van Sark, "Multi-objective optimization of energy arbitrage in community energy storage systems using different battery technologies," Appl. Energy, vol. 239, pp. 356-372, 2019.

[13] O. P. Mahela and A. G. Shaik, "Power quality improvement in distribution network using DSTATCOM with battery energy storage system," Int. J. Electr. Power Energy Syst., vol. 83, pp.229-240, 2016.

[14] F. Mohamad, J. Teh, C. M. Lai, and L. R. Chen, "Development of energy storage systems for power network reliability: A review," Energies. vol. 11, no. 9, 2018.

[15] S. Singh, S. K. Singh, S. Chanana, and Y. P. Singh, "Frequency regulation of an isolated hybrid power system with Battery energy storage system," in IEEE International Conference on Power, Instrumentation, Control and Computing (PICC), pp. 1-6, 2014.

[16] W. H. Tee, C. K. Gan, J. B. Sardi, K. A. Baharin, and K. K. Kong, "Probabilistic sizing of battery energy storage system for solar photovoltaic output smoothing," in IEEE International Conference on Power and Energy (PECon), pp.350-355, 2020.

[17] M. Shamshiri, C. K. Gan, J. Sardi, M. T. Au, and W. H. Tee, "Design of Battery Storage System for Malaysia Low Voltage Distribution Network with the Presence of Residential Solar Photovoltaic System," Energies , vol. 13, no. 18. 2020.

[18] L. Zhou, Y. Zhang, X. Lin, C. Li, Z. Cai, and P. Yang, "Optimal sizing of PV and BESS for a smart household considering different price mechanisms," IEEE access, vol. 6, pp. 41050–41059, 2018.

[19] M. R. Jannesar, A. Sedighi, M. Savaghebi, and J. M. Guerrero, "Optimal placement, sizing, and daily charge/discharge of battery energy storage in low voltage distribution network with high photovoltaic penetration," Appl. Energy, vol. 226,

pp. 957–966, 2018.

[20] K. Garifi, K. Baker, B. Touri, and D. Christensen, "Stochastic model predictive control for demand response in a home energy management system," in 2018 IEEE power & energy society general meeting (PESGM), pp. 1–5. 2018.

[21] A. Latif, M. Shahzad, P. Palensky, and W. Gawlik, "An alternate PowerFactory Matlab coupling approach," in 2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST), pp. 486–491, 2015.

[22] Y. Zhu, C. Liu, R. Dai, G. Liu, and Y. Xu, "Optimal battery energy storage placement for transient voltage stability enhancement," in 2019 IEEE Power & Energy Society General Meeting (PESGM), pp. 1–5, 2019.

[23] K. Johnstone, S. M. Blair, M. H. Syed, A. Emhemed, G. M. Burt, and T. I. Strasser, "A co-simulation approach using powerfactory and matlab/simulink to enable validation of distributed control concepts within future power systems," in 24th International Conference on Electricity Distribution, 2017.

[24] DIgSILENT, "PowerFactory 15 User Manual DIG SILENT PowerFactory," 2019. https://www.academia.edu/40677599/PowerFactory_15_User_Manual_DIG_SILENT_PowerFactory (accessed Aug. 11, 2022).

[25] T. Kerdphol, Y. Qudaih, and Y. Mitani, "Optimum battery energy storage system using PSO considering dynamic demand response for microgrids," Int. J. Electr. Power Energy Syst., vol. 83, pp. 58–66, 2016.

[26] M. Talamonti, M. Galluzzo, A. Chiricozzi, P. Quaglino, G. Fabbrocini, P. Gisondi, A. V. Marzano, C. Potenza, A. Conti, A. Parodi, and S. Piaserico, "Characteristic of chronic plaque psoriasis patients treated with biologics in Italy during the COVID-19 Pandemic: Risk analysis from the PSO-BIO-COVID observational study," Expert Opin. Biol. Ther., vol. 21, no. 2, pp. 271–277, 2021.

[27] S. M. H. Mousakazemi and N. Ayoobian, "Robust tuned PID controller with PSO based on two-point kinetic model and adaptive disturbance rejection for a PWR-type reactor," Prog. Nucl. Energy, vol. 111, pp. 183–194, 2019.

[28] X. Chen and G. Tang, "Solving static and dynamic multi-area economic dispatch problems using an improved competitive swarm optimization algorithm," Energy, vol. 238, p. 122035, 2022.

[29] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," IEEE Trans. Power Syst., vol. 26, no. 1, pp. 12–19, 2011.