NO-LESS: Near OptimaL CurtailmEnt Strategy Selection for Net Load Balancing in Micro Grids

Sanmukh R. Kuppannagari
Department of Electrical Engineering,
University of Southern California,
Los Angeles, California 90089
Email: kuppanna@usc.edu

Rajgopal Kannan
US Army Research Lab,
12015 Waterfront Drive,
Playa Vista, CA-90094
Email: rajgopal.kannan.civ@mail.mil

Viktor K. Prasanna
Department of Electrical Engineering,
University of Southern California,
Los Angeles, California 90089
Email: prasanna@usc.edu

***Abstract*—The integration of renewable energy sources such as solar PVs into micro grids has increased in recent years. However, the intermittent and unpredictable nature of renewable energy generation makes supply uncertain, in turn requiring continuous net load balancing to ensure grid stability. One technique to achieve net load balancing is curtailment, however, selection of the optimal set of curtailment strategies to achieve a targeted curtailment is NP-hard. State-of-the-art curtailment selection techniques approach this problem by either developing computationally expensive optimal curtailment strategies or reduce accuracy to generate sub-optimal solutions quickly. In this work, we develop NO-LESS: a Near OptimaL CurtailmEnt Strategy Selection algorithm. NO-LESS is a novel Fully Polynomial Time Approximation Scheme (FPTAS) which determines bounded (near optimal) curtailment strategies in a small amount of time while simultaneously satisfying practical constraints on strategy switching overhead and curtailment fairness. We perform both theoretical analysis and practical evaluation to show that NO-LESS is a scalable solution for performing net load balancing through curtailment strategy selection in Micro Grids.**

I. Introduction

Integration of distributed energy resources in micro grids, especially using renewable sources such as solar PVs is opening up new challenges in net load balancing [1]. Renewable energy generation can be intermittent and unpredicatable. The high variability in renewables generation due to weather conditions can lead to frequent load-supply mismatches when either the supply is higher than the load or vice versa.

Traditional net load balancing techniques such as Demand Response are inadequate as they do not address supply surplus. Although techniques such as storage can be used to smooth these imbalances, they are costly and require sophisticated control for providing both upward and downward reserves.

Recent technological advances in solar PV installations such as the development of micro-inverters have opened up new opportunities for net load balancing. Traditionally, the entire solar PV installation was connected to the grid as a monolithic generator. However, with the availability of micro-inverters, each solar panel (called module) in the installation can be independently connected (disconnected) to (from) the grid [2]. By disconnecting a subset of modules, the solar PV installation exhibits discrete solar curtailment configurations.

This work has been funded by the U.S. NSF under grant number ACI 1339756 and the DoE under award number DE-EE0008003.

Therefore, it is now possible to develop a holistic net load balancing framework which performs load curtailment during high demand periods and solar curtailment during periods with surplus supply. Since long-term renewables prediction is more uncertain, the net load balancing framework has to be designed to execute rapidly within short prediction horizons. Significant literature exists for performing fast and optimal curtailment selection when the nodes of the smart grid exhibit continuous curtailment values. However, if the nodes exhibit discrete solar or load curtailment values, as is the case in real world micro grids, then achieving a curtailment target from the nodes can be shown to be NP hard and hence a challenging task. State-of-the-art discrete curtailment selection works have approached this problem by resorting to certain concessions. They either provide computationally expensive optimal solutions or faster heuristics with no optimality bounds.

In this work, we show that this problem can be solved in polynomial time to within an arbitrarily small factor. We develop NO-LESS (Near OptimaL CurtailmEnt Strategy Selection), a fast curtailment algorithm for net load balancing applicable to any micro grid where the generators and consumers exhibit discrete[1] curtailment strategies. We also consider additional practical constraints such as the *cost incurred in switching* and the *inability to arbitrarily switch strategies* in our algorithm. Specifically, we develop an FPTAS that does the following: given an accuracy parameter $\epsilon > 0$, NO-LESS performs strategy selection to achieve a curtailment target in time polynomial in the number of nodes, strategies, and $1/\epsilon$ while ensuring that 1) no node is curtailed more than its maximum allowable value and 2) the total curtailment exceeds the target by no more than a $(1 + \epsilon)$ factor. NO-LESS achieves this while also incorporating some additional practical constraints based on our experience in curtailment in our microgrid, namely strategy switching overhead costs along with limitations on strategy switching in consecutive intervals.

II. Related Works

Supply-demand matching using load curtailment is a well studied field. Optimal polynomial time solutions have been developed for scenarios with continuous load curtailment such

[1]Continuous curtailment values can be discretized to fit into this algorithm.

as in [3] and [4]. However, this assumption is unsuitable for micro grids as the buildings exhibit discrete curtailment values.

In order to tractably solve the problem of achieving load curtailment using discrete curtailment values, stochastic optimization algorithms have been developed in works such as [5] and [6]. However, such works assume the availability of a large number of nodes in the grid. For example, for the technique in [5], the minimum number of customers required to achieve the desired curtailment value with more than 95% probability is quadratic in the curtailment value. Several fast heuristics [7], [8] have also been developed which do not provide any optimality guarantee and can have significantly high errors in worst case. Several other works [9], [10] focus on providing optimality guarantee by developing Integer/Mixed Integer Linear Programming (ILP/MILP) based solutions which are computationally expensive as solving an ILP/MILP takes exponential amount of time.

Solar curtailment has gained attention recently due to the challenge of surplus supply posed by the integration of renewables. Works [11], [12] performing solar curtailment assume the availability of continuous solar curtailment strategies by varying the voltage from the Maximum Power Point (MPP). This requires fine grained control of the solar panels. For some scenarios, fine grained control might not be available due to limitations of inverter technology. Authors in [2] develop a technique which reacts to over-voltages by simply disconnecting individual PV modules from the grid using the micro-inverters thus exhibiting discrete solar curtailment strategies. We leverage this capability to perform proactive solar curtailment with discrete curtailment strategies.

Previously, we developed a 2-factor algorithm for achieving minimum cost net load balancing with fairness [13]. In this work, we develop an (1+ϵ)-factor algorithm which ensures fairness in net load balancing.

III. Curtailment Modeling in Micro Grids

A. Motivation

In our model, a micro grid consists of several nodes. The nodes can be a producer of electricity, for example solar PVs, or a consumer of electricity, for example, buildings in a campus. Each producer node (solar PV) is associated with a discrete set of solar curtailment values achieved by (dis)connecting individual PV modules to the grid using micro-inverter technology [2]. Each consumer node can be switched to one of the several available load curtailment strategies [14]. A control center can remotely switch each node into one of several available strategies in each time interval of the curtailment horizon. Each such node-strategy pair exhibits a fixed amount of curtailment. Renewable generation and consumer load predictions are performed to predict the expected supply-load mismatch $\mathcal{C}$ during a given time horizon consisting of T time intervals. The control center then determines the node-strategy pairs across the intervals of T to achieve the desired curtailment target $\mathcal{C}$.

This setting has several constraints which make this problem hard. Firstly, each node can follow one of several available strategies, each associated with a fixed curtailment value. Hence, the curtailment values exhibited by a single node form a discrete set. However, such node strategy pairs are not large enough for the applicability of the law of large numbers which is critical for the success of stochastic optimization based techniques. A simple version of this problem, where a targeted curtailment value needs to be achieved using a set of discrete curtailment values with no additional constraints, can be shown to be NP hard by reducing the well known subset sum problem to it. Secondly, each strategy switch by a node requires some ramp time to come into effect. For certain pairs of strategies, this ramp time is prohibitively large. Hence, a node following some strategy in an interval cannot switch to any arbitrary strategy in the subsequent interval.

In our previous work [15], we developed an FPTAS for this problem to address only the first constraint. The algorithm did not take into account the strategy switching overheads. In this work, we develop a more sophisticated algorithm to address both the constraints.

B. Bounding Curtailment Unfairness

In order to ensure that certain nodes of the grid are not unfairly penalized by incurring disproportionately large curtailment, we assign a budget value C_{bmax} with each node b. NO-LESS algorithm, by ensuring that each node curtails within its targeted budget, avoids unfair penalization of each node of the micro grid.

C. Incorporating Strategy Switching Overheads

In order to incorporate strategy switching overheads, for a node b, we define a function Γ_b to model the cost of switching between allowable strategies. The purpose of the cost is to disallow frequent strategy switching when ramp time is high. If the node is allowed to switch from strategy j to k, then $\Gamma_b(j,k) < \infty$ denotes the cost of switching, else $\Gamma_b(j,k) = \infty$. Note that Γ_b can be represented using a 2D square matrix whose i,j entry represents the cost of switching from strategy i to j. This ensures that a single call of Γ_b requires $O(1)$ amount of time. For a node b, we limit the cost incurred in switching strategies by τ_b.

IV. NO-LESS

A. Problem Formulation

We are given a set of M nodes and N strategies. The entire curtailment horizon is divided into discrete time intervals. We are given a time varying curtailment matrix $\mathbf{C(t)} \in \mathbf{R}^{M \times N}$ with element $c_{bj}(t)$ denoting the discrete curtailment value of node b adopting strategy j at time interval t where $t \in \{1, \ldots, T\}$. Let C_{bmax} be the maximum curtailment value for node b. Let $\mathbf{X(t)}$ be the decision matrix with element $x_{bj}(t)$ denoting the corresponding decision variable at time t. The achievable curtailment value across the entire curtailment horizon is given by $\mathcal{C}$. τ_b denotes the limit on the total cost of strategy switches for a node b. We assume that strategy 1 is the default strategy with a curtailment value of 0 i.e. if a

node is not included in an interval of the curtailment horizon, it follows strategy 1.

Given the model above, the objective is to determine node-strategy pairs for each interval such that: (1) The curtailment target is achieved with minimum curtailment error: difference between the achieved and the targeted curtailment, (2) No node is curtailed more than its maximum allowable value, (3) only allowable strategy switches are performed by each node across each consecutive intervals, and (4) for each node, the cost incurred in switching strategies is within its allowable limit.

B. FPTAS for NO-LESS

We first develop a dynamic programming based FPTAS to determine the set of curtailment strategies followed by a single node b during the curtailment horizon to achieve a curtailment value of at most $(1+\epsilon)\mathcal{C}$, where ϵ is a user determined accuracy parameter. The cost of strategy switches will be bounded by τ_b (Section IV-B1). We then combine the results of all the nodes to achieve the targeted curtailment value from all the nodes (Section IV-B2).

1) Achieving Curtailment Target for a Single Node: Let $\mu = \frac{\epsilon\mathcal{C}}{T}$. For each $c_{bj}(t)$, we define $\widehat{c}_{bj}(t) = \lfloor \frac{c_{bj}(t)}{\mu} \rfloor$. We also define $\widehat{\mathcal{C}} = \lfloor \frac{\mathcal{C}}{\mu} \rfloor$. We define a boolean function Θ which returns true if a curtailment value $\widehat{\mathcal{C}_t}$ can be achieved using strategy S_k at time t incurring a cost $\leq q_t$ strategy switches and False otherwise. Θ can be defined using the following dynamic programming formulation:

$$\begin{aligned}\Theta(\widehat{\mathcal{C}_t}, t, S_k, q_t) &= \text{FALSE if } \widehat{\mathcal{C}_t} - \widehat{c}_{bk}(t) < 0 || q_t < 0 \\ &= ||_j \Theta(\widehat{\mathcal{C}_t} - \widehat{c}_{bk}(t), t-1, S_j, q_t - \Gamma_b(j,k)) \\ &\quad (\forall j \in \{1,\ldots,N\}) \ \ otherwise \end{aligned} \tag{1}$$

If any of $\Theta(\widehat{\mathcal{C}}, T, S_k, \tau_b)\ \forall k \in \{1,\ldots,N\}$ is True, we can determine the required strategies by traversing the recursive formulation and determining the strategies at each time t as described in Algorithm 1. The dynamic program can be solved by creating a table of size $\widehat{\mathcal{C}} \times T \times N \times \tau_b$. We will refer to the entire table as Θ to simplify the notations. We can initialize the table using the following equations:

$$\begin{aligned}\Theta(\mathcal{C}_l, 1, S_k, q) =& \text{TRUE if } \widehat{c}_{bk}(1) == \mathcal{C}_l \forall k \in \{1,\ldots,N\} \\ &\&\& \ q >= \Gamma_b(1k) \ \ \forall l \\ &\text{FALSE otherwise}\end{aligned} \tag{2}$$

Lemma 1. *Algorithm 1 finds the strategies to be followed in each interval by node b to achieve the curtailment value $\mathcal{C}$ within a cost of τ_b for strategy switches with a time complexity which is polynomial in N, T and $\frac{1}{\epsilon}$.*

Proof. We omit the proof of correctness in the interest of space as it can easily by argued using the arguments used for dynamic programming algorithms. Now, filling a single entry of the table requires $O(N)$ time. Hence, the algorithm requires $O(\frac{T^2N^2\tau_b}{\epsilon})$ which is the dominating term. We assume $\tau_b = O(T)$ i.e. the ratio of the largest to smallest cost is bounded and the number of switches is $\leq T$, the algorithm is polynomial in N, T and $\frac{1}{\epsilon}$. □

Algorithm 1: A $(1+\epsilon)$ polynomial time approximation algorithm to achieve the curtailment value $\mathcal{C}$ by a single node b

Fill entries $\Theta(\widehat{\mathcal{C}_l}, t, S_k, q_t) \forall \widehat{\mathcal{C}_l} \in \{0,\ldots,\widehat{\mathcal{C}}\}, \forall t \in \{1,\ldots,T\}, \forall S_k, k \in \{1,\ldots,N\}, \forall q_t \in \{1,\ldots,\tau_b\}$ using equations 1 and 2
$X(t) \leftarrow \phi \ \ \forall t \in \{1,\ldots,T\}$
$\mathcal{C}_{cur} \leftarrow \widehat{\mathcal{C}}$
$q_{cur} \leftarrow \tau_b$
for *Time* $t = T$ *to* 1 **do**
  if $\exists k \in 1,\ldots,N$ *s.t* $\Theta(\mathcal{C}_{cur}, t, S_k, q_{cur})$ **then**
    if $!\Theta(\mathcal{C}_{cur} - c_{bk}(t), t-1, S_k, q_{cur})$ **then**
      $q_{cur} \leftarrow q_{cur} - 1$
    $\mathcal{C}_{cur} \leftarrow \mathcal{C}_{cur} - c_{bk}(t)$
    $X(t) \leftarrow S_k$

Output: X, where $X(t)$ denotes the strategy to be followed at time t

Lemma 2. *The total curtailment value obtained by following the strategies output by Algorithm 1 is $\leq (1+\epsilon)\mathcal{C}^*$ where $\mathcal{C}^*$ is the total curtailment value obtained by following the strategies output by an optimal algorithm.*

Proof. Let $\mathcal{C}_x$ be the curtailment value obtained by following the strategies output by Algorithm 1. Let $\mathcal{C}^*$ be the optimal curtailment value. Clearly, $\mathcal{C}_x \leq \mu \sum_{t=1}^{T} c_{bX(t)}(t)$. Also, $\mathcal{C}^* \geq \mu \sum_{t=1}^{T} (c_{bX(t)}(t) - 1)$. This implies $\mathcal{C}_x \leq \mathcal{C}^* + \mu T \leq \mathcal{C}^* + \epsilon\mathcal{C}$. Now, since $\mathcal{C} \leq \mathcal{C}^*$, $\mathcal{C}_x \geq \mathcal{C}^*(1+\epsilon)$. □

Using Lemmas 1 and 2, we get the following theorem.

Theorem 1. *Algorithm 1 is a FPTAS to determine the strategy to be followed in each interval $1,\ldots,T$ to achieve a curtailment value $\mathcal{C}$ by a single node b with the cost of strategy switches is bounded by τ_b.*

2) Achieving Total Curtailment by all the Nodes: We will run step 1 of Algorithm 1 with curtailment value $\widehat{\mathcal{C}}_{bmax}$, the node-specific maximum curtailment value, $\epsilon' = \frac{\epsilon}{M}$, and limit on the cost on strategy switching τ_b for each node $b \in \{1,\ldots,M\}$ to create Θ_b. For a node b, define $\Phi_b = \cup\ \widehat{C}_l \mid \exists\{S_k, q\}$ with $\Theta_b(\widehat{C}_l, T, S_k, q) =$ TRUE. Assume Φ_b is in sorted order. Let $\mathcal{L}(\Phi_b)$ denote the number of entries in Φ_b

We will again use a dynamic programming based algorithm for this problem. We define a boolean function Ξ which returns True if a curtailment value $\widehat{\mathcal{C}_b}$ can be obtained by using nodes $\{1,\ldots,b\}$ and False otherwise. Ξ can be defined using the following formulation:

$$\begin{aligned}\Xi(\widehat{\mathcal{C}_b}, b) &= \text{TRUE if } \exists k \in \{1,\ldots,\mathcal{L}(\Phi_b)\} \\ &\qquad s.t. \Xi(\widehat{\mathcal{C}_b} - \Phi_b(j), b-1) \text{ is TRUE} \\ &= \text{FALSE otherwise}\end{aligned} \tag{3}$$

Ξ can be initialized for all $\widehat{\mathcal{C}_b} \in \{0, \ldots, \widehat{\mathcal{C}}\}$, where $\widehat{\mathcal{C}} = \lfloor \frac{\mathcal{C}}{\mu} \rfloor, \mu = \frac{\epsilon \mathcal{C}}{MT}$ by the following equations:

$$\Xi(\widehat{\mathcal{C}_b}, 1) = \text{TRUE if } \exists k \in \{1, \ldots, \mathcal{L}(\Phi_b)\} \;\; s.t. \widehat{\mathcal{C}_b} = \Phi_b(k)$$
$$= \text{FALSE otherwise} \quad (4)$$

Given a total curtailment value $\mathcal{C}$, Algorithm 2 can be used to combine the curtailment values that can be achieved from individual nodes to attain the total curtailment value.

Algorithm 2: A $(1+\epsilon)$ polynomial time approximation algorithm to achieve the curtailment value $\mathcal{C}$

1 Run Step 1 of Algorithm 1 for each node b with $\widehat{\mathcal{C}_{bmax}} = \lfloor \frac{\mathcal{C}_{bmax}}{\mu} \rfloor$, where $\mu = \frac{\epsilon \mathcal{C}}{MT}$, $\epsilon' = \frac{\epsilon}{M}$, and $\tau = \tau_b$ and produce Φ_b
2 Fill entries $\Xi(\widehat{\mathcal{C}_b}, b) \forall \widehat{\mathcal{C}_b} \in \{0, \ldots, \widehat{\mathcal{C}}\}, \forall b \in \{1, \ldots, M\}$ using equations 3 and 4
3 $X_b(t) \leftarrow \phi \; \forall b \in \{1, \ldots, M\}, \forall t \in \{1, \ldots, T\}$
4 $\mathcal{C}_{cur} \leftarrow \widehat{\mathcal{C}}$
5 **for** $b = M$ *to* 2 **do**
6 **if** $\exists k \in \{1, \ldots, \mathcal{L}(\Phi_b)\}$ *s.t.* $\Xi(\mathcal{C}_{cur} - \Phi_b(k), b-1) == 1$ **then**
7 Call Algorithm 1 with inputs $\Phi_b(k), T, \tau_b$ to produce output X
8 $X_b \leftarrow X$
9 $\mathcal{C}_{cur} \leftarrow \mathcal{C}_{cur} - \Phi_b(k)$

Output: X, where $X_b(t)$ denotes the strategy to be followed by node b at time t

Theorem 2. *Algorithm 2 is a FPTAS for NO-LESS*

Proof. The correctness can be argued similar to that of Algorithm 1. We skip the argument in the interest of space.

Let $\mathcal{C}_x$ be the curtailment value obtained by following the strategies output by Algorithm 2. Let $\mathcal{C}^*$ be the optimal curtailment value. Clearly, $\mathcal{C}_x \leq \mu \sum_{b=1}^{M} \sum_{t=1}^{T} c_{bX(t)}(t)$. Also, $\mathcal{C}^* \geq \mu \sum_{b=1}^{M} \sum_{t=1}^{T} (c_{bX(t)}(t) - 1)$. This implies $\mathcal{C}_x \leq \mathcal{C}^* + \mu MT \leq \mathcal{C}^* + \epsilon \mathcal{C}$. Now, since $\mathcal{C} \leq \mathcal{C}^*$, $\mathcal{C}_x \geq \mathcal{C}^*(1+\epsilon)$.

Step 1 of Algorithm 2 requires $O(\frac{MT^3N^2}{\epsilon})$ time. Step 2 requires $O(\widehat{\mathcal{C}})$ time to fill each of the $\widehat{\mathcal{C}} \times M$ entries hence, $O(\frac{M^3T^2}{\epsilon^2})$ time. These being the dominating terms, the total complexity of Algorithm 2 is $O(\frac{M^3T^2}{\epsilon^2} + \frac{MT^3N^2}{\epsilon})$ which is polynomial in M, N, T and $\frac{1}{\epsilon}$. □

V. Evaluation

We evaluate the algorithm developed in this work using the curtailment dataset obtained by the DR programs conducted in the University of Southern California (USC). The USC micro grid consists of more than 150 buildings equipped with smart meters to capture consumption data in 15-minute intervals. Each DR lasted for 4 hours during which each building switched to one out of 6 available strategies [14]. The curtailment data was obtained by using the algorithm developed in [16]. For solar curtailment, we use simulated data generated by using solar irradiance data for Los Angeles County [17] and varying the solar PV are from $10m^2$ to $20m^2$ and the yield from 5% to 15%. 6 different curtailment strategies: 0, $0.125 * O, 0.25 * O, 0.5 * O, 0.75 * O, O$, with O being the maximum output were generated. The cost of switching between strategies was fixed as 1. And the cost of strategy switching was limited to 10. The allowable strategy switches were $0 \leftrightarrow 1 \leftrightarrow 5$, $0 \leftrightarrow 2 \leftrightarrow 4 \leftrightarrow 5$ and $0 \leftrightarrow 3 \leftrightarrow 5$, where 0 is the default strategy with a curtailment value of 0. We implemented the NO-LESS in Java. The experiments were performed on Dell optiplex with 4-cores and 4 GB RAM.

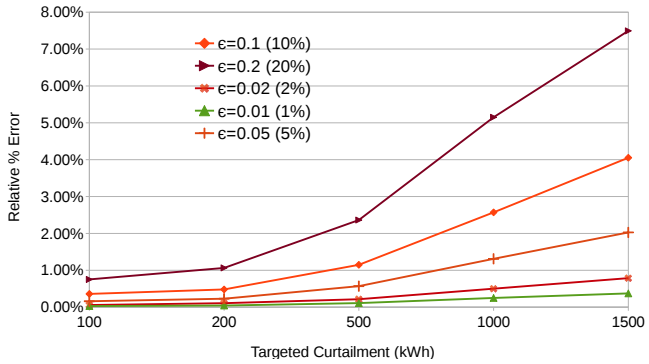

Fig. 1. Near Optimality of NO-LESS

A. Near Optimality

We implemented an Integer Linear Program (ILP) for the problem objective to obtain optimal solutions. We varied the targeted curtailment values from 100 kWh to 2000 kWh. The number of nodes was fixed at 20. Figure 1 shows the relative percentage error incurred by NO-LESS algorithm compared against the optimal solution. Relative percentage error is defined as: $\frac{(P-O)*100}{O}$, where P is the solution obtained by NO-LESS and O is the optimal solution obtained by the ILP. As shown in the figure, the relative percentage errors incurred by NO-LESS are much less than the worst case guarantee as determined by ϵ.

Notice that the relative percentage error increases as the curtailment target increases. This is because our algorithm is essentially a packing problem. For lower curtailment values, the algorithm has enough available node-strategy pairs to choose from to pack. However, as the curtailment value increases, the availability of such pairs becomes sparse. While the optimal algorithm tries all possible combination for optimizing the packing, our algorithm terminates as soon as it finds one which is within the worst case guarantee ϵ.

B. Scalability

NO-LESS algorithm has two critical parameters which affect scalability: the number of nodes M and the accuracy parameter ϵ. Note that our algorithm is independent of the targeted curtailment value.

Figure 2 shows the effect on the execution time of the number of nodes for various values of ϵ. For each ϵ, the execution time increases polynomially with respect to the

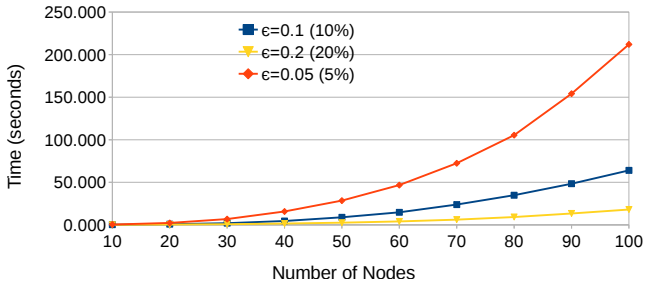

Fig. 2. Execution Time vs Number of nodes for different values of epsilon

number of nodes. Even for an accuracy parameter of 5%, the runtime is less than 4 minutes.

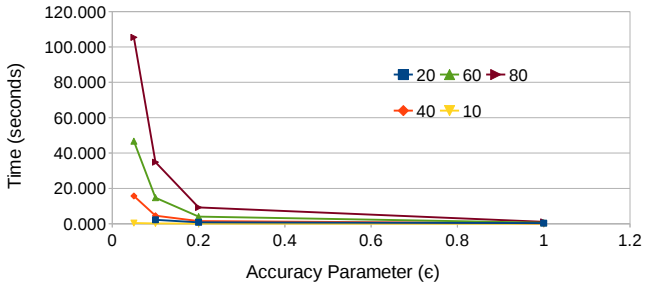

Fig. 3. Execution Time vs epsilon for different number of nodes

Figure 3 shows the trade-off between the accuracy parameter ϵ and the execution time in seconds for various number of nodes. The execution time increases quadratically with respect to $\frac{1}{\epsilon}$. For very small values of ϵ such as 0.01 (1%), this leads to very large runtimes such as 600 seconds for 50 nodes. However, our algorithm fills the dynamic programming table in such a way that all entries in a row can be independently populated by looking at the entries of previous columns. This provides us with an opportunity to easily parallelize the algorithm to achieve dramatic improvement in the runtime. We do not perform such analysis in this work.

C. Comparison with State-of-the-art Techniques

Table I compares the theoretical and experimental errors and the computation time of our algorithm for various values of ϵ with state-of-the-art techniques. Our algorithm, with a small increase in runtime is able to provide solutions with extremely low errors as shown in the table.

TABLE I
ERROR AND SCALABILITY (20 NODES) COMPARISON

Technique	Theoretical Error	Experimental Error	Time
Change Making [8]	None	2-10%	≈1 second
LP based algorithm [4]	None	20-95%	≈1 second
NO-LESS ($\epsilon = 0.05$)	5%	<1.5%	2.2 seconds
NO-LESS ($\epsilon = 0.01$)	1%	<0.25%	33 seconds

VI. CONCLUSION

Integration of renewables provides an economic way to supply for the loads of the consumers in a micro grids. However, this comes with a caveat as the problem of discrete curtailment strategy selection is NP-hard. In this work, using theoretical guarantees as well as practical evaluation, we prove that near optimal results can be obtained in polynomial time for this problem. We also provide the users with a parameter to trade-off accuracy versus computation time.

In our future works, we will incorporate the network model of the grid. We will also incorporate interval wise net load balancing. The simplicity of our problem formulation makes it easier to incorporate any such additional modifications or constraints with minimal effort.

REFERENCES

[1] P. Basak, S. Chowdhury, S. H. nee Dey, and S. Chowdhury, "A literature review on integration of distributed energy resources in the perspective of control, protection and stability of microgrid," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 8, pp. 5545–5556, 2012.

[2] O. Gagrica, P. H. Nguyen, W. L. Kling, and T. Uhl, "Microinverter curtailment strategy for increasing photovoltaic penetration in low-voltage networks," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 2, pp. 369–379, 2015.

[3] J.-Y. Kim, J.-H. Lee, and H.-M. Kim, "Optimal operation of green campus based on demand response," *International Journal of Control and Automation*, vol. 7, no. 12, pp. 415–424, 2014.

[4] N. Gatsis and G. B. Giannakis, "Cooperative multi-residence demand response scheduling," in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*. IEEE, 2011, pp. 1–6.

[5] J. Kwac and R. Rajagopal, "Demand response targeting using big data analytics," in *Big Data, 2013 IEEE International Conference on*. IEEE, 2013, pp. 683–690.

[6] Z. Chen, L. Wu, and Y. Fu, "Real-time price-based demand response management for residential appliances via stochastic optimization and robust optimization," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1822–1831, 2012.

[7] T. Logenthiran, D. Srinivasan, and T. Z. Shun, "Demand side management in smart grid using heuristic optimization," *Smart Grid, IEEE Transactions on*, vol. 3, no. 3, pp. 1244–1252, 2012.

[8] V. Zois, M. Frincu, C. Chelmis, M. R. Saeed, and V. Prasanna, "Efficient customer selection for sustainable demand response in smart grids," in *IGCC*. IEEE, 2014, pp. 1–6.

[9] Z. Luo, R. Kumar, J. Sottile, and J. C. Yingling, "An milp formulation for load-side demand control," *Electric machines and power systems*, vol. 26, no. 9, pp. 935–949, 1998.

[10] A. Barbato, C. Bolchini, M. Delfanti, A. Geronazzo, G. Accetta, A. Dede, G. Massa, and M. Trioni, "An energy management framework for optimal demand response in a smart campus," *ICGREEN*, p. 11, 2015.

[11] S. Lee, S. Iyengar, D. Irwin, and P. Shenoy, "Distributed rate control for smart solar arrays," in *Proceedings of the Eighth International Conference on Future Energy Systems*. ACM, 2017, pp. 34–44.

[12] A. Singh, S. Lee, D. Irwin, and P. Shenoy, "Sunshade: enabling software-defined solar-powered systems," in *Proceedings of the 8th International Conference on Cyber-Physical Systems*. ACM, 2017, pp. 61–70.

[13] S. R. Kuppannagari, R. Kannan, and V. K. Prasanna, "Optimal net-load balancing in smart grids with high pv penetration," *arXiv preprint arXiv:1709.00644*, 2017.

[14] S. R. Kuppannagari, R. Kannan, C. Chelmis, and V. K. Prasanna, "Implementation of learning-based dynamic demand response on a campus micro-grid," in *The 25th International Joint Conference on Artificial Intelligence*, 2016.

[15] S. R. Kuppannagari, R. Kannan, C. Chelmis, A. S. Tehrani, and V. K. Prasanna, "Optimal customer targeting for sustainable demand response in smart grids," *Procedia Computer Science*, vol. 80, pp. 324–334, 2016.

[16] C. Chelmis, S. Aman, M. R. Saeed, M. Frincu, and V. K. Prasanna, "Estimating reduced consumption for dynamic demand response," in *Proceedings of the Twenty-Ninght AAAI Conference on Artificial Intelligence. AAAI Press*, 2015.

[17] (2010) National solar radiation data base. [Online]. Available: http://rredc.nrel.gov/solar/old_data/nsrdb/1991-2010/hourly/list_by_state.html