Linearizing Bilinear Products of Shadow Prices and Dispatch Variables in Bilevel Problems for Optimal Power System Planning and Operations

Nicholas D. Laws, *Student Member, IEEE,* and Grani A. Hanasusanto

***Abstract*—This work presents a method for linearizing bilinear terms in the upper level of bilevel optimization problems when the bilinear terms are products of the primal and dual variables of the lower level. Bilinear terms of this form often appear in energy market optimization models where the dual variable represents the market price of energy and the primal variable represents a generator dispatch decision. Prior works have linearized such bilinear terms for specific problems. This work is the first to demonstrate how to linearize these terms in the most general case and the conditions required to perform the linearization for bilevel problems with integer or continuous variable in the upper level. The method is provided in an open source Julia module that allows researchers to write their bilevel programs in an intuitive fashion.**

***Index Terms*—Duality, Optimization methods, Power system economics, Power system planning.**

I. Introduction and Background

Since the restructuring of electricity markets began in the early 1980's [1] and the introduction of locational marginal pricing into large scale power markets in the 1990's researchers have been investigating electricity market design optimization problems. From a market participant point-of-view one of the most critical terms in a problem is the price signal (typically in \$/MWh) from the market operator multiplied by the energy delivered (MWh) by the participant, which together represent the participant's income. When both the price signal and energy delivered are decision variables in a mathematical program then the problem becomes bilinear.

In many electricity markets the price signal to market participants (or generators) is determined as the marginal price of the load balance constraint at any given network node at any given time step. The objective of the market model is to minimize the total cost of energy, or as it is commonly known: maximizing the social welfare. The constraints of the model typically represent an approximation to the power flow equations and take participant cost functions as input.

Equilibrium models allow modeling both the electricity market and participant behavior by including the power flow constraints and multiple, competing objective functions. Bilevel or Stackelberg Game formulations are common in electricity market models that include participant objectives, which are typically to maximize profits.

N. D. Laws is with the National Renewable Energy Laboratory, Golden, CO, USA and the Walker Department of Mechanical Engineering, The University of Texas at Austin, TX, USA (e-mail: nick.laws@nrel.gov).

G. A. Hanasusanto is with the Walker Department of Mechanical Engineering, The University of Texas at Austin, TX, USA (e-mail: grani.hanasusanto@utexas.edu).

Manuscript received August 24, 2021.

Ruiz *et al*. 2009 [2] is the earliest known example to demonstrate that bilinear terms for market price and participant dispatch can be linearized. Their model places the market participant in the upper level, which chooses its offer curve for energy generation, while the lower level models the electricity market given the other participants' offer curves. Fernandez-Blanco *et al*. 2016 [3] is the first to find a linearization for the same bilinear terms (products of lower level primal and dual variables) in the upper level of a bilevel program for revenue adequacy constraints. More recently, Naebi *et al*. 2020 [4] forms a bilevel problem to optimize the bidding strategy of a microgrid owner in a day ahead market. The upper level minimizes operating costs from the microgrid owner's perspective with the product of its exported power and the dual variable of the lower level, linear power flow load balance in its objective. (In other words, the microgrid operator knows its impact on the market price). The lower level minimizes the system operator's cost, including the payment to the microgrid owner, subject to linear power flow constraints. Xu *et al*. 2020 [5] proposes a bilevel model in which the upper level represents a coalition of PV system owners that can sell excess power to the grid or to other consumers. The upper level objective contains a bilinear product of the price to charge consumers and the level of excess PV production to be sold. The lower level objective is the sum of the PV owners' cost functions, which contain the benefit of selling excess PV and the cost of consuming grid power. The bilinear product of price and dispatch variables is linearized by setting the lower level primal objective equal to the dual objective. Additional problem specific examples of the linearization technique can be found in [6] and [7].

Each of the aforementioned examples presents problem-specific examples of how the bilinear products of shadow prices and dual variables can be linearized in bilevel problems using Strong Duality Theorem [8]. This paper presents a general algorithm for linearizing the bilinear terms of interest and determines the exact conditions under which the bilinear terms can be linearized in general bilevel problems. The algorithm is implemented in an open source Julia module for mathematical programming that allows researchers to write their bilevel problems in an intuitive fashion ([9], which extends [10]). After showing the linearization algorithm we present some simple and complex use-case examples to demonstrate the value of the linearization method for power system planning research questions. In the complex use-case, with a power flow model, we show that the linearization method makes otherwise intractable problems solvable in a matter of minutes. Using the open source module other researchers can take advantage of the linearization method for any bilevel problem with bilinear products of shadow prices and dispatch variables of interest.

II. Linearization Method with Integer Upper Level Variables

TABLE I: Sets, indices, parameters, and decision variables.

Decision Variables	
$\boldsymbol{x} \in \mathcal{R}^M$	upper level, primal decision variables
$\boldsymbol{y} \in \mathcal{R}^N$	lower level, primal decision variables
$\boldsymbol{\lambda} \in \mathcal{R}^J$	lower level, dual variables for equality constraints
$\overline{\boldsymbol{\mu}} \in \mathcal{R}^N_+$	lower level, non-negative, dual variables for upper bounds
$\underline{\boldsymbol{\mu}} \in \mathcal{R}^N_+$	lower level, non-negative, dual variables for lower bounds
Parameters	
$\boldsymbol{c} \in \mathcal{R}^N$	lower level cost coefficients for lower level decisions $\boldsymbol{y}$
$\boldsymbol{U} \in \mathcal{R}^{J\times M}$	lower level equality constraint coefficients for upper level decisions x
$\boldsymbol{V} \in \mathcal{R}^{J\times N}$	lower level equality constraint coefficients for lower level decisions $\boldsymbol{y}$
$\boldsymbol{w} \in \mathcal{R}^J$	lower level equality constraints right-hand-side
$\overline{\boldsymbol{y}} \in \mathcal{R}^N$	upper bounds for lower level, primal decision variables
$\underline{\boldsymbol{y}} \in \mathcal{R}^N$	lower bounds for lower level, primal decision variables
$\boldsymbol{A} \in \mathcal{R}^{J\times N}$	upper level coefficients for bilinear terms of lower level primal and dual variables
$\boldsymbol{B} \in \mathcal{R}^{M\times N}$	lower level coefficients for bilinear terms of lower level primal and upper level primal variables
Sets and Indices	
$\mathcal{A}$	$\{(j,n) \in \mathcal{J}\times\mathcal{N} : A_{jn} \neq 0\}$
$\mathcal{A}_\mathcal{J}$	$\{j \in \mathcal{J} : \exists n \in \mathcal{N}$ such that $A_{jn} \neq 0\}$
$\mathcal{A}_\mathcal{N}$	$\{n \in \mathcal{N} : \exists j \in \mathcal{J}$ such that $A_{jn} \neq 0\}$
$\mathcal{J}$	$1, 2, \dots, J$, $\lvert\mathcal{J}\rvert$ = number of lower level equality constraints
$\mathcal{J}_j \subseteq \mathcal{J}$	indices of lower level equality constraints connected to constraint j via non-zero values of $\boldsymbol{V}$, i.e. the constraints that share variables with constraint j and the constraints that share variables with those constraints (and so on recursively as described in Algorithm 1).
$\mathcal{J}_\cup$	$\bigcup_{j\in\mathcal{A}_\mathcal{J}} \mathcal{J}_j$
$\mathcal{M}$	$1, 2, \dots, M$, $\lvert\mathcal{M}\rvert$ = number of upper level variables
$\mathcal{N}$	$1, 2, \dots, N$, $\lvert\mathcal{N}\rvert$ = number of lower level variables
$\mathcal{N}_n \subseteq \mathcal{N}$	indices of lower level variables connected to variable y_n via non-zero values of $\boldsymbol{V}$
$\mathcal{N}_\cup$	$\bigcup_{n\in\mathcal{A}_\mathcal{N}} \mathcal{N}_n$
$\mathcal{AB}_\mathcal{N}$	$\{n \in \mathcal{A}_\mathcal{N} : \exists m \in \mathcal{M}$ such that $B_{mn} \neq 0\}$
$\mathcal{AB}$	$\{(j,n) \in \mathcal{A} : \exists m \in \mathcal{M}$ such that $B_{mn} \neq 0\}$
$\emptyset$	The empty set
$\mathcal{Z}$	The set of integers

We begin by assuming that the upper level variables $\boldsymbol{x}$ are integer such that any product of integer $\boldsymbol{x}$ and the continuous variables $\boldsymbol{y}$ or $\boldsymbol{\lambda}$ can be made linear using binary expansion [11][1]. The bilevel problem with bilinear terms in the upper level objective and a linear lower level is

$$\min_{\boldsymbol{x}\in\mathcal{Z}^M, \boldsymbol{y}\in\mathcal{R}^N} \; f(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{\lambda}^\mathsf{T}\boldsymbol{A}\boldsymbol{y} \tag{1a}$$

$$\text{s.t. } g(\boldsymbol{x},\boldsymbol{y}) \le 0 \tag{1b}$$

$$\boldsymbol{y} \in \arg\min_{\boldsymbol{y}\in\mathcal{R}^N} \; \boldsymbol{c}^\mathsf{T}\boldsymbol{y} + \boldsymbol{x}^\mathsf{T}\boldsymbol{B}\boldsymbol{y} \tag{1c}$$

$$\text{s.t. } \quad \underline{\boldsymbol{y}} \le \boldsymbol{y} \le \overline{\boldsymbol{y}} \quad (\underline{\boldsymbol{\mu}}, \overline{\boldsymbol{\mu}}) \tag{1d}$$

$$\boldsymbol{U}\boldsymbol{x} + \boldsymbol{V}\boldsymbol{y} = \boldsymbol{w} \quad (\boldsymbol{\lambda}). \tag{1e}$$

Table I summarizes the terms in Equation 1. Note that the method is also valid for bilinear terms of $\boldsymbol{\lambda}$ and $\boldsymbol{y}$ in the upper level constraints, but they are not shown for clarity.

The linearization algorithm is applicable when the upper level and/or the lower level problems are non-linear in constraints or objectives. However, the lower level constraints that include the lower level variables from the upper level bilinear terms must be linear to get an exact linearization of the upper level bilinear terms. Futhermore, we assume that the lower level problem is linear in its decision variables (given the upper level decisions) so that we can replace the lower level with its Karush Kuhn Tucker conditions to show single level problem equivalents to the non-linear bilevel problems of interest.

To linearize any $\lambda_j y_n$ term one must combine the lower level primal and dual constraints. The dual formulation of the lower level problem is shown below for reference.

$$\max_{\overline{\boldsymbol{\mu}},\underline{\boldsymbol{\mu}}\in\mathcal{R}^N_+,\boldsymbol{\lambda}\in\mathcal{R}^J} \; \underline{\boldsymbol{y}}^T\underline{\boldsymbol{\mu}} - \overline{\boldsymbol{y}}^T\overline{\boldsymbol{\mu}} + (\boldsymbol{w} - \boldsymbol{U}\boldsymbol{x})^T\boldsymbol{\lambda} \tag{2a}$$

$$\text{s.t. } \boldsymbol{V}^T\boldsymbol{\lambda} = \boldsymbol{c} + \overline{\boldsymbol{\mu}} - \underline{\boldsymbol{\mu}} + \boldsymbol{B}^T x \tag{2b}$$

The first step is to multiply the lower level primal constraints (1e) by $\boldsymbol{\lambda}$ component-wise:

$$\boldsymbol{V}\boldsymbol{y}\circ\boldsymbol{\lambda} = \boldsymbol{w}\circ\boldsymbol{\lambda} - \boldsymbol{U}\boldsymbol{x}\circ\boldsymbol{\lambda} \tag{3}$$

where $\circ$ denotes the Hadamard product.[2]

Second, the dual constraints (2b) are multiplied with $\boldsymbol{y}$ as follows:

$$(\boldsymbol{V}^\mathsf{T}\boldsymbol{\lambda})\circ\boldsymbol{y} = \boldsymbol{c}\circ\boldsymbol{y} + \overline{\boldsymbol{\mu}}\circ\boldsymbol{y} - \underline{\boldsymbol{\mu}}\circ\boldsymbol{y} + (\boldsymbol{B}^\mathsf{T}\boldsymbol{x})\circ\boldsymbol{y}. \tag{4}$$

Note that any $\overline{\mu}_n y_n$ can be linearized because of the upper bound

$$y_n \le \overline{y}_n. \tag{5}$$

The complementary slackness condition for (5) allows one to linearize $\overline{\mu}_n y_n$:

$$\overline{\mu}_n y_n = \overline{\mu}_n \overline{y}_n. \tag{6}$$

A similar result follows for any $\underline{\mu}_n y_n$. Combining the last result with the complementary slackness conditions gives:

$$(\boldsymbol{V}^\mathsf{T}\boldsymbol{\lambda})\circ\boldsymbol{y} = \boldsymbol{c}\circ\boldsymbol{y} + \overline{\boldsymbol{\mu}}\circ\overline{\boldsymbol{y}} - \underline{\boldsymbol{\mu}}\circ\underline{\boldsymbol{y}} + (\boldsymbol{B}^\mathsf{T}\boldsymbol{x})\circ\boldsymbol{y}. \tag{7}$$

[1]It is important to note that in some cases good bounds, which are necessary for the "big M" constraints used to linearize the product of integer and continuous variables, cannot be found [12].

[2]Note that one can also multiply each of the primal constraints by each of the components of $\boldsymbol{\lambda}$ to get J^2 equations. However, in practice the bilinear terms that appear in the upper level problem are bilinear in y_n and λ_j, where λ_j is the Lagrange multiplier of the constraint that involves y_n.

Equations (3) and (7) are then combined to produce a system of equations with the bilinear products of $\boldsymbol{\lambda}$ and $\boldsymbol{y}$ as the unknowns. In the following we show how to solve for a specific $\lambda_j V_{jn} y_n$.

Let the i^{th} row of (3) be defined as (P_i), which can be written:

$$(P_i) : \lambda_i V_{in} y_n = w_i \lambda_i - \lambda_i \sum_{k \in \mathcal{N} \setminus \{n\}} V_{ik} y_k - \lambda_i \sum_{m \in \mathcal{M}} U_{im} x_m \quad (8)$$

And, let the k^{th} row of (7) be defined as (D_k), which can be written:

$$(D_k) : y_k \sum_{i \in \mathcal{J}} V_{ik} \lambda_i = c_k y_k + \overline{\mu}_k \overline{y}_k - \underline{\mu}_k \underline{y}_k + y_k \sum_{m \in \mathcal{M}} B_{mk} x_m \quad (9)$$

Note that the choice of P for (P_i) and D for (D_k) are intentional: P is for *primal* constraints and D is for *dual* constraints.

Algorithm 1 outlines the procedure for determining the minimum set of the (P_i) and (D_k) equations needed to linearize a given $\lambda_j y_n$ term. Note that the algorithm refers to the indices of (P_i) as rows and (D_k) as columns because the sums over V_{jk} in (8) and (9) are over the rows and columns of $\boldsymbol{V}$ respectively.

The first step of Algorithm 1 is to check if V_{jn} is the only non-zero value in the n^{th} column of $\boldsymbol{V}$: in this case (D_n) provides the exact linearization of $\lambda_j y_n$ (and (P_i) is unnecessary):

$$y_n \lambda_j = \frac{1}{V_{jn}} \left(c_n y_n + \overline{\mu}_n \overline{y}_n - \underline{\mu}_n \underline{y}_n + y_n \sum_{m \in \mathcal{M}} B_{mn} x_m \right) \quad (10)$$

Note that (10) only applies under the condition that y_n is in a single lower level primal constraint. Additionally, the bilinear products of y_n and x_m in (10) can be linearized since we are assuming that $\boldsymbol{x}$ is integer in this section.

In the second step of Algorithm 1 the first primal equation (P_j) is added to the set of row indices that will be returned at the end of the algorithm (where j is an input). Additionally, for all the non-zero values in the j^{th} row of $\boldsymbol{V}$, except V_{jn}, the indices of the dual equations (D_k) are added to the set of column indices. In mathematical terms, this step is taking (P_j):

$$\lambda_j V_{jn} y_n = w_j \lambda_j - \lambda_j \sum_{k \in \mathcal{N} \setminus \{n\}} V_{jk} y_k - \lambda_j \sum_{m \in \mathcal{M}} U_{jm} x_m \quad (11)$$

and all of the (D_k) equations for $k \in \mathcal{N} \setminus \{n\}$ in order to replace the bilinear terms of λ_j and y_k on the right-hand-side of (11). Each (D_k) equation can add more bilinear terms of $\boldsymbol{\lambda}$ and $\boldsymbol{y}$ and so step three of Algorithm 1 adds additional equations if necessary.

In the third and final step of Algorithm 1 a recursive function, Algorithm 2, is used to search the array $\boldsymbol{V}$ for non-zero, "connected" values. We use the term "connected" to indicate that one could draw horizontal and vertical paths through $\boldsymbol{V}$ to connect non-zero entries to the first entry of interest V_{jn}, starting with a horizontal line each time. A horizontal line adds a (P_i) equation and a vertical line adds a (D_k) equation. The indices of the rows and columns are collected until a sufficient amount of equations are obtained to linearize the $\lambda_j y_n$ term in the upper level objective.

Note that Algorithm 2 is similar to — but not the same as — finding the blocks of a block-diagonal matrix. The difference is that Algorithm 2 does not necessarily find *all* of the non-zero values in a block. In other words, one does not need all of the (P_i) and (D_k) equations that may be available; one only needs as many equations as unknowns (where the unknowns are products of $\boldsymbol{\lambda}$ and $\boldsymbol{y}$ entries). Algorithm 2 has some conditions under which it returns as error: these errors occur when the search has indicated that redundant row or column indices should be appended to the final vectors. Mathematically, these

Algorithm 1: Minimum set of equations to linearize $\lambda_j y_n$

```
input : The 2D array $V$; and the integers (j, n) of
        non-zero $V_{jn}$.
output: Indices of $(P_i)$ and $(D_k)$ necessary to
        linearize a $\lambda_j y_n$ term.
1. if $V_{j'n} = 0 \; \forall j' \in \mathcal{J} \setminus \{j\}$ then
    return {},{n} (only need $D_n$)
end
2. Initialize arrays of integers for the rows and
   columns:
    $\mathcal{J}_j$ = {j}
    cols_to_check = $\{k \in \mathcal{N} \setminus \{n\} : V_{jk} \neq 0\}$
    $\mathcal{N}_n$ = copy(cols_to_check)
3. Recursive search to find all connections
foreach k in cols_to_check do
        rows, cols = recursive_array_search(V, j, k, {},
          {})
        $\mathcal{J}_j \leftarrow \mathcal{J}_j \cup$ rows
        $\mathcal{N}_n \leftarrow \mathcal{N}_n \cup$ cols
end
return $\mathcal{J}_j$, $\mathcal{N}_n$
```

Algorithm 2: recursive_array_search

```
input : The 2D array $V$; integers row $j$ and column
        $k$; and two vectors of integers to append to:
        rows and cols.
output: Two vectors of integers for the non-zero
        entries of $V$ connected to row $j$ and column $k$.
rs = { $j' \in \mathcal{J} \setminus \{j\} : V_{j'k} \neq 0$ }
if rs ∩ rows ≠ Ø then
    return error: redundant row
end
rows ← rows ∪ rs
foreach r ∈ rs do
    cs = { $k' \in \mathcal{N} \setminus \{k\} : V_{rk'} \neq 0$ }
    if {cs ∩ cols} ≠ Ø then
        return error: redundant column
    end
    cols ← cols ∪ cs
    foreach c ∈ cs do
        recursive_array_search(V, r, c, rows, cols)
    end
end
return rows, cols
```

errors indicate that there are more unknowns than equations and thus the system of equations is underdetermined.

Let the indices of (P_i) and (D_k) returned from Algorithm 1 for a given $(j,n) \in \mathcal{A}$ pair be defined as $\mathcal{J}_j$ and $\mathcal{N}_n$ respectively. The exact linearization of $\lambda_j y_n$ is:

$$\begin{aligned}\lambda_j y_n =& \frac{1}{V_{jn}}\left[\sum_{j'\in\mathcal{J}_j}\left(w_{j'}\lambda_{j'} - \lambda_{j'}\sum_{m\in\mathcal{M}}U_{j'm}x_m\right)\right.\\ &-\sum_{n'\in\mathcal{N}_n}\left(c_{n'}y_{n'} + \overline{\mu}_{n'}\overline{y}_{n'} - \underline{\mu}_{n'}\underline{y}_{n'}\right.\\ &\left.\left.+y_{n'}\sum_{m\in\mathcal{M}}B_{mn'}x_m\right)\right],\end{aligned} \tag{12}$$

which is simply a combination of (8) and (9) for all of the non-zero values of $\boldsymbol{V}$ connected to $\lambda_j y_n$, as demonstrated with the examples in Appendix A.

Finally, using the result (12) the mixed integer linear form of (1) is:

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda},\overline{\boldsymbol{\mu}},\underline{\boldsymbol{\mu}}}\ & f(\boldsymbol{x},\boldsymbol{y})\\ &+\sum_{(j,n)\in\mathcal{A}}\frac{A_{jn}}{V_{jn}}\left[\sum_{j'\in\mathcal{J}_j}\left(w_{j'}\lambda_{j'} - \lambda_{j'}\sum_{m\in\mathcal{M}}U_{j'm}x_m\right)\right.\\ &-\sum_{n'\in\mathcal{N}_n}\left(c_{n'}y_{n'} + \overline{\mu}_{n'}\overline{y}_{n'} - \underline{\mu}_{n'}\underline{y}_{n'}\right.\\ &\left.\left.+y_{n'}\sum_{m\in\mathcal{M}}B_{mn'}x_m\right)\right]\end{aligned} \tag{13a}$$

$$\text{s.t. } g(\boldsymbol{x},\boldsymbol{y}) \le 0, \tag{13b}$$
$$\boldsymbol{c} + \boldsymbol{B}^\intercal\boldsymbol{x} + \boldsymbol{V}^\intercal\boldsymbol{\lambda} + \overline{\boldsymbol{\mu}} - \underline{\boldsymbol{\mu}} = \boldsymbol{0} \tag{13c}$$
$$\underline{\boldsymbol{y}} \le \boldsymbol{y} \le \overline{\boldsymbol{y}} \tag{13d}$$
$$\boldsymbol{U}\boldsymbol{x} + \boldsymbol{V}\boldsymbol{y} = \boldsymbol{w} \tag{13e}$$
$$\overline{\boldsymbol{\mu}} \perp (\boldsymbol{y} - \overline{\boldsymbol{y}}) \tag{13f}$$
$$\underline{\boldsymbol{\mu}} \perp (\underline{\boldsymbol{y}} - \boldsymbol{y}) \tag{13g}$$

where the lower level problem has been replaced with the Karush Kuhn Tucker (KKT) conditions and the complementary constraints can be modeled as special order sets or using the "big M" method from [13]. Note that this section assumes that the $\boldsymbol{x}$ variables are integer and therefore the products of x_m and y_n or λ_j can be linearized using binary expansion [11].

It is important to note that finding valid values for the "big M" can be difficult [14]. The open source package in which the linearization method presented in this paper is implemented includes the options to use big M (Fortuny-McCarl) constraints, special order sets, or indicator constraints to linearize the complementary conditions used to integrate the lower level problem into the upper level. The latter two methods do not require defining bounds for the dual variables, but may be more difficult to solve than the "big M" method.

III. Linearization Method with Continuous Upper Level Variables

In Section II we assumed that $\boldsymbol{x}$ are integer such that all of the products of x_m and y_n or products of x_m and λ_j can be linearized using binary expansion [11]. Here we show the conditions under which a $\lambda_j y_n$ term can be linearized when the upper level variables $\boldsymbol{x}$ are continuous.

The conditions are divided into two groups with one group less restrictive than the other. The first group of conditions is less restrictive but does not allow lower level variables $\boldsymbol{y}$ to be bilinear in both the upper level problem with $\boldsymbol{\lambda}$ and the lower objective with $\boldsymbol{x}$. In mathematical terms this is when $\mathcal{AB} = \emptyset$, where $\mathcal{AB} \triangleq \{(j,n) \in \mathcal{A} : \exists m \in \mathcal{M} \text{ such that } B_{mn} \neq 0\}$.

The second group of conditions allows a problem to be linearized when bilinear products of $\lambda_j y_n$ are in the upper level *and* bilinear products of $x_m y_n$ are in the lower level objective for a given n. These types of problems are particularly relevant to energy system market models, in which the upper and lower level bilinear products together represent a zero-sum game. Demonstrative examples are provided in Section IV, in which the upper level products of $\lambda_j y_n$ represent payments to distributed generator owners and the lower level products of $x_m y_n$ represent generator owner income.

A. Conditions when $\mathcal{AB} = \emptyset$

Recall that the Algorithms 1 and 2 provide the sets $\mathcal{J}_j$ for each λ_j in the upper level objective. Let $\mathcal{J}_\cup \triangleq \bigcup_{j\in\mathcal{A}_\mathcal{J}}\mathcal{J}_j$, which includes the indices of all the lower level constraints that are connected (via non-zero values of $\boldsymbol{V}$) to the λ_j terms in the upper level objective. Therefore, in order to eliminate all bilinear terms of the form $\lambda_j U_{jm} x_m$ in (13a) the following condition must be met:

Condition 1. $U_{jm} = 0\ \forall j \in \mathcal{J}_\cup,\ \forall m \in \mathcal{M}$

Similar to Condition 1, let $\mathcal{N}_\cup \triangleq \bigcup_{n\in\mathcal{A}_\mathcal{N}}\mathcal{N}_n$, then one could assume that

Condition 2. $B_{mn} = 0\ \forall m \in \mathcal{M},\ \forall n \in \mathcal{N}_\cup$

to eliminate all bilinear terms of the form $x_m B_{mn} y_n$ from (13a). Under Conditions 1 and 2 the mixed integer result for (1) is

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda},\overline{\boldsymbol{\mu}},\underline{\boldsymbol{\mu}}}\ f(\boldsymbol{x},\boldsymbol{y}) + &\sum_{(j,n)\in\mathcal{A}}\frac{A_{jn}}{V_{jn}}\left[\sum_{j'\in\mathcal{J}_j}(w_{j'}\lambda_{j'})\right.\\ &\left.-\sum_{n'\in\mathcal{N}_n}\left(c_{n'}y_{n'} + \overline{\mu}_{n'}\overline{y}_{n'} - \underline{\mu}_{n'}\underline{y}_{n'}\right)\right]\end{aligned} \tag{14a}$$

$$\text{s.t. } g(\boldsymbol{x},\boldsymbol{y}) \le 0, \tag{14b}$$
$$\boldsymbol{c} + \boldsymbol{B}^\intercal\boldsymbol{x} + \boldsymbol{V}^\intercal\boldsymbol{\lambda} + \overline{\boldsymbol{\mu}} - \underline{\boldsymbol{\mu}} = \boldsymbol{0} \tag{14c}$$
$$\underline{\boldsymbol{y}} \le \boldsymbol{y} \le \overline{\boldsymbol{y}} \tag{14d}$$
$$\boldsymbol{U}\boldsymbol{x} + \boldsymbol{V}\boldsymbol{y} = \boldsymbol{w} \tag{14e}$$
$$\overline{\boldsymbol{\mu}} \perp (\boldsymbol{y} - \overline{\boldsymbol{y}}) \tag{14f}$$
$$\underline{\boldsymbol{\mu}} \perp (\underline{\boldsymbol{y}} - \boldsymbol{y}) \tag{14g}$$

B. Conditions when $\mathcal{AB} \neq \emptyset$

The case when Condition 2 is violated and Problem (1) has bilinear terms in the upper and lower level objectives of the form $\lambda_j A_{jn} y_n$ and $x_m B_{mn} y_n$, (where $A_{jn} \neq 0$ and $B_{mn} \neq 0$), for some n is particularly relevant to energy system market models. For example, take the case where $A_{jn} = 1$ and $B_{mn} = -1$ for some particular m, j, and n. Let y_n represent a lower level generator dispatch decision. Then λ_j represents the marginal cost of the dispatch decision y_n as well as the upper level's cost of purchasing power from the lower level. And $-x_m y_n$ in the lower level objective is the lower level's income for the generation y_n using the price signal x_m. Section IV provides an example of such a scenario. Thus it is useful to investigate the linearization of problems when $\mathcal{AB} \neq \emptyset$ (i.e. when Condition 2 is violated).

The problem of interest has the following structure:

$$\min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{x},\boldsymbol{y}) + \sum_{(j,n)\in\mathcal{A}} \lambda_j A_{jn} y_n \tag{15a}$$

$$\text{s.t. } g(\boldsymbol{x},\boldsymbol{y}) \le 0 \tag{15b}$$

$$\begin{aligned}\boldsymbol{y} \in \arg\min_{\boldsymbol{y}}\ & \boldsymbol{c}^\intercal \boldsymbol{y} + \sum_{m\in\mathcal{M}} \sum_{n\in\mathcal{A_N}} x_m B_{mn} y_n \\ &+ \sum_{m\in\mathcal{M}} \sum_{n\in\mathcal{N}\setminus(\mathcal{A_N}\cup\mathcal{N}_\cup)} x_m B_{mn} y_n \end{aligned} \tag{15c}$$

$$\text{s.t. } \underline{\boldsymbol{y}} \le \boldsymbol{y} \quad (\underline{\boldsymbol{\mu}}) \tag{15d}$$

$$\boldsymbol{y} \le \overline{\boldsymbol{y}} \quad (\overline{\boldsymbol{\mu}}) \tag{15e}$$

$$\sum_{n\in\mathcal{N}} V_{jn} y_n = w_j \quad (\lambda_j), \quad \forall j \in \mathcal{J}_\cup \tag{15f}$$

$$\sum_{m\in\mathcal{M}} U_{jm} x_m + \sum_{n\in\mathcal{N}} V_{jn} y_n = w_j \quad (\lambda_j), \quad \forall j \in \mathcal{J}\setminus\mathcal{J}_\cup. \tag{15g}$$

Note that the products of $\boldsymbol{x}$ and $\boldsymbol{y}$ in the lower level objective (15c) are linearized when the lower level problem is replaced with the KKT conditions. And the set of y_n for all $n \in \mathcal{N}\setminus(\mathcal{A_N}\cup\mathcal{N}_\cup)$ in the last sum of (15c) are the values of $\boldsymbol{y}$ that are *not* in the upper level objective nor connected to the $y_n, n \in \mathcal{A_N}$, in the upper level objective. Recall that the connected indices are provided by Algorithm 1 and captured in $\mathcal{N}_\cup$. We will show shortly that the connected y_n values must not be in the lower level objective with $\boldsymbol{x}$ terms to prevent $y_n x_m$ terms from showing up in the (D_k) equations needed to linearize the $\lambda_j y_n$ in the upper level objective. Also, Condition 1 is reflected in (15f).

Now, applying Condition 1 to (12) gives

$$\lambda_j y_n = \frac{1}{V_{jn}} \left[\sum_{j'\in\mathcal{J}_j} w_{j'}\lambda_{j'} - \sum_{n'\in\mathcal{N}_n} \left(c_{n'} y_{n'} + \overline{\mu}_{n'} \overline{y}_{n'} - \underline{\mu}_{n'} \underline{y}_{n'} + y_{n'} \sum_{m\in\mathcal{M}} B_{mn'} x_m \right) \right], \ \forall (j,n) \in \mathcal{A}. \tag{16}$$

We wish to eliminate the $y_{n'} B_{mn'} x_m$ terms when $\mathcal{AB} \neq \emptyset$. Recall that the $y_{n'} B_{mn'} x_m$ terms in (12) and (16) come from the (D_k) equations with $B_{mk} \neq 0$, and that the (D_k) equations for all $k \in \mathcal{N}_\cup$ are necessary to linearize the upper level $\lambda_j A_{jk} y_k$ terms.

Let us assume that a less restrictive version of Condition 2 holds:

Condition 2′. $B_{mn} = 0\ \forall m \in \mathcal{M},\ \forall n \in \mathcal{N}_\cup \setminus \mathcal{A_N}$

Condition 2′ implies that none of the lower level variables *connected* to the $\lambda_j A_{jn} y_n$ terms (provided by Algorithm 1) are in the lower level objective with $y_n B_{mn} x_m$ terms, except the lower level variables in the upper level objective ($y_n\ \forall n \in \mathcal{A_N}$). Applying Condition 2′ to (16) gives:

$$\begin{aligned}\lambda_j y_n = \frac{1}{V_{jn}} \Bigg[& \sum_{j'\in\mathcal{J}_j} w_{j'}\lambda_{j'} \\ & - \sum_{n'\in\mathcal{N}_n\setminus\mathcal{A_N}} \left(c_{n'} y_{n'} + \overline{\mu}_{n'} \overline{y}_{n'} - \underline{\mu}_{n'} \underline{y}_{n'} \right) \\ & - \sum_{n'\in\mathcal{N}_n\cap\mathcal{A_N}} \Bigg(c_{n'} y_{n'} + \overline{\mu}_{n'} \overline{y}_{n'} - \underline{\mu}_{n'} \underline{y}_{n'} \\ & \qquad + y_{n'} \sum_{m\in\mathcal{M}} B_{mn'} x_m \Bigg) \Bigg], \ \forall (j,n) \in \mathcal{A}. \end{aligned} \tag{17}$$

Let us also assume that Condition 3 holds:

Condition 3. $\mathcal{A_N} \setminus \{n\} \subseteq \mathcal{N}_n\ \forall n \in \mathcal{A_N}$

$$\Rightarrow \mathcal{N}_n \cap \mathcal{A_N} = \mathcal{A_N}\setminus\{n\}\ \forall n \in \mathcal{A_N}$$

Condition 3 implies that the $y_n\ \forall n \in \mathcal{A_N}$ variables of interest are connected to each other via non-zero values of $\boldsymbol{V}$. Section IV-B provides an example problem, in which the $y_n\ \forall n \in \mathcal{A_N}$ are indexed on time and connected to each other via another time-indexed variable in each equality constraint that is restricted to be no more than a certain value across all time.

Condition 3 allow us to rewrite (17) as

$$\begin{aligned}\lambda_j y_n = \frac{1}{V_{jn}} \Bigg[& \sum_{j'\in\mathcal{J}_j} w_{j'}\lambda_{j'} \\ & - \sum_{n'\in\mathcal{N}_n\setminus\mathcal{A_N}} \left(c_{n'} y_{n'} + \overline{\mu}_{n'} \overline{y}_{n'} - \underline{\mu}_{n'} \underline{y}_{n'} \right) \\ & - \sum_{n'\in\mathcal{A_N}\setminus\{n\}} \Bigg(c_{n'} y_{n'} + \overline{\mu}_{n'} \overline{y}_{n'} - \underline{\mu}_{n'} \underline{y}_{n'} \\ & \qquad + y_{n'} \sum_{m\in\mathcal{M}} B_{mn'} x_m \Bigg) \Bigg], \ \forall (j,n) \in \mathcal{A}. \end{aligned} \tag{18}$$

Applying (9) to the last summation in (18) gives:

$$\begin{aligned}\lambda_j y_n = \frac{1}{V_{jn}} \Bigg[& \sum_{j'\in\mathcal{J}_j} w_{j'}\lambda_{j'} \\ & - \sum_{n'\in\mathcal{N}_n\setminus\mathcal{A_N}} \left(c_{n'} y_{n'} + \overline{\mu}_{n'} \overline{y}_{n'} - \underline{\mu}_{n'} \underline{y}_{n'} \right) \\ & - \sum_{n'\in\mathcal{A_N}\setminus\{n\}} \left(y_{n'} \sum_{j'\in\mathcal{J}} V_{j'n'} \lambda_{j'} \right) \Bigg], \ \forall (j,n) \in \mathcal{A}, \end{aligned} \tag{19}$$

This step is key to eliminating the bilinear $y_{n'}B_{mn'}x_m$ terms when $\mathcal{AB} \neq \emptyset$. The next steps are to impose conditions that allow us to move the last summation in (19) to the left hand side to get a single sum of terms over the set $\mathcal{A}$.

Let us assume that Condition 4 holds:

Condition 4. $V_{j'n} = 0 \; \forall j' \in \mathcal{J} \setminus \{j\}, \; \forall (j,n) \in \mathcal{A}$.

Condition 4 is equivalent to each y_n for all $n \in \mathcal{A_N}$ being in only one lower level constraint. Condition 4 implies that

$$y_k \sum_{j' \in \mathcal{J}} V_{j'k}\lambda_{j'} = \lambda_j V_{jk} y_k, \; \forall (j,k) \in \mathcal{A} \tag{20}$$

Note that Condition 4 requires that Step 1 of the Algorithm be skipped. The revised algorithm for Conditions 1, 2′, 3, and 4 is shown in Algorithm 3.

Algorithm 3: Minimum set of equations to linearize $\lambda_j y_n$ under Conditions 1, 2′, 3, and 4

```
input : The 2D array $V$; and the integers ($j$, $n$) of
        non-zero $V_{jn}$.
output: Indices of ($P_i$) and ($D_k$) necessary to
        linearize a $\lambda_j y_n$ term.
1. Initialize arrays of integers:
    $\mathcal{J}_j$ = $\{j\}$
    cols_to_check = $\{k \in \mathcal{N} \setminus \{n\} : V_{jk} \neq 0\}$
    $\mathcal{N}_n$ = copy(cols_to_check)
2. Recursive search to find all connections
foreach $k$ in cols_to_check do
    rows, cols = recursive_array_search(V, $j$, $k$, {}, {})
    $\mathcal{J}_j \leftarrow \mathcal{J}_j \cup$ rows
    $\mathcal{N}_n \leftarrow \mathcal{N}_n \cup$ cols
end
return $\mathcal{J}_j$, $\mathcal{N}_n$
```

Condition 4 allows us to write (19) as

$$\lambda_j y_n = \frac{1}{V_{jn}} \Bigg[\sum_{j' \in \mathcal{J}_j} w_{j'}\lambda_{j'} - \sum_{n' \in \mathcal{N}_n \setminus \mathcal{A_N}} \left(c_{n'}y_{n'} + \overline{\mu}_{n'}\overline{y}_{n'} - \underline{\mu}_{n'}\underline{y}_{n'} \right) - \sum_{(j',n') \in \mathcal{A} \setminus \{(j,n)\}} \left(y_{n'}V_{j'n'}\lambda_{j'} \right) \Bigg], \; \forall (j,n) \in \mathcal{A}, \tag{21}$$

Rearranging (21) gives:

$$\sum_{(j',n') \in \mathcal{A}} \lambda_{j'}V_{j'n'}y_{n'} = \sum_{j' \in \mathcal{J}_j} w_{j'}\lambda_{j'} - \sum_{n' \in \mathcal{N}_n \setminus \mathcal{A_N}} \left(c_{n'}y_{n'} + \overline{\mu}_{n'}\overline{y}_{n'} - \underline{\mu}_{n'}\underline{y}_{n'} \right), \; \forall (j,n) \in \mathcal{A}. \tag{22}$$

Note that since (22) is valid for all $(j,n) \in \mathcal{A}$ it implies that $\mathcal{N}_n$ are equal for all $n \in \mathcal{A_N}$ and that $\mathcal{J}_j$ are equal for all $j \in \mathcal{A_J}$.

Lastly, we see that to replace $\sum_{(j,n) \in \mathcal{A}} \lambda_j A_{jn} y_n$ with the last result for $\sum_{(j,n) \in \mathcal{A}} \lambda_j V_{jn} y_n$ we must require that the two sums are equal to a proportionality constant p:

Condition 5. $A_{jn} = pV_{jn} \; \forall (j,n) \in \mathcal{A}$

$$\Rightarrow \sum_{(j,n) \in \mathcal{A}} \lambda_j A_{jn} y_n = p \sum_{(j,n) \in \mathcal{A}} \lambda_j V_{jn} y_n.$$

With Condition 5 we can write (22) as:

$$\sum_{(j,n) \in \mathcal{A}} \lambda_j A_{jn} y_n = p \Bigg[\sum_{j' \in \mathcal{J}_j} w_{j'}\lambda_{j'} - \sum_{n' \in \mathcal{N}_n \setminus \mathcal{A_N}} \left(c_{n'}y_{n'} + \overline{\mu}_{n'}\overline{y}_{n'} - \underline{\mu}_{n'}\underline{y}_{n'} \right) \Bigg] \; \forall (j,n) \in \mathcal{A}. \tag{23}$$

Substituting (23) into (1) and replacing the lower level with the KKT conditions, under Conditions 1, 2′, 3, 4 and 5 the mixed integer result is shown in (24). Note that any $(j,n) \in \mathcal{A}$ can be used in (24a) to define the sets $\mathcal{J}_j$ and $\mathcal{N}_n$.

$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda},\overline{\boldsymbol{\mu}},\underline{\boldsymbol{\mu}}} f(\boldsymbol{x},\boldsymbol{y}) + p \Bigg[\sum_{j' \in \mathcal{J}_j} (w_{j'}\lambda_{j'}) - \sum_{n' \in \mathcal{N}_n \setminus \mathcal{A_N}} \left(c_{n'}y_{n'} + \overline{\mu}_{n'}\overline{y}_{n'} - \underline{\mu}_{n'}\underline{y}_{n'} \right) \Bigg] \tag{24a}$$

$$\text{s.t. } g(\boldsymbol{x},\boldsymbol{y}) \leq 0, \tag{24b}$$
$$\boldsymbol{c} + \boldsymbol{B}^\intercal \boldsymbol{x} + \boldsymbol{V}^\intercal \boldsymbol{\lambda} + \overline{\boldsymbol{\mu}} - \underline{\boldsymbol{\mu}} = \boldsymbol{0} \tag{24c}$$
$$\underline{\boldsymbol{y}} \leq \boldsymbol{y} \tag{24d}$$
$$\boldsymbol{y} \leq \overline{\boldsymbol{y}} \tag{24e}$$
$$\boldsymbol{Ux} + \boldsymbol{Vy} = \boldsymbol{w} \tag{24f}$$
$$\overline{\boldsymbol{\mu}} \perp (\boldsymbol{y} - \overline{\boldsymbol{y}}) \tag{24g}$$
$$\underline{\boldsymbol{\mu}} \perp (\underline{\boldsymbol{y}} - \boldsymbol{y}) \tag{24h}$$

To summarize all of the conditions under which (24) is valid:

- Condition 1: $U_{jm} = 0 \; \forall j \in \mathcal{J}_\cup, \; \forall m \in \mathcal{M}$
  - None of the connected constraints contain $\boldsymbol{x}$ terms.
- Condition 2′: $B_{mn} = 0 \; \forall m \in \mathcal{M}, \; \forall n \in \mathcal{N}_\cup \setminus \mathcal{A_N}$
  - None of the connected variables are multiplied with $\boldsymbol{x}$ in the lower level objective, except the $\boldsymbol{y}$ in the upper level objective that are multiplied with $\boldsymbol{\lambda}$.
- Condition 3: $\mathcal{A_N} \setminus \{n\} \subseteq \mathcal{N}_n \; \forall n \in \mathcal{A_N}$
  - Each of the y_n in the upper level objective are connected to each other via non-zero values of $\boldsymbol{V}$.
- Condition 4 $V_{j'n} = 0 \; \forall j' \in \mathcal{J} \setminus \{j\}, \; \forall j \in \mathcal{A_J}$
  - Each of the y_n in the upper level objective are in only one lower level equality constraint.
- Condition 5 $A_{jn} = pV_{jn} \; \forall (j,n) \in \mathcal{A}$
  - All of the coefficients of the upper level $\lambda_j y_n$ terms are proportional to the corresponding coefficients in the lower level constraints to the same constant p.

The examples in Section IV meet the Conditions 1, 2′, 3, 4 and 5. The theme in both problems is an energy market model with a load balance constraint in the lower level, bilinear products in the upper level objective of lower level dispatch variables and the load balance dual variables, and bilinear products in the lower level objective of upper level price signal variables and the same lower level dispatch variables as in the upper level objective.

C. A note on separable lower level problems

It is important to note that the conditions required to linearize the bilinear products of shadow prices and primal variables can be applied to sub-matrices when the lower level problem is separable. For example, in a multi-follower Stackelberg game the lower level is likely to be separable, such as when modeling multiple distributed energy reosource (DER) owners or grid customers. In these cases Conditions 3 and 5 should be checked against the blocks of $\boldsymbol{A}$ and $\boldsymbol{V}$ corresponding to each sub-problem. An example of a separable lower level problem is provided in Section IV

IV. Use-case examples

It is important to note that the use of the linearization algorithm is not limited to the problem types shown in these examples. Indeed, the conditions required for the linearization algorithm are met in each of the references mentioned in Section I. For example, other use cases include:

- optimizing generator offer curves in an energy market [2];
- applying generator revenue constraints in a market clearing process [3];
- and optimal profit sharing in a community microgrid [6].

A. Simple examples without time indices

The first use-case example shows a step-by-step linearization process for a scenario in which the upper level model minimizes the total cost of power with the option to purchase power from the bulk system or from customer owned DER. The lower level can choose to meet its demand from the grid at some retail rate or invest in DER to lower its total cost.

$$\min_{\boldsymbol{x},\boldsymbol{y}} \; c_{\text{LMP}} x_0 + \lambda y_e \tag{25a}$$
$$\text{s.t. } x_0 + y_e - y_i - d_2 = 0 \tag{25b}$$
$$x_\lambda \geq 0 \tag{25c}$$
$$0 \leq y_e \perp y_i \geq 0 \tag{25d}$$
$$\boldsymbol{y} \in \arg\min_{\boldsymbol{y}} \; c_{\text{DER}} y_{\text{DER}} + c_i y_i - x_\lambda y_e \tag{25e}$$
$$\text{s.t. } y_i - y_e + y_{\text{DER}} = d_1 \quad (\lambda) \tag{25f}$$
$$\overline{y}_{\text{DER}} \geq y_{\text{DER}} \geq 0, \; (\overline{\mu}_{\text{DER}}, \underline{\mu}_{\text{DER}}) \tag{25g}$$
$$\overline{y}_e \geq y_e \geq 0 \; (\overline{\mu}_e, \underline{\mu}_e) \tag{25h}$$
$$\overline{y}_i \geq y_i \geq 0 \; (\overline{\mu}_i, \underline{\mu}_i). \tag{25i}$$

In example (25) the upper level (UL) can purchase power x_0 at the feeder head at the wholesale price c_{LMP} and/or from the lower level (LL) at a price of the UL's choosing. The UL chooses x_λ to set the LL's marginal cost of power λ when the LL chooses to export power y_e ("e" for export). The LL considers buying power y_i from grid at the retail rate c_i ("i" for import) and/or purchasing the DER capacity y_{DER} at the cost c_{DER} to meet its demand d_1. Constraint (25b) is the system load balance, which includes an uncontrollable demand d_2 (in practice this constraint is replaced with a power flow model). Constraint (25d) is the UL enforcement of no simultaneous export and import[3]. Constraint (25f) is the LL's load balance. The lower level dual variables, μ_{DER}, μ_e, and μ_i, are show in parentheses.

Let $\boldsymbol{y} \triangleq [y_e, y_i, y_{\text{DER}}]^\intercal$. The lower level has one equality constraint, making $\boldsymbol{V} = [-1 \; 1 \; 1]$. In this case we wish to linearize the product λy_e, making the indices of interest $j = 1$ for the first and only equality constraint in the lower level, and $n = 1$ because we put y_e in the first index of $\boldsymbol{y}$.

First, let us check the linearization conditions for this problem. Note that the set $\mathcal{AB}$ is not empty because we have a bilinear term in the upper and lower level objectives of the form $\lambda_j A_{jn} y_n$ and $x_m B_{mn} y_n$. Thus, we must check the Conditions 1, 2′, 3, 4, and 5. To check the conditions we need the sets $\mathcal{J}_\cup$, $\mathcal{N}_\cup$, $\mathcal{A}$ and their sub-sets $\mathcal{J}_j$, $\mathcal{N}_n$, $\mathcal{A}_\mathcal{N}$. With $\mathcal{N} = \{1, 2, 3\}$ for the three LL variables and $\mathcal{J} = \{1\}$ for the single LL constraint, we get:

- $\mathcal{A} = \{(j, n) \in \mathcal{J} \times \mathcal{N} : A_{jn} \neq 0\} = \{(1, 1)\}$
- $\mathcal{A}_\mathcal{N} = \{n \in \mathcal{N} : \exists j \in \mathcal{J} \text{ such that } A_{jn} \neq 0\} = \{1\}$

With only one pair of (j, n) in $\mathcal{A}$ we only need to call Algorithm 3 once with the values for $\boldsymbol{V}$, $j = 1$, and $n = 1$. Algorithm 3 first initializes $\mathcal{J}_1 = \{1\}$ and defines the cols_to_check as $\{2, 3\}$ because $V_{1,2}$ and $V_{1,3}$ are non-zero. The cols_to_check is copied to start the set $\mathcal{N}_1$ and then the recursive search is started. The recursive search loops over the values in cols_to_check and calls Algorithm 2, appending the results to the sets $\mathcal{J}_1$ and $\mathcal{N}_1$. In this case no new non-zero values are found (there is only one row in $\boldsymbol{V}$) and so Algorithm 2 returns the values that it was provided. Finally, Algorithm 3 returns the sets $\mathcal{J}_1 = \{1\}$ and $\mathcal{N}_1 = \{2, 3\}$.

Now, since we only have one pair of (j, n) in $\mathcal{A}$ the union sets $\mathcal{J}_\cup$ and $\mathcal{N}_\cup$ are equal to the sets $\mathcal{J}_1$ and $\mathcal{N}_1$ respectively. With the necessary sets defined we can use (23) to linearize the product of λ and y_e in the UL objective:

$$\begin{aligned} \lambda_e(-1)y_e &= d_1\lambda - (c_g y_{\text{DER}} + \overline{y}_{\text{DER}}\overline{\mu}_{\text{DER}} + c_i y_i + \overline{y}_i\overline{\mu}_i) \\ \Rightarrow \lambda y_e &= c_{\text{DER}} y_{\text{DER}} + \overline{y}_{\text{DER}}\overline{\mu}_{\text{DER}} + c_i y_i + \overline{y}_i\overline{\mu}_i - d_1\lambda. \end{aligned} \tag{26}$$

With this last result, we replace the lower level problem in (25) with its KKT conditions to get the mixed integer linear

[3]Allowing simultaneous power import and export would require two isolated meters: one measuring demand and one measuring DER production.

program:

$$\min_{\boldsymbol{x},\boldsymbol{y}} \; c_{\text{LMP}} x_0 + c_{\text{DER}} y_{\text{DER}} + \overline{y}_{\text{DER}} \overline{\mu}_{\text{DER}} + c_i y_i + \overline{y}_i \overline{\mu}_i - d_1 \lambda \tag{27a}$$

$$\text{s.t. } x_0 + y_e - y_i - d_2 = 0 \tag{27b}$$
$$x_\lambda \geq 0 \tag{27c}$$
$$0 \leq y_e \perp y_i \geq 0 \tag{27d}$$
$$y_i - y_e + y_{\text{DER}} = d_1 \quad (\lambda) \tag{27e}$$
$$y_{\text{DER}} \geq 0, y_e \geq 0, y_i \geq 0 \tag{27f}$$
$$-x_\lambda + \lambda - \mu_e = 0 \tag{27g}$$
$$c_i - \lambda - \mu_i = 0 \tag{27h}$$
$$c_{\text{DER}} - \lambda - \mu_{\text{DER}} = 0 \tag{27i}$$
$$0 \leq y_{\text{DER}} \perp \underline{\mu}_{\text{DER}} \geq 0 \tag{27j}$$
$$0 \leq y_e \perp \underline{\mu}_e \geq 0 \tag{27k}$$
$$0 \leq y_i \perp \underline{\mu}_i \geq 0 \tag{27l}$$
$$y_{\text{DER}} - \overline{y}_{\text{DER}} \perp \overline{\mu}_{\text{DER}} \geq 0 \tag{27m}$$
$$y_e - \overline{y}_e \perp \overline{\mu}_e \geq 0 \tag{27n}$$
$$y_i - \overline{y}_i \perp \overline{\mu}_i \geq 0 \tag{27o}$$

Example (25) (and its mixed-integer linear version (27)) is useful for showing how DER can benefit system operators. First, let us assume that the DER system has a relatively high cost of 10 \$/MW when compared to the other cost values, which are $c_{\text{LMP}} = 1$ \$/MW, $c_i = 1$ \$/MW, $d_1 = 1$ MW, and $d_2 = 2$ MW. In this case it is not in the LL's interest to buy DER and so $y_{\text{DER}} = 0$ and the LL purchases all of its power $d_1 = 1$ from the grid. Also, the UL purchases all power from the bulk system at $c_{\text{LMP}} = 1$ to meet the total demand $d_1 + d_2 = 3$ MW. Therefore, the UL's cost is \$3 and the LL's cost is \$1.

Now, let us assume that the DER system cost is equivalent to the other cost values at 1 \$/MW. The LL can now meet its demand for equivalent costs from either the grid or from a DER system. However, it is in the UL's best interest for demand to be met by the LL's DER system because the UL can lower its total cost from \$3 to \$2 by paying the LL \$2 for exporting excess DER power in to the grid to meet demand d_2 instead of meeting the total demand $d_1 + d_2$ from the bulk system. Therefore, the UL chooses $x_\lambda = 1$ \$/MW, which incentivizes the LL to purchase $y_{\text{DER}} = 3$ MW. The LL meets its demand $d_1 = 1$ MW behind-the-meter and exports 2 MW, which meets demand $d_2 = 2$ MW (and $x_0 = 0$ MW). The LL's cost is \$1 (the same as in the high DER cost scenario), but the UL reduces its cost from \$3 to \$2.

In summary, in this simple demonstration, only when the marginal cost of power from DER for the LL is less than (or equal to) retail rate will the LL purchase DER, which allows the UL to purchase excess power. When the LL can export excess power, and the UL can lower its total cost by purchasing DER exports, the UL will set the LL's marginal cost of power by choosing the minimum compensation rate to incentivize the LL to export the optimal amount of power that minimizes the total system cost of power. Note that in practice the decision variables are indexed on time; and, with solar PV as a DER option, there can be times when the LL has a zero marginal cost of power. Therefore, determining the DER solutions in practice are not as simple as comparing the cost coefficients.

B. Complex example with separable lower level problem

In this planning example we have a distribution system planner in the upper level that is considering purchasing battery energy storage systems for installation at three different nodes in a distribution system in order to reduce its operating cost in a real-time energy market. The planner also accounts for purchasing exported PV power from customers and sending a time-of-use price signal to refrigerated warehouses with price-responsive cooling systems. The upper level model is shown in (28). Tables III and IV summarize the variables, parameters and sets in (28). The objective (28a) includes three components to minimize: (1) the cost of energy purchased on the bulk market at the feeder head; (2) the cost of energy purchased from distributed, customer-owned photovoltaic (PV) systems; and (3) the capital costs of battery systems. We assume an analysis period of 20 years and a discount rate of 5%. For the bulk market price $c_{\text{LMP},t}$ we use the average hourly real-time market prices from ERCOT over the year of 2019 [15]. A year of load is simulated at an hourly resolution by randomly assigning different U.S. Department of Energy Commercial Reference Building profiles to the load nodes [16]. The load nodes are defined in [17], from which we take the 38 node network model. Constraints (28c) – (28g) define a linear power flow model, commonly known as "LinDistFlow" [18]. Constraint (28h) limits the squared voltage magnitude. Constraints (28i) and (28j) define the net power injection from system operator owned battery systems. Constraints (28k) and (28l) define the net power injection from customer nodes with PV systems. Constraints (28m) and (28n) define the net power injection from nodes with price-responsive refrigerated warehouses. Constraints (28o) and (28p) define the net power injection from nodes with uncontrollable load. Constraint (28q) is structural and prevents simultaneous export and import rom nodes with PV systems. Constraints (28r) – (28v) define the operational limits of the system operator's battery systems. Finally, constraint (28w) says that the lower level decisions y must be optimal for the lower level problem (29).

Problem (29) shows the lower level problem, with the objective to minimize the total cost of energy for all customers in the distribution system. Tables III and IV summarize the variables, parameters and sets in (29). The first half of the lower level objective (29a) represents the cost of energy for price responsive refrigerated warehouses that have a known retail rate $c_{i,n,t}$ and a time-varying price signal from the upper level problem $x_{i,n,t}$. The second half of (29a) represents the cost of energy for customers that can install PV systems. These customers also pay the retail rate $c_{i,n,t}$ for imported power but can also recieve compensation for exported, excess PV power from the upper level via $x_{e,n,t}$. Constraints (29b) and (29c) are the load balance constraints for the PV and warehouse customers respectively. Constraint (29d) limits the PV power used to meet load to the PV capacity times a known, normalized solar PV production factor from [19]. Constraints (29e) -

$$\min_{\boldsymbol{P},\boldsymbol{Q},\boldsymbol{w},\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda}} \text{pwf}\sum_{t\in\mathcal{T}}\left(c_{\text{LMP},t}x_{0,t}+\sum_{n\in\mathcal{N}_{\text{PV}}}\left[\lambda_{n,t}y_{e,n,t}\right]\right)+\sum_{n\in\mathcal{N}_{\text{B}}}\left(c_{\text{BkW}}x_{\text{BkW},n}+c_{\text{BkWh}}x_{\text{BkWh},n}\right) \tag{28a}$$

$$\text{s.t. } x_{0,t}\geq 0,\ x_{e,t}\geq 0,\ x_{i,t}\geq 0, \qquad \forall t\in\mathcal{T} \tag{28b}$$

$$P_{0,t}=P_{01,t} \qquad \forall t\in\mathcal{T} \tag{28c}$$

$$Q_{0,t}=Q_{01,t}, \qquad \forall t\in\mathcal{T} \tag{28d}$$

$$P_{ij,t}+P_{j,t}-\sum_{k:j\to k}P_{jk}=0, \qquad \forall j\in\mathcal{N}_+,\ \forall t\in\mathcal{T} \tag{28e}$$

$$Q_{ij,t}+Q_{j,t}-\sum_{k:j\to k}Q_{jk}=0, \qquad \forall j\in\mathcal{N}_+,\ \forall t\in\mathcal{T} \tag{28f}$$

$$w_{j,t}=w_{i,t}-2\left(r_{ij}P_{ij,t}+x_{ij}Q_{ij,t}\right), \qquad \forall j\in\mathcal{N}_+,\ \forall t\in\mathcal{T} \tag{28g}$$

$$(v_j^{\max})^2\geq w_{j,t}\geq (v_j^{\min})^2, \qquad \forall j\in\mathcal{N},\ \forall t\in\mathcal{T} \tag{28h}$$

$$P_{j,t}=x_{\text{B}^-,j,t}-x_{\text{B}^+,j,t}, \qquad \forall j\in\mathcal{N}_{\text{B}},\ \forall t\in\mathcal{T} \tag{28i}$$

$$Q_{j,t}=f_{pf,j,t}\left(x_{\text{B}^-,j,t}-x_{\text{B}^+,j,t}\right), \qquad \forall j\in\mathcal{N}_{\text{B}},\ \forall t\in\mathcal{T} \tag{28j}$$

$$P_{j,t}=y_{e,j,t}-y_{i,j,t}, \qquad \forall j\in\mathcal{N}_{\text{PV}},\ \forall t\in\mathcal{T} \tag{28k}$$

$$Q_{j,t}=f_{pf,j,t}\left(y_{e,j,t}-y_{i,j,t}\right), \qquad \forall j\in\mathcal{N}_{\text{PV}},\ \forall t\in\mathcal{T} \tag{28l}$$

$$P_{j,t}=-y_{i,j,t}, \qquad \forall j\in\mathcal{N}_W,\ \forall t\in\mathcal{T} \tag{28m}$$

$$Q_{j,t}=-f_{pf,j,t}y_{i,j,t}, \qquad \forall j\in\mathcal{N}_W,\ \forall t\in\mathcal{T} \tag{28n}$$

$$P_{j,t}=-d_{j,t}, \qquad \forall j\in\mathcal{N}_U,\ \forall t\in\mathcal{T} \tag{28o}$$

$$Q_{j,t}=-f_{pf,j,t}d_{j,t}, \qquad \forall j\in\mathcal{N}_U,\ \forall t\in\mathcal{T} \tag{28p}$$

$$y_{e,j,t}\perp y_{i,j,t}, \qquad \forall j\in\mathcal{N}_{\text{PV}},\ \forall t\in\mathcal{T} \tag{28q}$$

$$x_{\text{SOC},j,t}=x_{\text{SOC},j,t-1}+f_{hr}\left(x_{\text{B}^+,j,t}\eta-x_{\text{B}^-,j,t}/\eta\right) \qquad \forall j\in\mathcal{N}_{\text{B}},\ \forall t\in\mathcal{T} \tag{28r}$$

$$x_{\text{BkW},j}\geq x_{\text{B}^+,j,t}+x_{\text{B}^-,j,t} \qquad \forall j\in\mathcal{N}_{\text{B}},\ \forall t\in\mathcal{T} \tag{28s}$$

$$x_{\text{BkWh},j}\geq x_{\text{SOC},j,t} \qquad \forall j\in\mathcal{N}_{\text{B}},\ \forall t\in\mathcal{T} \tag{28t}$$

$$x_{\text{SOC},j,0}=0.5x_{\text{BkWh},j} \qquad \forall j\in\mathcal{N}_{\text{B}} \tag{28u}$$

$$x_{\text{SOC},j,N_t}=0.5x_{\text{BkWh},j} \qquad \forall j\in\mathcal{N}_{\text{B}} \tag{28v}$$

$$\boldsymbol{y}=\boldsymbol{y}^\star(\boldsymbol{x}) \tag{28w}$$

(29g) define the refrigerated warehouse temperature dynamics, starting condition, and temperature limits. We assume that the warehouses are have freezing units that can at most reach zero °C but can be cooled to as low as -20 °C in order to lower their energy costs by coasting through high price periods. We assume that the distribution system is in Austin, Texas, which defines the PV production factor and the ambient temperature used as an input to the refrigerated warehouse models.

By inspection the lower level model (29) is separable between the set of PV nodes $\mathcal{N}_{\text{PV}}$ and the warehouse nodes $\mathcal{N}_W$. Because we wish to linearize the bilinear product $\lambda_{n,t}y_{e,n,t}$ in (28a), which is only defined for the PV nodes, we only need to check the linearization conditions for the components of the lower level model (29) that are relevant to the PV nodes. A Julia module to programatically check the linearization conditons, including for separable problems like this example, is available in [20][4].

The upper level is allowed to install battery systems at up to three nodes (2, 7, and 24) in the network while the lower level can install PV systems on up to five nodes (9, 17, 22, 31, and 34). Using the baseline values the optimal solution is for the lower level customers to install small PV systems to reduce their utility bills and it is not economical for the upper level to install storage systems. To produce more interesting results we increase the bulk energy costs by integer values from the baseline $1\times$ to $5\times$. Table II summarizes the results with increasing bulk energy costs. In table II we can see some expected trends: as the bulk energy costs increase less energy is purchased on the bulk market and more PV energy is purchased from customers, which encourages larger PV systems. However, there are not clear trends in the battery sizes nor energy throughput. The lack of trends in the battery results is not surprising: batteries can serve many purposes including energy arbitrage, peak shaving, and grid services. In fact, in the $5\times$ bulk cost scenario so much PV energy is exported that it is necessary to use the storage systems to keep the voltage within limits.

For more information regarding model results readers are encouraged to see the code available online [21]. The examples in this section are meant is demonstrative use-cases and only represent a fraction of the types of questions that might be answered using the linearization technique for power system planning. Furthermore, the price signal from the upper level to the lower level can also be used in transactive control context by removing capacity decisions, shortening the time horizon, increasing the time resolution, and using forecasts for the uncontrolled demand and PV production.

[4]The module in [20] will be merged into [9] for ease-of-use. When debug logging is enabled the module will report which conditions did not pass.

$$\begin{aligned}
\boldsymbol{y}^\star(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} \ & \text{pwf}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}_W}\left(y_{i,n,t}\left[c_{i,n,t}+x_{i,n,t}\right]\right)+\text{pwf}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}_{\text{PV}}}\left(c_{i,t}y_{i,n,t}-x_{e,n,t}y_{e,n,t}\right) \\
& + \sum_{n\in\mathcal{N}_{\text{PV}}} c_{\text{PV}}y_{\text{PV},n} && \text{(29a)} \\
\text{s.t. } & y_{i,n,t}+y_{\text{pvprod},n,t}=d_{n,t}+y_{e,n,t}, \ (\lambda_{n,t}) && \forall n\in\mathcal{N}_{\text{PV}}, \ \forall t\in\mathcal{T} \ \text{(29b)} \\
& y_{i,n,t}=d_{n,t}+y_{\text{HVAC},n,t}/\text{COP} && \forall n\in\mathcal{N}_{W}, \ \forall t\in\mathcal{T} \ \text{(29c)} \\
& y_{\text{pvprod},n,t}\le y_{\text{PV},n}f_{\text{PV},n,t} && \forall n\in\mathcal{N}_{\text{PV}}, \ \forall t\in\mathcal{T} \ \text{(29d)} \\
& y_{\text{T},n,t}=y_{\text{T},n,t-1}+f_{hr}\left(\boldsymbol{A}y_{\text{T},n,t-1}+\boldsymbol{B}\left[y_{\text{HVAC},n,t},\boldsymbol{u}_{n,t}\right]^T\right) && \forall n\in\mathcal{N}_{\text{PV}}, \ \forall t\in\mathcal{T} \ \text{(29e)} \\
& y_{\text{T},n,0}=\text{T}_{n,t=0} && \forall n\in\mathcal{N}_{\text{PV}} \ \text{(29f)} \\
& \text{T}_{\text{hi}}\ge y_{\text{T},n,t}\ge\text{T}_{\text{lo}} && \forall n\in\mathcal{N}_{\text{PV}}, \ \forall t\in\mathcal{T} \ \text{(29g)} \\
& y_{e,n,t}\ge 0, y_{i,n,t}\ge 0, && \forall n\in\mathcal{N}_{\text{PV}}, \ \forall t\in\mathcal{T} \ \text{(29h)}
\end{aligned}$$

cost multiplier	Bulk MWh purhased	PV MWh purhased	PV MW	Battery MW	Battery MWh	Battery MWh throughput
1×	129,895	2,396	[0.04, 0.00, 0.89, 0.12, 0.97]	[0.00, 0.00, 0.00]	[0.00, 0.00, 0.00]	0
2×	127,952	4,162	[0.06, 0.00, 1.42, 0.19, 1.66]	[0.27, 0.29, 0.31]	[0.58, 0.61, 0.66]	1,026
3×	121,799	9,890	[0.14, 0.42, 2.84, 0.40, 3.50]	[0.79, 0.18, 0.00]	[1.47, 0.34, 0.00]	1,020
4×	94,092	36,981	[0.35, 2.66, 8.24, 1.53, 12.8]	[0.30, 0.35, 0.00]	[0.62, 0.74, 0.00]	752
5×	72,352	61,378	[0.56, 5.00, 13.3, 2.13, 21.2]	[0.34, 0.46, 0.28]	[0.72, 0.96, 0.58]	1,313

TABLE II: Results for the use-case example in Section IV-B. Energy purchased and throughput values are per year. The cost multiplier is applied to the bulk energy costs only. PV capacities are listed in order of nodes [9, 22, 31, 34, 17]. Battery capacities are listed in order of nodes [2, 7, 24].

C. Solution time impact

Using the example from Section (IV-B) we compare solution times with and without the upper level bilinear terms $\lambda_t y_{e,t}$ replaced with the linearization. In both cases the model is reformulated as a single level problem. Both the mixed integer-linear and the mixed integer-bilinear problems were solved using Gurobi 9.1 on 16-core 3.4GHz Linux PC with 32GB of RAM using "big M" constraints for the complementary constraints. The mixed integer-bilinear problem is (28) combined with the KKT conditions for (29). The mixed integer-linear problem is not shown due to space constraints, but is available in the public repository [21].

Both the linearized and bilinear problems have 350,405 binary variables and 2,417,777 continuous variables. The bilinear problem also has 43,800 bilinear objective terms. After 25 seconds in the presolve the linearized problem has 103,588 binary variables and 519,031 continuous variables. The linear problem solves in 128 seconds with a gap of 0.01%. After 21 seconds in the presolve the bilinear problem has 83,143 binary variables, 540,209 continuous variables, and 21,180 bilinear constraints. The bilinear model takes 8,447 seconds to get to a 57% gap, which is not improved until the operating system kills the problem at 16,032 seconds due to running out of memory. In short, the linearization method makes the otherwise intractable bilinear bilevel problem from Section (IV-B) solve in a few minutes. Similar results are expected for other large planning problems like the example in Section (IV-B).

V. Conclusion

This work presents a method for linearizing bilinear products of lower level primal and dual variables in the upper level of bilevel optimization problems, and the conditions required for the linearization to be exact. The linearization method is especially relevant for modeling large scale energy distribution systems with many stakeholders and is therefore applicable to a growing number of problems as energy markets expand and adapt to new regulations such as FERC Order 2222 [22] and the increasing adoption of distributed energy resources [23]. By publishing this method we hope that more use cases will be discovered for the linearization technique.

For future work we intend to leverage the method in an open source mathematical programming package [9], [24]. for studying compensation mechanisms of distributed energy resources serving as power system upgrade deferrals (c.f. [17]). Another future research direction involves using the optimal price signals from one level to the other as a transactive control mechanism. We will also explore more accurate and complex power flow approximations such as the second-order cone approximation for the Branch Flow Model.

Appendix A
Examples to demonstrate the Algorithms

Example 1. The simplest case for linearizing a certain $\lambda_j y_n$ term occurs when the y_n in the upper level objective bilinear term is in only one lower level constraint. In this case, step 1 of Algorithm 1 returns and (10) provides the linearization of $\lambda_j y_n$. Note that (10) is a particular instance of (9). In this example we present the *next* simplest case, which is when y_n is in more than one constraint but the other $\boldsymbol{y}$ variables in constraint j are in no other lower level constraints.

In Step 2 of Algorithm 1 the indices of (D_k) in the set $\{k \in \mathcal{N} \setminus \{n\} : V_{jk} \neq 0\}$ are added to the set of column indices to check using the recursive Algorithm 2. And the

TABLE III: Parameters, and decision variables for Problems (28) and (29). Scalar parameter values are shown in square brackets and some vector parameters include source references.

Decision Variables

$P_{0,t}, Q_{0,t}$	real, reactive power imported at feeder head, node 0 in time step t
$P_{j,t}, Q_{j,t}$	real, reactive power injected at node j in time step t
$P_{ij,t}, Q_{ij,t}$	real, reactive power flow on line ij in time step t
$w_{j,t}$	voltage magnitude squared at node j in time step t
$x_{i,j,t}$	retail price adder to refrigerated warehouse at node j in time step t
$x_{\text{B}+,j,t}$	battery charge rate at node j in time step t
$x_{\text{B}-,j,t}$	battery discharge rate at node j in time step t
$x_{\text{BkW},j}$	battery inverter power rating at node j
$x_{\text{BkWh},j}$	battery storage capacity at node j
$x_{\text{SOC},j,t}$	battery state of charge at node j in time step t
$y_{i,n,t}$	imported power at node n in time step t
$y_{e,n,t}$	exported power at node n in time step t
$y_{\text{HVAC},n,t}$	HVAC thermal power at node n in time step t
$y_{\text{T},n,t}$	interior temperature at node n in time step t
$y_{\text{HVAC},n,t}$	thermal power of HVAC system at node n in time step t
$y_{\text{PV},n}$	PV capacity at node n
$y_{\text{pvprod},n,t}$	PV production used at node n in time step t

Parameters

$c_{\text{LMP},t}$	Locational Marginal Price paid by upper level in time step t [ref. [15]]
$c_{i,n,t}$	retail price of energy [\$0.25/kWh]
c_{BkW}	cost of battery inverter [\$700/kW]
c_{BkWh}	cost of battery capacity [\$350/kWh]
c_{PV}	cost of PV capacity [\$1,400/kWh]
$f_{pf,j,t}$	power factor at node j in time step t [0.1]
r_{ij}, x_{ij}	resistance, reactance of line ij [ref. [17]]
$f_{\text{PV},n,t}$	PV production factor at node n in time step t [ref. [19]]
f_{hr}	fraction of hour in each time step (for example, $f_{hr} = 0.25$ for 15 minute time steps) [1.0]
$d_{n,t}$	uncontrolled demand at node n in time step t [ref. [16]]
COP	HVAC system coefficient of performance [4.55]
η	battery charge and discharge efficiency [0.95]
T_{hi}, T_{lo}	upper, lower temperature limit [0, -20]
$\text{T}_{n,t=0}$	initial interior temperature at node n in time step t [-1]
$\boldsymbol{A}$	HVAC system state matrix $[[\frac{-1}{RC}]$ where $R = 0.00025$ K/kW and $C = 10^5$ kJ/K]
$\boldsymbol{B}$	HVAC system input matrix $[[\frac{1}{RC} \;\; \frac{1}{C}]]$
$\boldsymbol{u}_{n,t}$	HVAC system exogenous inputs at node n in time step t (outdoor temperature) [ref. [19]]
N_t	integer number of time steps [8,760]
pwf	present worth factor assumming an annual cash flow year over year for the analysis period of 20 years and a discount rate of 5% [12.46]
$v_j^{\max}, v_j^{\min}$	maximum, minimum voltage limit at node j

set $\mathcal{J}_j$ is initialized with $\{j\}$. Algorithm 2 then checks for non-zero values of $\boldsymbol{V}$ above and below each V_{jk} entry for each of the column indices. If no non-zero values are found then Algorithm 2 returns the same sets that were passed to it, meaning that no more equation indices are needed to linearize the $\lambda_j y_n$ term.

TABLE IV: Sets and indices for Problems (28) and (29).

$\mathcal{T}$	set of integer time steps, $1, \dots, N_t$
$\mathcal{N}$	set of integer nodes in the network, $\{0, 1, \dots, N\}$
$\mathcal{N}_+$	set of positive integer nodes, $\{1, \dots, N\}$
$\mathcal{N}_{\text{PV}} \subset \mathcal{N}$	set of integer nodes that can buy PV in lower level
$\mathcal{N}_U \subset \mathcal{N}$	set of integer uncontrolled nodes
$\mathcal{N}_W \subset \mathcal{N}$	set of integer refrigerated warehouse nodes
$\mathcal{N}_{\text{B}}$	set of integer nodes that have battery decisions

Take one particular k' in $\mathcal{N} \setminus \{n\}$ for example. The special case that $V_{ik'} = 0 \; \forall i \in \mathcal{J} \setminus \{j\}$ is illustrated as follows:

$$\begin{array}{rc} & k'\text{-th col.} \\ \boldsymbol{V} = \; j\text{-th row} & \left[\begin{array}{ccc} & 0 & \\ & \vdots & \\ & 0 & \\ \dots V_{jn} \dots & V_{jk'} & \dots \\ & 0 & \\ & \vdots & \\ & 0 & \end{array}\right]. \end{array} \tag{30}$$

When $y_{k'}$ is only in constraint j then $(D_{k'})$ is:

$$\lambda_j V_{jk'} y_{k'} = c_{k'} y_{k'} + \overline{\mu}_{k'} \overline{y}_{k'} - \underline{\mu}_{k'} \underline{y}_{k'} + y_{k'} \sum_{m \in \mathcal{M}} B_{mk'} x_m. \tag{31}$$

Equation (31) can then be substituted into (P_j), shown in (11), to eliminate the bilinear term of λ_j and $y_{k'}$ in the sum over $k \in \mathcal{N} \setminus \{n\}$. A similar result follows for eliminating all of the $\lambda_j y_k$ terms on the right hand side of (11). ■

Example 2. Continuing from our previous example, now let us assume that the k'-th column of $\boldsymbol{V}$ has one other non-zero entry. Now, additional combinations of the (P_i) and (D_k) equations are necessary to eliminate the $\lambda_j y_{k'}$ term in (11). This is where step three of Algorithm 1, which relies on Algorithm 2, comes in.

For this example let $V_{i'k'} \neq 0$ for a particular i' in $\mathcal{J} \setminus \{j\}$, and let $V_{ik'} = 0 \; \forall i \in \mathcal{J} \setminus \{j, i'\}$. Also, let the i'-th row of $\boldsymbol{V}$ contain one other non-zero value $V_{i'\ell}$, and let $V_{i\ell} = 0 \; \forall i \in \mathcal{J} \setminus \{i'\}$. This case is illustrated in (32).

$$\begin{array}{rc} & \qquad \ell\text{-th col.} \; k'\text{-th col.} \\ V = \begin{array}{r} j\text{-th row} \\ i'\text{-th row} \end{array} & \left[\begin{array}{ccccc} & 0 & 0 & & \\ & \vdots & \vdots & & \\ & \vdots & 0 & & \\ \dots V_{jn} & 0 & V_{jk'} & \dots & \\ 0 \dots \; 0 & V_{i'\ell} & V_{i'k'} & 0 \dots 0 & \\ & 0 & 0 & & \\ & \vdots & \vdots & & \\ & 0 & 0 & & \end{array}\right] \end{array} \tag{32}$$

Algorithm 2 is passed row j and column k' from step 3 of Algorithm 1. Algorithm 2 first finds all the row indices of non-zero values of $\boldsymbol{V}$ in column k' (except $V_{jk'}$) and checks that those rows have not already been added to the set of rows. (Recall that redundant rows or columns found by the Algorithms indicate an underdetermined system of equations). The set of rows in Algorithm 2 now contains i' since $V_{i'k'} \neq 0$. Finally, Algorithm 2 loops over each row found to check for non-zero values of $\boldsymbol{V}$. If any values are found they are added to the columns set to search in another call to Algorithm 2 (hence the name "recursive_array_search"). In this case column ℓ is appended to the empty set of columns and so Algorithm 2 calls itself once with $r = i'$, $c = \ell$, $rows = \{i'\}$, $cols = \{\ell\}$, which finds no more non-zero entries in $\boldsymbol{V}$. Thus Algorithm 2 returns $rows = \{i'\}$, $cols = \{\ell\}$ to Algorithm 1, which appends the returned sets to $\mathcal{J}_j$ and $\mathcal{N}_n$, making the final sets $\mathcal{J}_j = \{j, i'\}$ and $\mathcal{N}_n = \{k', \ell\}$,

Now $(D_{k'})$ gives:

$$\lambda_j V_{jk'} y_{k'} + \lambda_{i'} V_{i'k'} y_{k'} = c_{k'} y_{k'} + \overline{\mu}_{k'} \overline{y}_{k'} - \underline{\mu}_{k'} \underline{y}_{k'} + y_{k'} \sum_{m \in \mathcal{M}} B_{mk'} x_m. \tag{33}$$

Since the i'-th row of $\boldsymbol{V}$ contains only one other non-zero value $V_{i'\ell}$, and the other values in column ℓ of $\boldsymbol{V}$ are zero as illustrated in (32), then adding equations $(P_{i'})$ and (D_ℓ) allows one to linearize the $\lambda_{i'} V_{i'k'} y_{k'}$ term in (33) in a similar fashion to the previous example. ■

Acknowledgment

This work was authored in part by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

The work of G. A. Hanasusanto was supported by the National Science Foundation under Grant 1752125.

References

[1] R. Raineri, "Chile: Where it all started," *Electricity market reform: an international perspective*, pp. 77–108, 2006.
[2] C. Ruiz and A. J. Conejo, "Pool strategy of a producer with endogenous formation of locational marginal prices," *IEEE Transactions on Power Systems*, vol. 24, no. 4, pp. 1855–1866, 2009.
[3] R. Fernández-Blanco, J. M. Arroyo, and N. Alguacil, "On the solution of revenue-and network-constrained day-ahead market clearing under marginal pricing—part i: An exact bilevel programming approach," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 208–219, 2016.
[4] A. Naebi, S. SeyedShenava, J. Contreras, C. Ruiz, and A. Akbarimajd, "Epec approach for finding optimal day-ahead bidding strategy equilibria of multi-microgrids in active distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 117, p. 105702, 2020.
[5] X. Xu, J. Li, Y. Xu, Z. Xu, and C. S. Lai, "A two-stage game-theoretic method for residential pv panels planning considering energy sharing mechanism," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 3562–3573, 2020.
[6] B. Cornélusse, I. Savelli, S. Paoletti, A. Giannitrapani, and A. Vicino, "A community microgrid architecture with an internal local market," *Applied Energy*, vol. 242, pp. 547–560, 2019.
[7] S. Wogrin, S. Pineda, and D. A. Tejada-Arango, "Applications of bilevel optimization in energy and electricity markets," in *Bilevel Optimization*. Springer, 2020, pp. 139–168.
[8] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.
[9] Bileveljump: A bilevel optimization extension of the jump package. [Online]. Available: https://github.com/joaquimg/BilevelJuMP.jl
[10] I. Dunning, J. Huchette, and M. Lubin, "Jump: A modeling language for mathematical optimization," *SIAM review*, vol. 59, no. 2, pp. 295–320, 2017.
[11] F. Glover, "Improved linear integer programming formulations of nonlinear integer problems," *Management Science*, vol. 22, no. 4, pp. 455–460, 1975.
[12] T. Kleinert, M. Labbé, F. a. Plein, and M. Schmidt, "There's no free lunch: on the hardness of choosing a correct big-m in bilevel optimization," *Operations research*, vol. 68, no. 6, pp. 1716–1721, 2020.
[13] J. Fortuny-Amat and B. McCarl, "A representation and economic interpretation of a two-level programming problem," *Journal of the operational Research Society*, vol. 32, no. 9, pp. 783–792, 1981.
[14] S. Pineda and J. M. Morales, "Solving linear bilevel problems using big-ms: not all that glitters is gold," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2469–2471, 2019.
[15] Ercot real-time market information. [Online]. Available: https://www.ercot.com/mktinfo/rtm
[16] Deru et al., "U.s. department of energy commercial reference building models of the national building stock," National Renewable Energy Laboratory, Tech. Rep. TP-5500-46861, 2011.
[17] P. Andrianesis, M. Caramanis, R. D. Masiello, R. D. Tabors, and S. Bahramirad, "Locational marginal value of distributed energy resources as non-wires alternatives," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 270–280, 2019.
[18] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Transactions on power Delivery*, vol. 4, no. 1, pp. 725–734, 1989.
[19] National Renewable Energy Laboratory, "PVWatts Calculator," 2020. [Online]. Available: https://developer.nrel.gov/docs/solar/pvwatts/
[20] Github repository that extends bileveljump.jl to automatically linearize the bilinear products in this paper. [Online]. Available: https://github.com/NLaws/BilevelJuMP.jl/tree/bilinear
[21] Github repository with code for the examples shown in the paper. [Online]. Available: github.com/nlaws/coder_examples
[22] (2020, September) FERC News Order 2222. [Online]. Available: https://www.ferc.gov/news-events/news/ferc-opens-wholesale-markets-distributed-resources-\landmark-action-breaks-down
[23] G. Brinkman, D. Bain, G. Buster, C. Draxl, P. Das, J. Ho, E. Ibanez, R. Jones, S. Koebrich, S. Murphy *et al.*, "The north american renewable integration study (naris): A canadian perspective," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2021.
[24] I. Dunning, J. Huchette, and M. Lubin, "Jump: A modeling language for mathematical optimization," *SIAM review*, vol. 59, no. 2, pp. 295–320, 2017.

Nicholas D. Laws Nicholas D. Laws is a PhD student in the Webber Energy Group at The University of Texas at Austin and a Research Engineer at the National Renewable Energy Laboratory. His research focuses on the integration of clean distributed energy resources in ways that benefit all stakeholders. He holds a Bachelor of Science in Aerospace Engineering from Boston University and a Master of Science from Dartmouth College, Thayer School of Engineering.

Grani A. Hanasusanto Grani A. Hanasusanto is an Assistant Professor of Operations Research and Industrial Engineering at The University of Texas at Austin (UT). Before joining UT, he was a postdoctoral researcher at the College of Management of Technology at École Polytechnique Fédérale de Lausanne. He holds a PhD degree in Operations Research from Imperial College London and an MSc degree in Financial Engineering from the National University of Singapore. He is the recipient of the 2018 NSF CAREER Award. His research focuses on the design and analysis of tractable solution schemes for decision-making problems under uncertainty, with applications in operations management, energy systems, finance, machine learning and data analytics.