Solving Combined Sizing and Dispatch of PV and Battery Storage for a Microgrid using ADMM

Robert Steven,a Oleksiy Klymenko,a Michael Shorta,b\*

aSchool of Chemistry and Chemical Engineering, University of Surrey, Guildford, Surrey, GU2 7XH, UK.

bInstitute for Sustainability, University of Surrey, Guildford, Surrey, GU2 7XH, UK.

\*m.short@surrey.ac.uk

Abstract

This paper presents results for solving a mixed-integer linear programming-based optimal microgrid design problem in a distributed fashion using the alternating direction method of multipliers (ADMM). A 7-bus network, with options for installing photovoltaic panels, boilers and battery storage, is considered. A representative-season, multi-year formulation is presented and, when solved in a distributed fashion, was decomposed temporally. It is shown that when a sufficiently large number of years are considered, assuming the ADMM iterations are computed in parallel, then the overall solve time can be less than that of the same problem solved in a centralised manner, whilst still producing near-optimal results. When considering a time period of 50 years, a reduction of 43 % in overall solve time is achieved. This highlights the potential use of this technique to enable the design of microgrids at scales where centralised methods may struggle.

**Keywords**: Distributed Energy Systems, Renewable Energy, Distributed Optimisation, ADMM

* 1. Introduction

To meet the pressing need of net-zero carbon emissions, electrical grids have been undergoing a recent radical process of decarbonisation. One of the keys to enabling this has been distributed energy resources such as photovoltaic (PV) panels and battery storage which can be co-located at electrical loads to form connected or islanded microgrids.

The optimal design problem of determining the optimal placement and sizing of these assets to jointly minimise the capital (CAPEX) and operational (OPEX) expenditure, over a fixed time period, can be formulated as a mixed-integer linear programming (MILP) problem. Problems of this form can be decomposed and solved in a distributed manner using ADMM, described by Boyd et al. (2011) as a means of solving convex optimisation problems containing separable objective functions, using a “decomposition-coordination” process. The objective terms for the original problem are decomposed into smaller subproblems, solved in parallel, with coordination steps to ensure a uniform overall solution (Boyd et al., 2011). Whilst a well-known drawback of ADMM is its slow convergence rate (Boyd et al., 2011), with a sufficient level of parallelisation, the overall solve time for other problem formulations using ADMM has been shown to be reduced below that of the equivalent centralised problem formulation (Guo et al., 2017). The literature shows that ADMM has been applied extensively to problems relating to microgrids, such as energy trading (Paudel and Gooi, 2019), economic dispatch (Chen and Yang, 2018) and optimal power flow (Wang et al., 2020) but has not, to the authors knowledge, previously been used for microgrid design problems. This is further evidenced by the fact that design problems are not referenced as a considered application area in a recent survey examining the use of ADMM in smart power grids (Maneesha and Swarup, 2021).

* 1. Methodology
     1. Main Problem Formulation

An MILP formulation is used, employing the modelling elements presented by De Mel et al. (2024). These include PV panels, boilers and battery storage, as well as electrical and heat energy balance. To preserve the linearity of the formulation, the DC-OPF power flow approximation (Frank and Rebennack, 2016) was used. The original formulation used a representative seasonal approach, whereby a single 24-hour period of averaged electrical and heat load, with a 1-hour temporal resolution, was considered for each season (De Mel et al., 2024). This was extended to multiple years by adding additional linking constraints for the three decision variables, shown below in Eq. (4). The objective function for each individual season is defined as the total seasonal cost (TSC) in Eq. (1).

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

The terms in Eq. (1) include the operational costs for the PV, boiler and battery technologies considered here, as well as the cost of purchasing power from the main grid (COPEX,grid), income from selling excess power back to the grid (Cexport,income) and income from a feed-in-tariff scheme (CFIT,income). The reader is encouraged to refer to De Mel et al. (2024) for further details.

Only a single investment, made at the start of the overall time period, is considered, and so the overall objective function, the total cost (TC), becomes the sum of this CAPEX and the TSC across all seasons and years, defined in Eq. (2).

|  |  |
| --- | --- |
|  | (2) |

The subscript *m* denotes the time period index, with ***M*** being the set of all considered time periods, namely 4 seasons in each of the considered years. Extending the considered time period here allows the model to take into account trends and changes in parameters such as electricity pricing, grid carbon intensity and electrical load.

* + 1. Problem Decomposition

The problem formulation is decomposed temporally, with a single subproblem for each *m*, using the consensus version of ADMM (Boyd et al., 2011). The consensus formulation given by Boyd et al. (2011) is as follows:

|  |  |
| --- | --- |
| s.t. | (3) |

For this problem, the consensus variables are given as:

|  |  |
| --- | --- |
|  | (4) |

Where *n*panel,PV is the number of PV panels on each roof, *H*gen,max is the maximum boiler heat generation and *Cap*batt is the installed battery capacity. Here, *x*m defines the copy of the consensus variables in each subproblem *m* whilst *z* is defined globally as the “central collector” (Boyd et al., 2011). The addition of the consensus constraint, shown generally in Eq. (3) and specifically for this problem in Eq. (4), gives the formulation a separable objective function, tied together with complicating constraints, suitable for solving using ADMM (Boyd et al., 2011).

The augmented Lagrangian for the problem then becomes:

|  |  |
| --- | --- |
|  | (5) |

Here, *y*m is the Lagrange multiplier associated with the consensus constraint and *ρ* is the ADMM penalty parameter (Boyd et al., 2011). The reader is referred to Boyd et al. (2011) for the corresponding ADMM algorithmic steps. These steps are carried out iteratively until either the maximum number of iterations is reached or convergence is declared by reducing the primal gap below the given threshold. The threshold is defined as *ε*, with the corresponding primal gap convergence criteria across all subproblems for iteration *k*:

|  |  |
| --- | --- |
|  | (6) |

* + 1. Test Network

The network used for the results presented here is the 7-bus network from De Mel et al. (2022). Data for only a single year is provided, to extrapolate this to multiple years the electrical load, heat load and irradiance values were multiplied by a uniformly distributed random value between 0.5 and 1.5. Whilst this is not expected to be representative of a real-world scenario, it provides appropriate datasets for each considered year.

* + 1. Experiments

The performance of ADMM in terms of convergence is known to depend on the chosen value of the penalty parameter, *ρ*, that is used (Boyd et al., 2011). Here, we start with an exploratory set of computational experiments with values of *ρ* between 1000 and 0.01, number of years considered between 1 and 50 and note the results for required primal gap convergence thresholds, *ε*, of 1, 0.1 and 0.01. The problem was also solved in a centralised manner and CPU solve times for both centralised and ADMM formulations were recorded, along with optimality gaps. The optimality gap here is simply taken as the difference between the overall objective value of the two formulations:

|  |  |
| --- | --- |
|  | (7) |

Here, a positive value for the optimality gap indicates that the ADMM formulation returned a sub-optimal result.

To determine the solve time for the ADMM solution, it is assumed that each of the subproblems are being solved in parallel (the *x*k+1 update step in Boyd et al. (2011)) and the time taken for any communication of the *x*k+1 values, as well as the updates for *z* and *y* are negligible. As this is a synchronous implementation, the total solve time for the ADMM solution is taken as the maximum *x*k+1 solve time for all subproblems in each iteration, summed over all iterations. The solve times are presented as a ratio, where a ratio value larger than 1 indicates that the ADMM formulation took longer to solve compared to the centralised formulation and vice versa.

* + 1. Implementation

All problems were implemented with the Julia programming language (Bezanson et al., 2017) using the JuMP modelling language (Lubin et al., 2023) and solved using the Gurobi solver (Gurobi, 2023). All results were computed on a Dell Latitude 7420 laptop with an Intel i5-1135G7 processor running at 2.40 GHz and 16 GB of RAM.

* 1. Results
     1. Initial Centralised Results

The objective value and CPU solve time for the centralised formulation are shown in Figure 1.

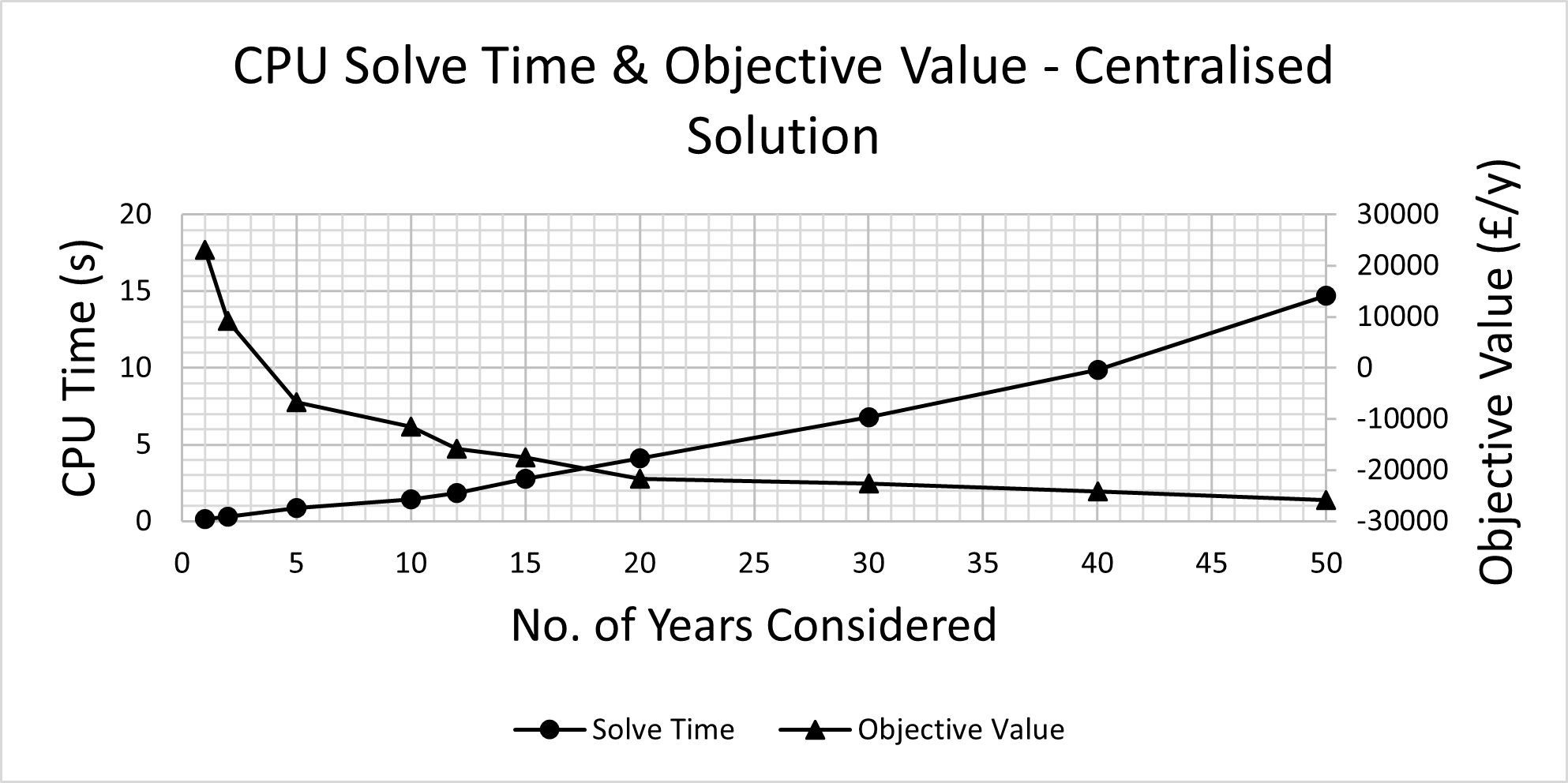


Figure 1: Centralised Solution CPU Solve Time and Objective Value

The objective value here is presented as the overall cost (including CAPEX, OPEX and income), normalised by the number of years considered, to give the total cost per year.

These results correspond to the following values for the decision variables highlighted in the previous section, a subset of which are shown below in Table 1.

Table 1 Optimal Decision Variables for 1 and 50 Years

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 year | | | 50 years | | |
| Load | No. of PV Panels | Max Heat Gen (kWh) | Battery Capacity (kWh) | No. of PV Panels | Max Heat Gen (kWh) | Battery Capacity (kWh) |
| 1 | 85 | 6.67 | 0.00 | 85 | 14.85 | 7.11 |
| 2 | 400 | 31.84 | 0.00 | 400 | 70.87 | 33.60 |
| 3 | 342 | 18.10 | 0.00 | 342 | 40.30 | 25.62 |
| 4 | 85 | 4.25 | 0.00 | 85 | 9.47 | 6.79 |
| 5 | 314 | 20.96 | 0.00 | 314 | 46.65 | 22.16 |

The number of installed PV panels selected remains the same for both the minimum and maximum number of years considered, this number is the maximum possible number of panels for the given load roof areas. Batteries on the other hand are only installed when the investment cost is spread over more years and is outweighed by export income.

* + 1. ρ Value Sweep

The following plot shows the optimality gap for the ADMM formulation and CPU solve time ratio, for a primal gap convergence threshold of 1. From the plot in Figure 2, a value of *ρ*=0.1 is selected as an optimal trade-off between both the optimality gap and the CPU solve time ratio.

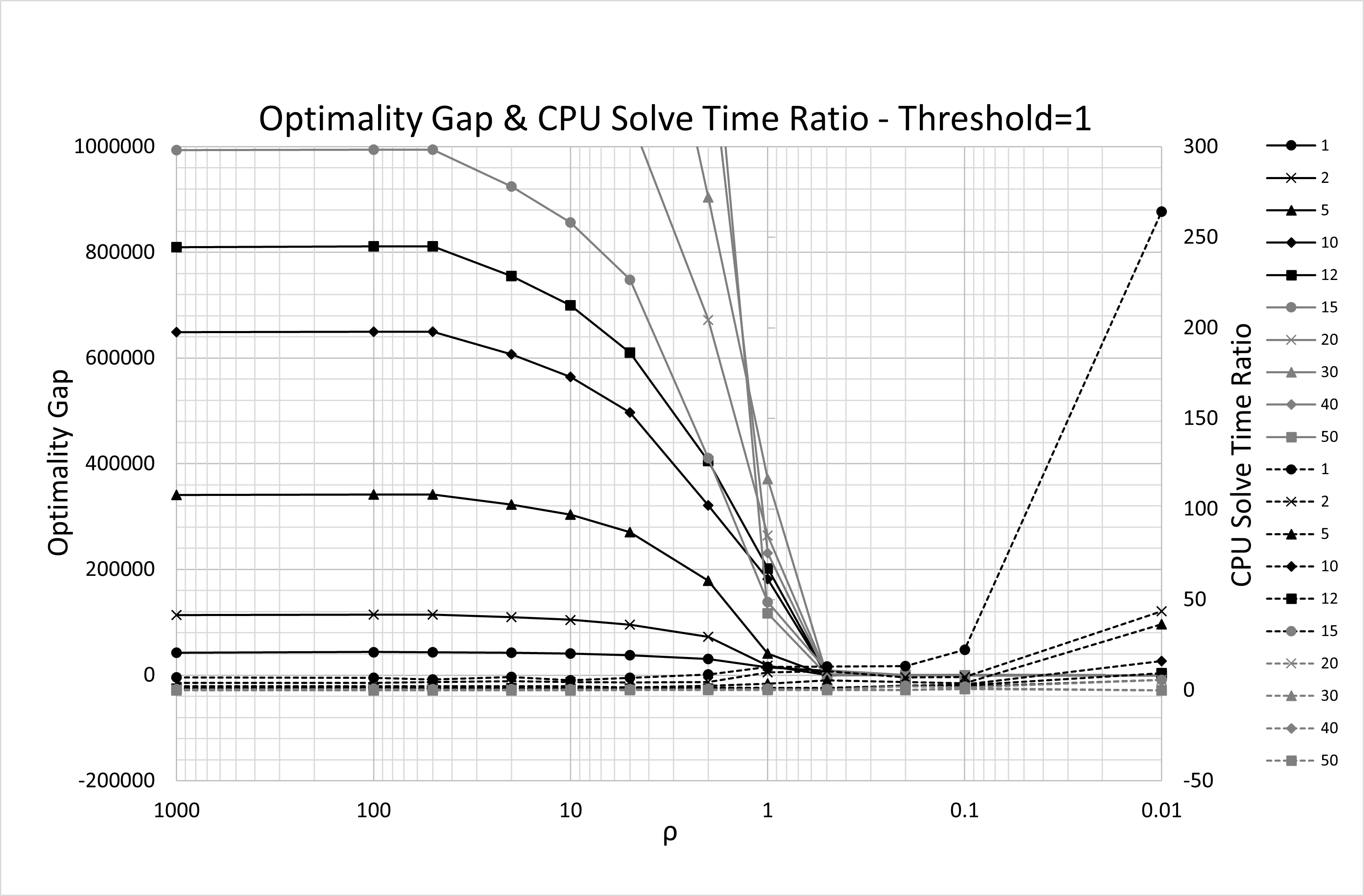


Figure 2: *ρ* Value Sweep – Optimality Gap (Solid Line) & CPU Solve Time Ratio (Dashed Line)

* + 1. Performance Across No. of Considered Years

The plot shown in Figure 3 gives the performance of the ADMM formulation with *ρ*=0.1 in terms of CPU solve time ratio and optimality gap, considering a number of years between 1 and 50. The final point for 50 years and a threshold of 0.01 is omitted as ADMM did not converge within the iteration limit.

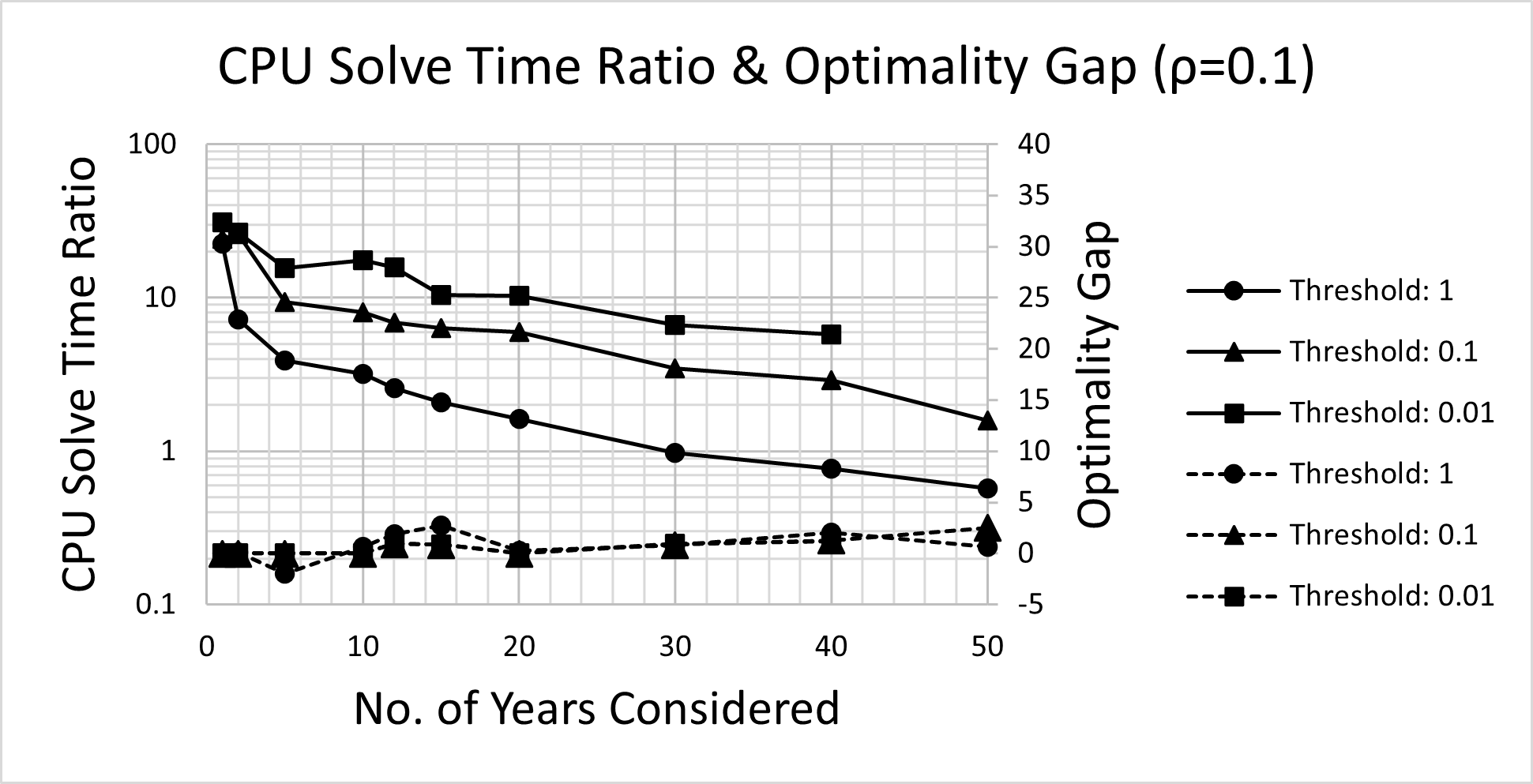


Figure 3: CPU Solve Time Ratio (Solid Line) and Optimality Gap (Dashed Line) for ρ =0.1

* 1. Discussion

In the initial results in Figure 1 and Table 1, whilst the number of PV panels remains constant, battery storage is only selected for a longer considered time period, reflecting the larger investment that this entails. Due to the ability to sell energy back to the grid, objective values become increasingly attractive for longer timescales, paying back the up-front investment costs in PV and batteries. The time taken to solve the formulation in a centralised manner scales with the problem size, from <1 CPUs to 14 CPUs for 50 years.

For the ADMM formulation, when sweeping the value of ρ, a clear region exists affording both a low optimality gap, which increases if ρ is set too high, and a minimal CPU solve time ratio, which increases if ρ is set too low. With *ρ* set to 0.1, the plot in Figure 3 demonstrates a low optimality gap, implying that the solution mimics that of the centralised formulation, with a decreasing CPU solve time ratio as more years are considered. For a primal gap convergence threshold of 1, there is a “break-even” point at 30 years, when considering more years than this, the overall solve time for the ADMM solution is lower than that of the centralised solution when decomposed temporally into each considered time period. This demonstrates the enhanced ability of the ADMM formulation to provide solutions in a timely manner when considering more time periods.

* 1. Conclusions

The results presented here highlight the potential for ADMM to speed-up the solving of MILP microgrid design problems, whilst still providing results that closely match those calculated in a centralised fashion. When considered over 50 years, use of the ADMM formulation resulted in a reduction in CPU solve time of 43 %, assuming parallel computation. Care must be taken however, as these time savings from use of the parallel formulation are only seen with sufficiently large problem sizes.

Further work will involve adding additional objective function terms, such as asset lifetimes and associated maintenance and disposal costs, and adding the ability for capital expenditure to be made at different points throughout the considered time period. Testing with the larger Modified IEEE EU LV Network and using the AC-OPF power flow formulation, as presented in De Mel et al. (2024) will also be completed. The formulation presented above can be extended to include elements such as heat pumps, electric vehicle charging, and demand response and further research will also look at incorporating spatial decomposition into the formulation and comparing the results obtained from ADMM with other decomposition methods.

References

J. Bezanson, A. Edelman, S. Karpinski and V. B. Shah, 2017, Julia: A fresh approach to numerical computing, SIAM review, 59, 1, 65-98

S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, 2011, Distributed optimisation and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning, 3, 1, 1-122

G. Chen and Q. Yang, 2018, An ADMM-Based Distributed Algorithm for Economic Dispatch in Islanded Microgrids, IEEE Transactions on Industrial Informatics, 14, 9, 3892-3903

I. De Mel, O. Klymenko and M. Short, 2022, Optimal Design of Distributed Energy Systems Considering the Impacts on Electrical Power Networks, arxiv

I. De Mel, O. V. Klymenko and M. Short, 2024, Complementarity reformulations for the optimal design of distributed energy systems with multiphase optimal power flow, International Journal of Electrical Power & Energy Systems, 155, B, 109610

S. Frank and S. Rebennack, 2016, An introduction to optimal power flow: Theory, formulation, and examples, 48, 12, 1172-1197

J. Guo, G. Hug and O. K. Tonguz, 2017, A Case for Nonconvex Distributed Optimization in Large-Scale Power Systems, IEEE Transactions on Power Systems, 32, 5, 3842-3851

Gurobi Optimization, LLC, 2023, Gurobi Optimizer Reference Manual

M. Lubin, O. Dowson, J. Dias Garcia, J. Huchette, B. Legat and J. P. Vielma, 2023, JuMP 1.0: Recent improvements to a modeling language for mathematical optimization, Mathematical Programming Computation

A. Maneesha and K. S. Swarup, 2021, A survey on applications of Alternating Direction Method of Multipliers in smart power grids, Renewable and Sustainable Energy Reviews, 152, 111687

A. Paudel and H. B. Gooi, 2019, Pricing in Peer-to-Peer Energy Trading Using Distributed Optimization Approach, 2019 IEEE Power & Energy Society General Meeting (PESGM)

Y. Wang, T. L. Nguyen, C. Ju, Y. Xu, B. Wang and X. Feng, 2020, A Distributed Secondary-Tertiary Coordinated Control Framework for Islanded Microgrids, 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), 1, 603-608