A Consumer Appliance-Level Framework for Optimized Load Shedding

Christian. Y. Cahig ✉, Abdul Aziz G. Mabaning

Updated as of Thursday 22nd October, 2020 14:47.

***Abstract*—Load shedding can be the most viable solution for a utility in scenarios where the balance between power supply and demand is at risk. As it directly tampers a consumer's access to amenities and benefits, load shedding has to be an optimized process. This paper retells a recently proposed framework where the smallest sheddable unit of load is a consumer appliance. The value proposition of such setup is two-fold: the consumer can avoid total interruption of comfort and productivity, and the utility can maximize the utilization of available supply. We also critique the orignally proposed framework and identify three gaps regarding the optimization problem formulation, the solution method, and the case study scenarios. We propose methodologies to address the said gaps and present experiments on synthetic datasets to illustrate our concerns. Lastly, we make this work entirely available to the public.**

***Index Terms*—Optimization, power systems, optimal load shedding.**

I. Introduction

Load shedding is a course of action a utility company can take when the balance between power supply and demand is at risk [1], or when the system voltage or frequency exceeds its allowable lower limit for more than a minute [2]. Compared to investing on the installation of generating capacities to curb threats on the supply-demand balance, load shedding is more economical and realistic in short-term considerations.

Since this strategy directly deals with the customers, load shedding has to be an optimized process. As such, one can find numerous works in literature (*e.g.*, fast load shedding [3], clustering-based load shedding [4], and load shedding during disaster events [5]) on feeder-level implementations. In this setting, the smallest "sheddable" unit of load is the total load connected to a feeder. This coarse treatment is prone to excessive shedding of load and an unoptimized utilization of available supply. Accordingly, we see works on residence-level implementation: *e.g.*, using smart meters, single-board computers, server monitoring [6], detecting rate-of-change of frequency [7], and using a mobile distribution-level PMU [8].

Moving forward in this trend, [9] proposed a shedding framework where the smallest "sheddable" unit of load is a consumer appliance. This granular scheme addresses the concern of excessive load dropping in a feeder-level implementation, as well as improves the operation of a residence-level setup. Keeping both the utility and the customer in the framework means both parties can make the most of the situation in their perspectives. By having the ability to prioritize appliances according to personal judgment, a consumer can avoid total interruption of comfort and/or productivity within his/her premises. On the other hand, the utility can maximize the utilization of available power supply.

This work is a reimplementation of the framework originally proposed by [9] and is part of the curricular requirements of an academic course on Optimization in Power Systems. Although we are building on the basic formulation in [9], this present work differs in a number of ways. First, we critique the original work and emphasize some methodological rooms for improvement. Second, as the dataset used in the original study is undisclosed, we develop synthetic datasets based on the well-known open-sourced test cases (*e.g.*, IEEE 37-Node Test Feeder [10]). Third, we consider different solution strategies, in addition to the heuristics-based approach in [9], for the core optimization problem. Lastly, to promote reproducibility, we make our data files, source codes, and documentation available to the public.

The remainder of this paper proceeds as follows. Section II presents the consumer appliance-level framework and formulates the core optimization problem. In Section III, we discuss a critique of the original work of [9] and propose some methods to address the identified issues. Section IV details the procedure, analyses, and benchmarks for the datasets developed. Section V concludes the work.

II. Basic Problem Formulation

This section expounds the load shedding framework whose core notion is that a consumer appliance is the smallest "sheddable" unit of load. We first present in Section II-A how the distribution system is modelled in terms of how the utility and the consumers participate in the framework. We then describe in Section II-B the process of nominating appliances as loads to be shed, which involves two optimization tasks: one on the consumer's side and one on the utility's. The consumer-side problem is formulated in Section II-C and the utility-side in Section II-D. More importanty, we remind the reader that this framework is orginally proposed by [9], and that this section is merely a retelling of their novelty.

A. System Configuration

Consider a distribution system with C loads (see Fig. 1), where each load c represents a consumer (*i.e.*, a registered customer of the utility). We assume a communication network that is not wholly centralized: (i) appliances within the premises

The authors are with the Department of Electrical Engineering and Technology, Mindanao State University - Iligan Institute of Technology, Iligan City, Philippines.

✉ christian.cahig@g.msuiit.edu.ph

Project page: https://github.com/christian-cahig/CALOLS

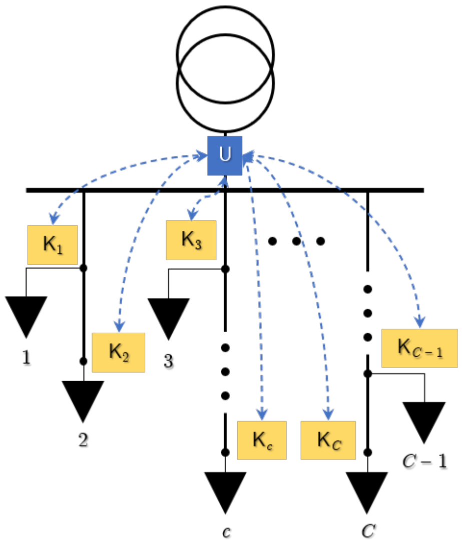

Fig. 1: System single line diagram and network topology for the consumer appliance-level load shedding. Although the figure depicts one load at each node for brevity, the framework presented is not limited to three-phase loads.

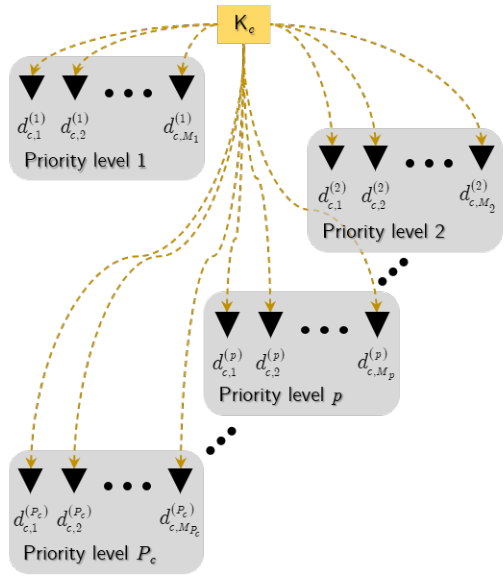

Fig. 2: Grouping of appliances into priority levels for consumer c.

of consumer c are monitored and controlled by consumer node K_c; (ii) all consumer nodes communicate with central node U in the utility side (*e.g.*, housed in a substation); and (iii) communication between K_c's and U have a bidirectional protocol.

For the appliance-level shedding to be possible, a consumer c assigns P_c priority levels, according to his/her preference and prerogative (see Fig. 2). In this case, we take $p = 1$ as the highest priority of being supplied—*i.e.*, appliances within this priority level are to be supplied first when possible—and $p = P_c$ as the least priority. There are M_p appliances within level p, and each of them are denoted by $d_{c,i}^{(p)}$. Without introducing another variable, let us use $d_{c,i}^{(p)}$ as the power rating of that appliance. We can then represent the appliances in level p by a vector of its power ratings $\mathbf{d}_c^{(p)}$ and the appliances within consumer c as a vector $\mathbf{d}_c$, that is,

$$\mathbf{d}_c^{(p)} = \left[d_{c,1}^{(p)}, d_{c,2}^{(p)}, \ldots, d_{c,M_p}^{(p)}\right]^\intercal \in \mathbb{R}^{M_p} \tag{1}$$

$$\mathbf{d}_c = \left[\mathbf{d}_c^{(1)}, \mathbf{d}_c^{(2)}, \ldots, \mathbf{d}_c^{(P_c)}\right]^\intercal \in \mathbb{R}^{N_c} \tag{2}$$

where $N_c = \sum_{i=1}^{P_c} M_i$ is the total number of appliances in c. From (1)-(2), it follows that the active power ratings of the appliances add up to

$$\sum_{p=1}^{P_c} \sum_{i=1}^{M_p} d_{c,i}^{(p)} \tag{3}$$

B. Nomination of Appliances for Load Shedding

In the event where load shedding has to be implemented, the utility has to determine how much load needs to be shed. If the expected available supply for the system during the impending shedding period is S_{sys}, and the expected total load during the same period is L_{sys}, then we can define a *reduction ratio* ρ as

$$\rho = \frac{S_{\text{sys}}}{L_{\text{sys}}} \tag{4}$$

Assuming a load shedding scenario where S_{sys} may not meet L_{sys}, ρ is the fraction of L_{sys} that must be shed. This means that a consumer has to adjust his/her consumption by $\rho - \mu$, where μ is a correcting margin[1] determined by the utility to compensate for load forecast errors. Both ρ and μ are then communicated by U to all K_c's.

For a consumer c, whose total consumption at the moment K_c receives ρ and μ, is L_c, his/her expected supply for the shedding period, S_c, is given by

$$S_c = (\rho - \mu)\, L_c \tag{5}$$

K_c has to maximize the utilization of S_c among all appliances it oversees according to the priority levels (see Fig. 3). At $p = 1$, the supply $S_c^{(1)} = S_c$ is optimally distributed among appliances $\mathbf{d}_c^{(1)}$. This results in two groups of appliances for $p = 1$: those that are to be supplied ($\mathbf{p}_c^{(1)}$), and those that are nominated for shedding ($\mathbf{n}_c^{(1)}$). Because appliance ratings are fixed and considered indivisible, we can retrieve $\tilde{S}_c^{(1)}$, which is the portion of $S_c^{(1)}$ that is not allocated to power any more appliances at $p = 1$. Then, $\tilde{S}_c^{(1)}$ becomes the supply to be optimally distributed among appliances at $p = 2$ (*i.e.*, $S_c^{(2)} = \tilde{S}_c^{(1)}$) and the process continues through $p = P_c$. The unallocated supply at the lowest priority level is also the unallocated portion of S_c, *i.e.*, $\tilde{S}_c = \tilde{S}_c^{(P_c)}$. We can define a vector for all appliances in c "preliminarily" nominated for shedding:

$$\mathbf{n}_c = \left[\mathbf{n}_c^{(1)}, \mathbf{n}_c^{(2)}, \ldots, \mathbf{n}_c^{(P_c)}\right]^\intercal \tag{6}$$

[1] μ can be set as the maximum forecasting error that is likely to happen, *e.g.*, 95% of errors in the system's forecast records do not exceed this value.

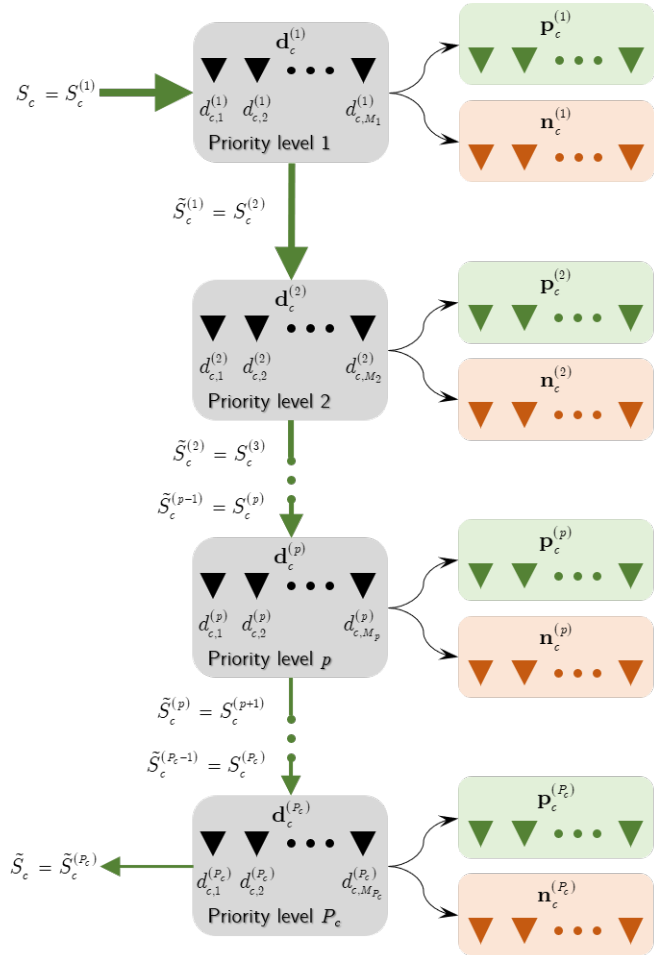

Fig. 3: Utilization of S_c among the appliances of consumer c.

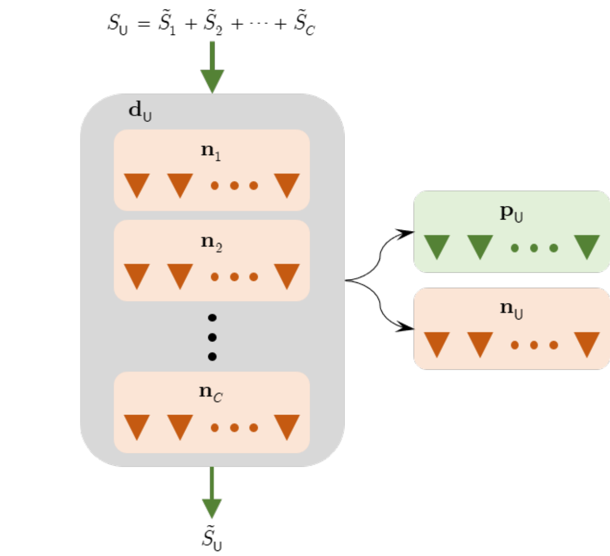

Fig. 4: Utilization of S_{U} among appliances preliminarily nominated by K_c's.

Then, K_c reports $\tilde{S}_c$ and $\mathbf{n}_c$ to U.

Having $\tilde{S}_c$ and $\mathbf{n}_c$, $\forall c = 1, 2, \ldots, C$, we define the total unallocated supply from preliminary nominations S_{U} and the vector of appliances preliminarily nominated for shedding from all consumers $\mathbf{d}_{\mathsf{U}}$ as

$$S_{\mathsf{U}} = \sum_{c=1}^{C} \tilde{S}_c \tag{7}$$

$$\mathbf{d}_{\mathsf{U}} = [\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_C]^{\mathsf{T}} \tag{8}$$

To further maximize utilization, U redistributes S_{U} among $\mathbf{d}_{\mathsf{U}}$ (see Fig. 4). As with the procedure for each consumer priority level, this results in two groups of appliances: $\mathbf{p}_{\mathsf{U}}$ contains those in $\mathbf{d}_{\mathsf{U}}$ that are to be supplied using S_{U}, while $\mathbf{n}_{\mathsf{U}}$ is the final roster of appliances nominated for shedding. The utility relays $\mathbf{p}_{\mathsf{U}}$ (and/or $\mathbf{n}_{\mathsf{U}}$) to all K_c's. Then the actual shedding implementation commences.

C. Consumer-side Optimization

At priority level p, K_c has to allocate $S_c^{(p)}$ among appliances denoted by $\mathbf{d}_c^{(p)}$. Each appliance is a quantized load of fixed amount; hence, K_c can only alter the distribution of $S_c^{(p)}$ by either "picking up" or nominating for shedding an appliance. We denote a binary decision variable k_i for appliance $d_{c,i}^{(p)} \in \mathbf{d}_c^{(p)}$ such that

$$k_i = \begin{cases} 0, & \text{if } d_{c,i}^{(p)} \text{ is nominated for shedding} \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

The amount from $S_c^{(p)}$ that is utilized for picking up appliances is given by

$$\begin{aligned} \sum_{i=1}^{M_p} k_i d_{c,i}^{(p)} &= k_1 d_{c,1}^{(p)} + k_2 d_{c,2}^{(p)} + \ldots + k_{M_p} d_{c,M_p}^{(p)} \\ &= \left(\mathbf{d}_c^{(p)}\right)^{\mathsf{T}} \mathbf{k} \end{aligned} \tag{10}$$

where $\mathbf{k} = \left[k_1, k_2, \ldots, k_{M_p}\right]^{\mathsf{T}}$ is the vector representation of the decision variables. The unallocated portion of $S_c^{(p)}$ is

$$\tilde{S}_c^{(p)} = S_c^{(p)} - \sum_{i=1}^{M_p} k_i d_{c,i}^{(p)} = S_c^{(p)} - \left(\mathbf{d}_c^{(p)}\right)^{\mathsf{T}} \mathbf{k} \tag{11}$$

Hence, optimizing the utilization of $S_c^{(p)}$ corresponds to minimizing $\tilde{S}_c^{(p)}$. In this sense, $S_c^{(p)}$ is a "cost" to be minimized. Moreover, one has to restrict the picking up of appliances so that $\tilde{S}_c^{(p)}$ is nonnegative, *i.e.*,

$$S_c^{(p)} - \left(\mathbf{d}_c^{(p)}\right)^{\mathsf{T}} \mathbf{k} \geq 0 \tag{12}$$

From (9)-(12), the consumer-side optimization problem is

$$\underset{\mathbf{k}}{\text{minimize}} \quad \left\{\tilde{S}_c^{(p)} = S_c^{(p)} - \left(\mathbf{d}_c^{(p)}\right)^{\mathsf{T}} \mathbf{k}\right\} \tag{13a}$$

$$\text{subject to} \quad S_c^{(p)} - \left(\mathbf{d}_c^{(p)}\right)^{\mathsf{T}} \mathbf{k} \geq 0, \tag{13b}$$

$$k_i \in \{0, 1\}, \; i = 1, 2, \ldots, M_p \tag{13c}$$

The solution to (13), $\mathbf{k}^*$, is the switching combinations for appliances $\mathbf{d}_c^{(p)}$ that minimizes $\tilde{S}_c^{(p)}$. Note that (13) is solved at each priority level p for each consumer c.

D. Utility-side Optimization

The distribution of S_{U} among $\mathbf{d}_{\mathsf{U}}$ is similar to (13). Simply put, utility-side optimization seeks to "squeeze" out more utilization of supply with the intuition that $\tilde{S}_c$ may be enough to pick up some appliances in $\mathbf{n}_b$, $b \neq c$.

Each appliance $d_{\mathsf{U},i} \in \mathbf{d}_{\mathsf{U}}$ can only be controlled by U in two ways. We define a binary variable u_i to indicate this decision parameter:

$$u_i = \begin{cases} 0, & \text{if } d_{\mathsf{U},i} \text{ is nominated for shedding} \\ 1, & \text{otherwise} \end{cases} \tag{14}$$

If $\mathbf{d}_{\mathsf{U}}$ has M_{U} appliances, then the amount from S_{U} expended for picking up appliances is given by

$$\begin{aligned} \sum_{i=1}^{M_{\mathsf{U}}} u_i d_{\mathsf{U},i} &= u_1 d_{\mathsf{U},1} + u_2 d_{\mathsf{U},2} + \ldots + u_{M_{\mathsf{U}}} d_{\mathsf{U},M_{\mathsf{U}}} \\ &= (\mathbf{d}_{\mathsf{U}})^{\intercal} \mathbf{u} \end{aligned} \tag{15}$$

where $\mathbf{u}$ is the vector representation of the decision variables, and $\mathbf{u}, \mathbf{d}_{\mathsf{U}} \in \mathbb{R}^{M_{\mathsf{U}}}$. The unallocated portion of S_{U} is

$$\tilde{S}_{\mathsf{U}} = S_{\mathsf{U}} - \sum_{i=1}^{M_{\mathsf{U}}} u_i d_{\mathsf{U},i} = S_{\mathsf{U}} - (\mathbf{d}_{\mathsf{U}})^{\intercal} \mathbf{u} \tag{16}$$

where $\tilde{S}_{\mathsf{U}}$ is to be minimized but cannot be negative:

$$S_{\mathsf{U}} - (\mathbf{d}_{\mathsf{U}})^{\intercal} \mathbf{u} \geq 0 \tag{17}$$

Finally, the utility-side optimization problem is

$$\underset{\mathbf{k}}{\text{minimize}} \quad \left\{ \tilde{S}_{\mathsf{U}} = S_{\mathsf{U}} - (\mathbf{d}_{\mathsf{U}})^{\intercal} \mathbf{u} \right\} \tag{18a}$$

$$\text{subject to} \quad S_{\mathsf{U}} - (\mathbf{d}_{\mathsf{U}})^{\intercal} \mathbf{u} \geq 0, \tag{18b}$$

$$u_i \in \{0, 1\}, \; i = 1, 2, \ldots, M_{\mathsf{U}} \tag{18c}$$

The solution to (18), $\mathbf{u}^*$, is the switching combinations for appliances $\mathbf{d}_{\mathsf{U}}$ that minimizes the $\tilde{S}_{\mathsf{U}}$.

III. A Critique of the Original Work

While the intent of the work by [9] is commendable in adding a demand-response dimension to load shedding, there are certain points in their methodology that we find insufficiently justified (at least as presented in their paper). These issues as well as prospective procedures to address them are discussed in the succeeding subsections.

A. On the Cascading Structure of Consumer-side Optimization

The prioritized utilization of S_c (see Fig. 3) can be recast as a non-cascading procedure. For example, the priority levels of appliances $\mathbf{d}_c = [d_{c,1}, d_{c,2}, \ldots, d_{c,N_c}]^{\intercal}$ can be encoded as a vector of corresponding priority weights $\boldsymbol{\phi}_c = [\phi_{c,1}, \phi_{c,2}, \ldots, \phi_{c,N_c}]^{\intercal}$. In this manner, two appliances $d_{c,i}$ and $d_{c,j}$ are of the same priority levels if $\phi_{c,i} = \phi_{c,j}$. Moreover, the consumer-side optimization (13) need not be performed P_c times, and can be recast as a single-stage minimization of $\tilde{S}_c$:

$$\underset{\mathbf{k}}{\text{maximize}} \quad \left\{ \tilde{S}_c = S_c - (\boldsymbol{\phi}_{\mathbf{c}} \odot \mathbf{d}_c)^{\intercal} \mathbf{k} \right\} \tag{19a}$$

$$\text{subject to} \quad S_c - (\boldsymbol{\phi}_{\mathbf{c}} \odot \mathbf{d}_c)^{\intercal} \mathbf{k} \geq 0, \tag{19b}$$

$$k_i \in \{0, 1\}, \; i = 1, 2, \ldots, N_c \tag{19c}$$

where $\odot$ denotes element-wise multiplication, and $\mathbf{k} = [k_1, k_2, \ldots, k_{N_c}]$ is a modified vector representation of the decision variables corresponding to $\mathbf{d}_c$.

Although there is a burden in properly setting $\boldsymbol{\phi}_{\mathbf{c}}$, it is reasonable to wonder whether the computational burden of per-priority level optimization (13) is significantly less than that of (19) or any variants thereof. A possible approach for setting $\boldsymbol{\phi}_{\mathbf{c}}$ is as follows:

1) Create a histogram of the appliance ratings $\mathbf{d}_c$ where the number of bins is equal to P_c.
2) Suppose that the sums of the appliance ratings within the bins are $b_{c,P_c}, \ldots, b_{c,2}, b_{c,1}$, where $b_{c,1} > b_{c,2} > \ldots > b_{c,P_c}$.
3) The priority weight $\phi_{c,i}$ for appliance $d_{c,i}$ with priority level p would be

$$\phi_{c,i} = \frac{b_{c,p}}{\sum_{j=1}^{N_c} d_{c,j}}$$

This basically weights the appliance rating $d_{c,i}$ with a unitless factor $\phi_{c,i}$ so that the values of the objective and constraint functions in (13) and in (19) have the same physical units.

B. On the Rush to Heuristics

The "rush to heuristics" may have been unjustified at worst and improvable at best. In [9], genetic algorithm (GA) and binary particle swarm optimization (BPSO) with modified initialization schemes were used as the solution method. The consumer-side (13) and utility-side (18) optimization are integer linear programming problems, which can be solved efficiently by properly chosen numerical approaches. This makes the authors' claim of GA being more suitable than BPSO for real-time applications of less confidence. To address this methodological gap, we consider a comparison of GA and BPSO with performant numerical approaches such as linear programming relaxation with Gomory cuts and ℓ_1 regularization.

We also emphasize that the procedure for tuning GA and BPSO parameters (*e.g.*, population size, mutation rate, crossover rate, inertia weight) was not discussed in the original paper. These parameters are known to affect search performance and quality of results. We propose an ablation study of these parameters in order to highlight their effects in solving the optimization tasks.

C. On Case Study Scenarios

Given the implications of future real-life implementation, the case studies should have included more scenarios in addition to having "low" and "high" diversity of appliance ratings. Equally pressing scenarios could be varying number of consumer appliances and fixed/varied number of priority levels. A more serious lapse is that the case studies did not indicate the ground truth optimum values. This raises a valid concern mainly because GA and BPSO (among other evolutionary and population-based heuristics) are known to provide sub-optimal and/or inconsistent results due to reliance on some form of random distribution.

We address this concern by developing datasets from one common test case but with varying N_c and P_c. Since the data from the actual distribution utility used in [9] are undisclosed, we resort to well-known open-sourced power system test cases. We also propose to conduct benchmarks on these datasets considering different metrics (*e.g.*, optimal $\tilde{S}_c^{(p)}$ for all p and c (13) and/or $\tilde{S}_{\mathrm{U}}$ (18) considering various ρ and μ (4)-(5)).

IV. Datasets

Our datasets are based on the IEEE 37-Node Test Feeder[2] (37NTF) [10]. In Section IV-A, we describe the procedure in modifying 37NTF and deriving the datasets. Section IV-B is a summary of quantitative measures of the datasets. Section IV-C discusses the benchmark tests performed on the datasets. Source codes and related files will be available in `Dataset/` within the project repository.

A. Preparation

Being a delta-connected system, 37NTF have point loads drawing unbalanced power across phases A-B, B-C, and C-A at their respective nodes. We treat each phase of a point load in 37NTF as one consumer, provided that it has a non-zero active power value. For example, the load at node "742" which draws active powers of 8 kW across A-B and 85 kW across B-C is encoded as consumers "742-1" and "742-2" with respective L_c's of 8 kW and 85 kW. There are 32 consumers in total. Information about the consumers and their respective power demands are available in the spreadsheet file `Dataset/Ported Load Values.xlsx`.

For simplicity, we assume that a consumer c's appliance active power ratings add up to L_c, that is,

$$L_c = \sum_{p=1}^{P_c} \sum_{i=1}^{M_p} d_{c,i}^{(p)}$$

To assign power ratings for the N_c appliances, we map the interval $[0, L_c]$ unto $[0, 1]$, and sample $N_c - 1$ points within $[0, 1]$ according to the continuous uniform distribution $U\,[0.05, 1)$, thereby dividing $[0, 1]$ (and consequently $[0, L_c]$) into N_c sub-intervals. The length of a sub-interval in $[0, 1]$ is the active power rating of an appliance expressed as a fraction of L_c. In principle, the lengths of the equivalent sub-intervals in $[0, L_c]$ (that is, the appliance active power ratings) add up to L_c. Considering numerical precision, appliance ratings are rounded to 5 decimal places, and the deficit between L_c and the sum of the rounded ratings is added to the rating of a randomly chosen appliance. Each appliance is then assigned a priority level p drawn from a discrete uniform distribution $U_d\,[1, P_c]$.

Our treatment of L_c and how it is divided into N_c appliance ratings is independent of the unit for active power used by 37NTF (or any originating test case). Although this does not reflect the active distribution of appliance ratings in practice, it is sufficient for investigating the concerns raised in Section III-C. We then prepare three datasets (namely, *37NTF-A*, *37NTF-B*, and *37NTF-C*) that differ in terms of N_c and P_c. All procedures are accomplished using NumPy [11] and pandas [12], [13], and are documented in `Dataset/Preparation.ipynb`.

In 37NTF-A, consumers have the same number of appliances and number of priority levels: $N_c = 100$ and $P_c = 5$. Note that this does not mean consumers have identical appliances nor the same number of appliances in a priority level. 37NTF-A can be accessed via the file `Dataset/37NTF-A.xlsx`.

In 37NTF-B, all consumers have the same number of priority levels: $P_c = 5$. However, the consumer with the largest active power demand (say, $\hat{L_c}$) has the most number of appliances, $\hat{N}_c = 100$. For all other consumers, the number of appliances is given by

$$N_c = \left\lfloor \frac{L_c}{\hat{L_c}} \hat{N}_c \right\rfloor + U_d\,[10, 20]$$

The intuition for this scenario is that the consumer with the largest demand can be expected to have the most number of appliances. 37NTF-B is stored as a spreadsheet file `Dataset/37NTF-B.xlsx`.

In 37NTF-C, all consumers have the same number of appliances: $N_c = 100$. They only differ in the number of priority levels; for a consumer c, P_c is drawn from $U_d\,[1, 5]$. This scenario seeks to capture the heterogeneity of consumer preference in terms of appliance prioritization. A consumer with $P_c = 5$ could be someone who prefers some appliances to be picked up first before others. On the other end, $P_c = 1$ means consumer c treats all appliances equally important to be picked up. 37NTF-C is available as a spreadsheet file `Dataset/37NTF-C.xlsx`.

B. Quantitative Summaries

C. Benchmarks

V. Conclusion

Donec molestie, magna ut luctus ultrices, tellus arcu nonummy velit, sit amet pulvinar elit justo et mauris. In pede. Maecenas euismod elit eu erat. Aliquam augue wisi, facilisis congue, suscipit in, adipiscing et, ante. In justo. Cras lobortis neque ac ipsum. Nunc fermentum massa at ante. Donec orci tortor, egestas sit amet, ultrices eget, venenatis eget, mi. Maecenas vehicula leo semper est. Mauris vel metus. Aliquam erat volutpat. In rhoncus sapien ac tellus. Pellentesque ligula.

References

[1] M. E. Jabian, R. Funaki, and J. Murata, "Load Shedding Optimization Considering Consumer Appliance Prioritization Using Genetic Algorithm for Real-time Application," *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 486–491, 2018.

[2] Grid Management Committee, "Philippine Grid Code," Energy Regulatory Commission, Tech. Rep., October 2016.

[3] C. Wester, T. Smith, J. Theron, and D. McGinn, "Developments in Fast Load Shedding," in *2014 67th Annual Conference for Protective Relay Engineers*. IEEE, Mar. 2014.

[2]Data files are obtained from IEEE PES's list of test feeders: https://site.ieee.org/pes-testfeeders/resources/.

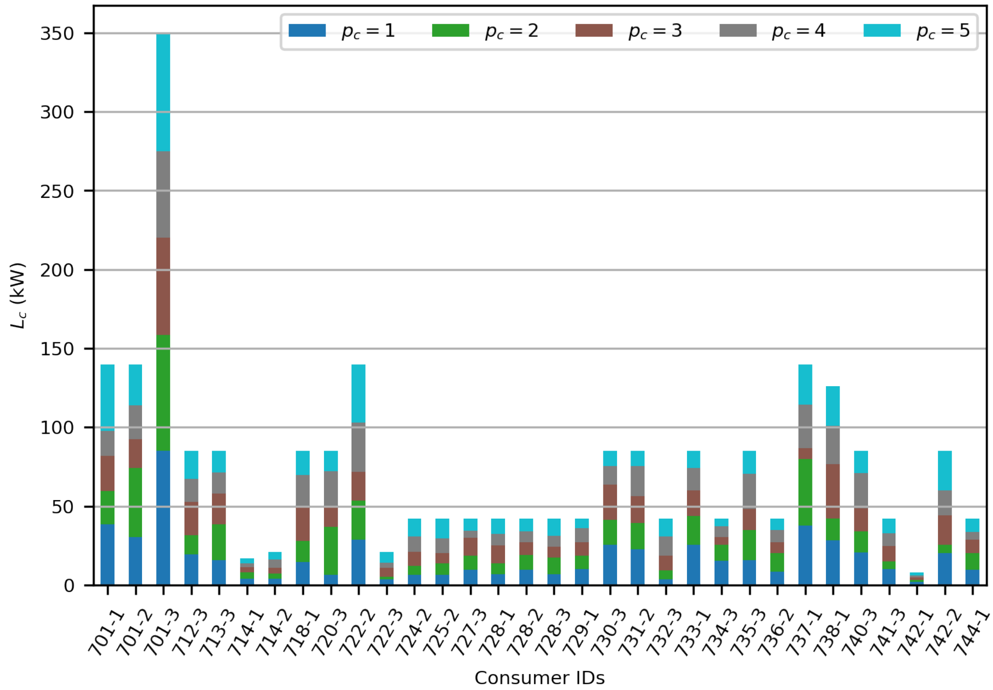

Fig. 5: Utilization of S_{U} among appliances preliminarily nominated by K_c's.

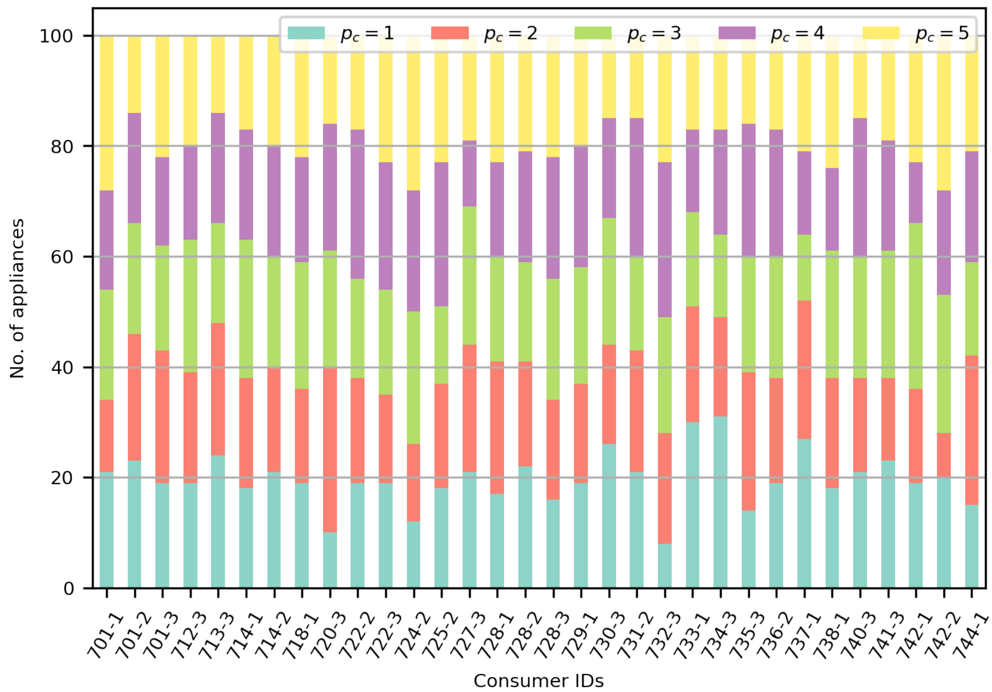

Fig. 6: Utilization of S_{U} among appliances preliminarily nominated by K_c's.

[4] B. Potel, F. Cadoux, L. De Alvaro Garcia, and V. Debusschere, "Clustering-based method for the feeder selection to improve the characteristics of load shedding," *IET Smart Grid*, vol. 2, no. 4, pp. 659–668, Dec. 2019.

[5] S. Babaei, R. Jiang, and C. Zhao, "Distributionally Robust Distribution Network Configuration Under Random Contingency," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 3332–3341, Feb. 2020.

[6] T. Bhattacherjee, A. K. Saha, S. P. Ramalingam, P. K. Shanmugam, and S. Padmanaban, "Server Monitoring and Priority based Automatic Load Shedding Algorithm (SEMPALS)," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*. IEEE, Oct. 2019.

[7] L. Sigrist, "A UFLS Scheme for Small Isolated Power Systems Using Rate-of-Change of Frequency," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 2192–2193, Jul. 2015.

[8] W. Yao, S. You, W. Wang, X. Deng, Y. Li, L. Zhan, and Y. Liu, "A Fast Load Control System Based on Mobile Distribution-Level Phasor Measurement Unit," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 895–904, Jan. 2020.

[9] M. E. Jabian, R. Funaki, and J. Murata, "Consumer Appliance-level Load Shedding Optimization for Real-time Application," 2020.

[10] W. H. Kersting, "Radial Distribution Test Feeders," in *2001 IEEE Power Engineering Society Winter Meeting. Conference Proceedings*. IEEE,

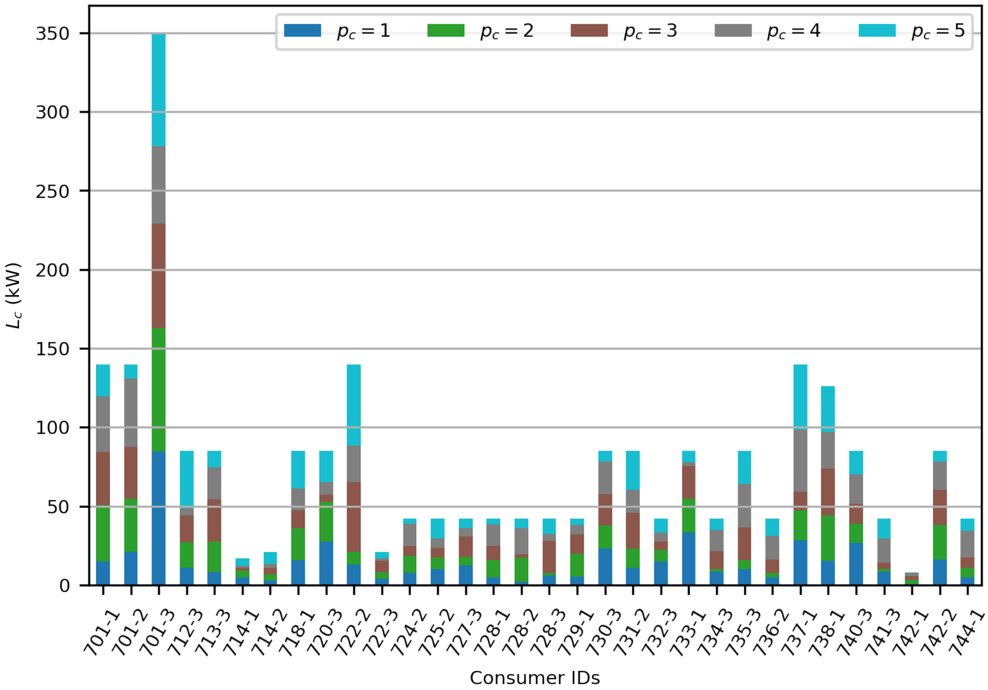

Fig. 7: Utilization of S_{U} among appliances preliminarily nominated by K_c's.

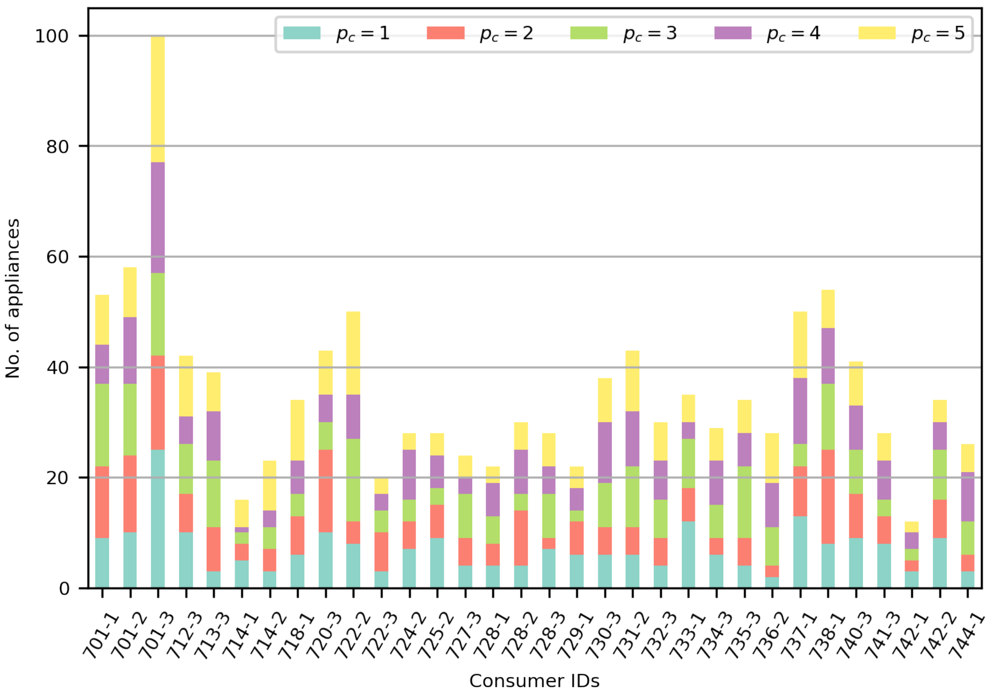

Fig. 8: Utilization of S_{U} among appliances preliminarily nominated by K_c's.

2001.

[11] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[12] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds. SciPy, 2010, pp. 56–61.

[13] J. Reback, W. McKinney, jbrockmendel, J. V. D. Bossche, T. Augspurger, P. Cloud, gfyoung, Sinhrks, S. Hawkins, A. Klein, M. Roeschke, J. Tratner, T. Petersen, C. She, W. Ayd, MomIsBestFriend, M. Garcia, J. Schendel, A. Hayden, D. Saxton, V. Jancauskas, A. McMaster, P. Battiston, Skipper Seabold, chris-b1, h-vetinari, K. Dong, S. Hoyer, W. Overmeire, and M. Winkel, "pandas-dev/pandas: Pandas 1.1.3," Oct. 2020.

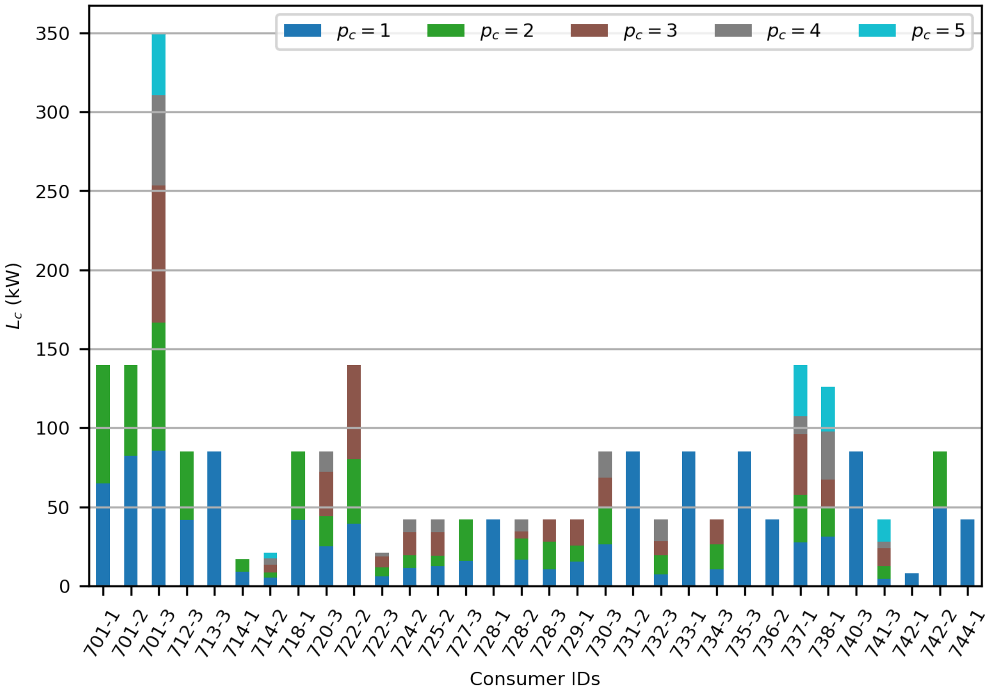

Fig. 9: Utilization of S_{U} among appliances preliminarily nominated by K_c's.

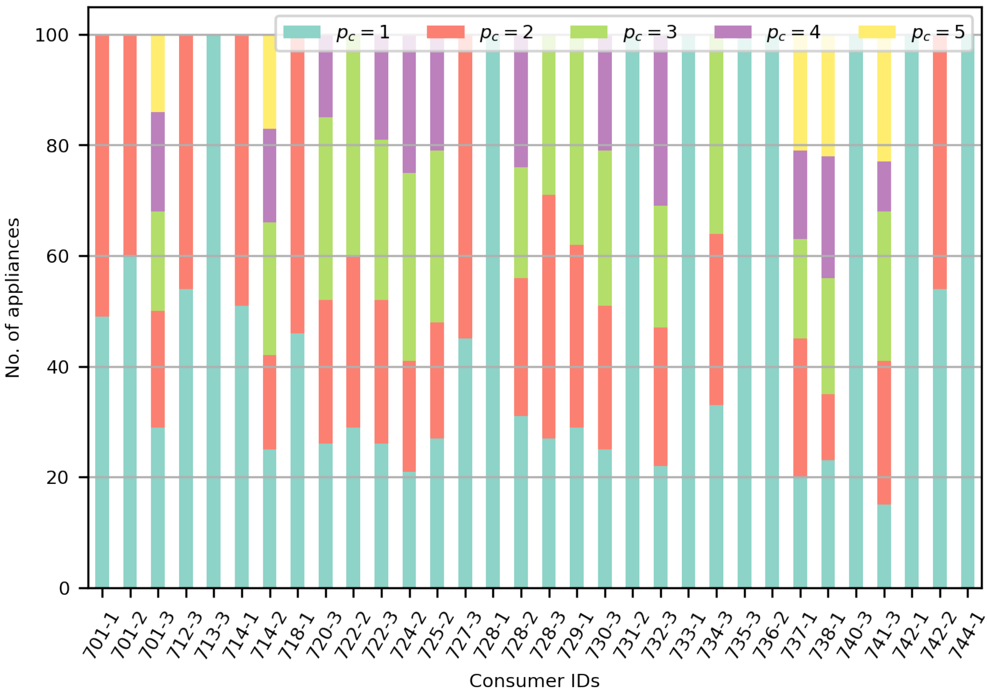

Fig. 10: Utilization of S_{U} among appliances preliminarily nominated by K_c's.