



Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Convexification of Bilinear Terms over Network Polytopes

Erfan Khademnia, Danial Davarnia

To cite this article:

Erfan Khademnia, Danial Davarnia (2024) Convexification of Bilinear Terms over Network Polytopes. Mathematics of Operations Research

Published online in Articles in Advance 22 Apr 2024

. <https://doi.org/10.1287/moor.2023.0001>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2024, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Convexification of Bilinear Terms over Network Polytopes

Erfan Khademnia,^a Danial Davarnia^{a,*}
^aIndustrial and Manufacturing Systems Engineering, Iowa State University, Ames, Iowa 50011

*Corresponding author

Contact: erfank@iastate.edu (EK); davarnia@iastate.edu,  <https://orcid.org/0000-0002-8602-4454> (DD)

Received: January 3, 2023

Revised: May 26, 2023; February 18, 2024

Accepted: March 25, 2024

Published Online in Articles in Advance:
April 22, 2024

MSC2020 Subject Classifications: Primary:
90C11; Secondary: 90C35, 90C57

<https://doi.org/10.1287/moor.2023.0001>

Copyright: © 2024 INFORMS

Abstract. It is well-known that the McCormick relaxation for the bilinear constraint $z = xy$ gives the convex hull over the box domains for x and y . In network applications where the domain of bilinear variables is described by a network polytope, the McCormick relaxation, also referred to as linearization, fails to provide the convex hull and often leads to poor dual bounds. We study the convex hull of the set containing bilinear constraints $z_{i,j} = x_i y_j$ where x_i represents the arc-flow variable in a network polytope, and y_j is in a simplex. For the case where the simplex contains a single y variable, we introduce a systematic procedure to obtain the convex hull of the above set in the original space of variables, and show that all facet-defining inequalities of the convex hull can be obtained explicitly through identifying a special tree structure in the underlying network. For the generalization where the simplex contains multiple y variables, we design a constructive procedure to obtain an important class of facet-defining inequalities for the convex hull of the underlying bilinear set that is characterized by a special forest structure in the underlying network. Computational experiments conducted on different applications show the effectiveness of the proposed methods in improving the dual bounds obtained from alternative techniques.

Funding: This work was supported by Air Force Office of Scientific Research [Grant FA9550-23-1-0183]; National Science Foundation, Division of Civil, Mechanical and Manufacturing Innovation [Grant 2338641].

Supplemental Material: The online appendix is available at <https://doi.org/10.1287/moor.2023.0001>.

Keywords: network problems • bilinear terms • McCormick relaxations • disjunctive programming • cutting planes

1. Introduction

Bilinear constraints in conjunction with network models appear in various mixed-integer and nonlinear programming (MINLP) applications, including the decision diagrams (Davarnia and Van Hoes [11], Davarnia [10], Salemi and Davarnia [27], Salemi and Davarnia [28]), the fixed-charge network flow problems (Rebennack et al. [26]), network models with complementarity constraints, such as transportation problems with conflicts (Ficker et al. [18]) and the red-blue transportation problems (Vancroonenburg et al. [34]). A common occurrence of bilinear terms in network models pertains to bilevel network problems after being reformulated as a single-level program through either using a dual formulation of the inner problem or incorporating optimality conditions inside the outer problem (Ben-Ayed and Blair [4], Chiou [8]). These reformulation approaches are widely used in the network interdiction problems, where newly added bilinear terms are relaxed using a *linearization* technique based on the McCormick bounds (McCormick [24]) over a box domain; see Smith and Lim [32] for an exposition. While these relaxations provide the convex hull of the bilinear constraint over a box domain (Al-Khayyal and Falk [1]), they often lead to weak relaxations when the variables domain becomes more complicated as in general polyhedra (Gupte et al. [19], Davarnia [9]). It has been shown (Boland et al. [5], Luedtke et al. [23]) that when the number of bilinear terms in the underlying function increases, while the variables are still in a box domain, the McCormick bounds can be very poor compared with the ones obtained from the convex and concave envelopes of the function. It is shown in Bonami et al. [6] that while McCormick relaxation performs poorly for box-constrained quadratic problems, a stronger relaxation can be achieved by adding Chvátal-Gomory valid inequalities.

There are various studies in the literature that develop convexification methods for multilinear functions as a generalization of bilinear forms, but the side constraints for the involved variables are often limited to variable bounds; see Del Pia and Khajavirad [14], Del Pia and Khajavirad [15], and Khajavirad [21] for an exposition to different classes of valid inequalities for multilinear sets. In Del Pia and Di Gregorio [13] and Del Pia and Walter [16], the authors study the Chvátal rank of different classes of cutting planes, including those associated with β -cycles of the underlying hypergraph, in the multilinear polytope that arises in binary polynomial problems, while Buchheim et al. [7] provides

a characterization of the multilinear polytope in terms of the acyclicity of its corresponding hypergraph, which guarantees integrality of optimal solutions obtained by the standard linearization procedure. The authors in Gupte et al. [20] derive extended formulations for the convex hull of the graph of a bilinear function on the n -dimensional unit cube through identifying the facets of the Boolean Quadratic Polytope. In Fampa and Lee [17], an efficient method to convexify bilinear functions through McCormick relaxations is proposed which takes advantage of the structural convexity in a symmetric quadratic form. Further, Muller et al. [25] introduces a new class of valid inequalities, tailored for the spatial branch-and-bound framework, based on projection techniques that improve the quality of McCormick inequalities. Other works in the literature consider a polyhedral, often triangular, subdivision of the domain to derive strong valid inequalities for a bilinear set; see Tawarmalani et al. [33], Sherali and Alameddine [30], and Locatelli and Schoen [22] for examples of such approaches. Further, Davarnia et al. [12] proposes an aggregation procedure, referred to as *extended cancel-and-relax* (EC&R), to simultaneously convexify the graph of bilinear functions over a general polytope structure. In this paper, we make use of the EC&R procedure to derive convexification results for a bilinear set where the side constraints on variables are described by a network flow model as defined next.

For $N := \{1, \dots, n\}$, $M := \{1, \dots, m\}$, $K := \{1, \dots, \kappa\}$, and $T := \{1, \dots, \tau\}$, we consider

$$S = \{(x; y; z) \in \Xi \times \Delta_m \times \mathbb{R}^\kappa \mid y^T A^k x = z_k, \forall k \in K\},$$

where $\Xi = \{x \in \mathbb{R}^n \mid Ex \geq f, 0 \leq x \leq u\}$ is a *primal* network polytope described by the flow-balance and arc capacity constraints, and $\Delta_m = \{y \in \mathbb{R}_+^m \mid \mathbf{1}^T y \leq 1\}$ is a simplex. When variables y are binary, Δ_m represents a *special ordered set of type I* (SOS1); see Beale and Forrest [3] for an exposition. Such simplex structures appear in various applications and can be obtained by reformulating the underlying polytopes through extreme point decomposition; see Davarnia et al. [12] for a detailed account. In the above definition, $E \in \mathbb{R}^{\tau \times n}$, $f \in \mathbb{R}^\tau$, $u \in \mathbb{R}^n$, and $A^k \in \mathbb{R}^{m \times n}$ is a matrix with all elements equal to zero except one that is equal to one, that is, if $A_{ji}^k = 1$ for some $(i, j) \in N \times M$, the bilinear constraint with index k represents $y_j x_i = z_k$.

The contributions of this paper are as follows. We propose a systematic procedure to convexify S and derive explicit inequalities in its description. The resulting cutting planes are directly obtained in the original space of variables. We show that facet-defining inequalities in the convex hull description can be explicitly derived by identifying special tree and forest structures in the underlying network, leading to an interpretable and efficient cut-generating oracle. The inequalities obtained from our proposed algorithms can be added to the typical McCormick relaxations to strengthen the formulation and improve the bounds.

Since we use the specialized aggregation technique introduced in Davarnia et al. [12] in our derivation process, we next explain the novelties of this work compared with Davarnia et al. [12]. In particular Davarnia et al. [12], applies the proposed aggregation technique to a special case of S where Ξ is a *dual* network polytope described by the dual of the network flow formulation, inspired by the network interdiction applications, with the goal of obtaining explicit forms for the facet-defining inequalities that would be practical to generate. These *explicit* convexification results are obtained only for the case with $m = 1$ and $k = 1$. In this paper, we extend and generalize these results from two directions. (i) We obtain the convexification results for set S where Ξ is a *primal* network polytope, which appears in a broader application domain beyond the network interdiction problem, such as those discussed above. Due to the structural difference between the primal and dual network polytopes, the framework developed in Davarnia et al. [12] is not applicable to the network problems studied in this paper, advocating the need for a novel framework specifically designed for this new set. Consequently, the form of facet-defining inequalities for the network polytope obtained in this paper is fundamentally different from those derived in Davarnia et al. [12]. (ii) We obtain explicit convexification results for cases with general m and k . This generalization makes the problem structure, and hence the analysis, much more complicated due to losing the totally unimodularity property that exists when $m = k = 1$; see Section 3 for a detailed discussion. From a practical point of view, as shown in Section 4, the cutting planes obtained through our newly designed algorithms for the general cases with $m > 1$ and $k > 1$ can significantly improve the bounds obtained by adding cutting planes for the cases with $m = k = 1$ as well as the bounds obtained by alternative methods such as reformulation-linearization technique (RLT), highlighting the practical advantage and importance of the extended convexification results developed in this paper.

The remainder of the paper is organized as follows. We give a brief background on the EC&R procedure as a basis of our analysis in Section 2. In Section 3, we obtain convexification results for bilinear terms defined over a network polytope. Preliminary computational experiments are presented in Section 4 to show the effectiveness of the developed cut-generating frameworks. We give concluding remarks in Section 5.

1.2. Notation

Bold letters represent vectors. We refer to the j th row (resp. j th column) of a matrix $A \in \mathbb{R}^{m \times n}$ by A_j (resp. by $A_{\cdot j}$). For a given set $S \subseteq \mathbb{R}^n$, we denote by $\text{conv}(S)$ its convex hull. We use symbol $\pm$ to show both cases with $+$ and $-$. For example, when we use $l^\pm$ in an expression, it means that expression holds for both cases l^+ and l^- .

2. Extended Cancel-and-Relax

In this section, we present the EC&R procedure adapted for S . The following theorem is at the core of this procedure, as shown in theorem 2.7 of Davarnia et al. [12].

Theorem 1. *A convex hull description of S can be obtained by the linear constraints in Ξ and Δ_m together with all class- $l^\pm$ EC&R inequalities for all $l \in K$.*

A class- $l^\pm$ EC&R inequality is obtained through a weighted aggregation of the constraints in the description of S . The procedure to generate this inequality is as follows.

1. We select $l \in K$ to be the index of a bilinear constraint used in the aggregation, which we refer to as the *base equality*. We also select a sign indicator $+$ or $-$ to indicate whether a weight 1 or -1 is used for the base equality during aggregation.

2. Defining T as the index set of the nonbound constraints in Ξ , we select $\mathcal{I}_1, \dots, \mathcal{I}_m$ and $\tilde{\mathcal{I}}$ as subsets of T whose intersection is empty. Then, for each $j \in M$ and for each $t \in \mathcal{I}_j$ (resp. $t \in \tilde{\mathcal{I}}$), we multiply the constraint $E_t x \geq f_t$ by $\gamma_t^j y_j$ where $\gamma_t^j \geq 0$ (resp. by $\theta_t(1 - \sum_{i \in M} y_i)$ where $\theta_t \geq 0$).

3. We select $\mathcal{J}$ and $\tilde{\mathcal{J}}$ as disjoint subsets of N . Then, for each index $i \in \mathcal{J}$, we multiply $x_i \geq 0$ by $\lambda_i(1 - \sum_{j \in M} y_j)$ where $\lambda_i \geq 0$, and for each $i \in \tilde{\mathcal{J}}$, we multiply $u_i - x_i \geq 0$ by $\mu_i(1 - \sum_{j \in M} y_j)$ where $\mu_i \geq 0$.

The above sets are compactly represented as $[\mathcal{I}_1, \dots, \mathcal{I}_m, \tilde{\mathcal{I}} | \mathcal{J}, \tilde{\mathcal{J}}]$, which is called an *EC&R assignment*. Each EC&R assignment is identified by its class- $l^\pm$ where l is the index of the base equality and $\pm$ is its sign indicator. We next aggregate all aforementioned weighted constraints. During the aggregation, we require that weights γ, θ, λ and μ be chosen in such a way that:

(C1) at least $\sum_{j \in M} |\mathcal{I}_j| + |\tilde{\mathcal{I}}| + |\mathcal{J}| + |\tilde{\mathcal{J}}|$ bilinear terms are canceled (i.e., their coefficient becomes zero), and

(C2) if $\cup_{j \in M} \mathcal{I}_j \cup \tilde{\mathcal{I}} \cup \mathcal{J} \cup \tilde{\mathcal{J}} \neq \emptyset$, for each constraint used in the aggregation (including the base equality), at least one bilinear term among all those created after multiplying that constraint with its corresponding weight is canceled.

The desired EC&R inequality is then obtained by relaxing (i.e., replacing) the remaining bilinear terms of the form $x_i y_j$ in the aggregated inequality using either $u_i y_j - x_i y_j \geq 0$ or $-x_i y_j + z_k \geq 0$ for some $k \in K$ such that $A_{ji}^k = 1$. Similarly, the bilinear terms of the form $-x_i y_j$ is relaxed using either $x_i y_j \geq 0$ or $x_i y_j - z_k \geq 0$ for some $k \in K$ such that $A_{ji}^k = 1$. The resulting linear inequality is referred to as a class- $l^\pm$ EC&R inequality.

A summary of the derivation steps for the EC&R procedure is given in the electronic companion; see Davarnia et al. [12] for a detailed exposition. Here, we only present an end result that will be used in the subsequent sections.

For each $l \in K$, define $K_l = K \setminus \{l\}$, and let $\pi^l = (\beta^+; \beta^-; \{\gamma^j\}_{j \in M}; \theta; \{\eta^j\}_{j \in M}; \{\rho^j\}_{j \in M}; \lambda; \mu)$ be the vector of aggregation weights used in the EC&R procedure. In particular, the index l is determined in Step 1 of the EC&R procedure by picking a base equality l with either $+1$ or -1 weights. The components γ^j (resp. θ) capture the weights for y_j (resp. $1 - \sum_{j \in M} y_j$) when multiplied with the nonbound constraints in Ξ as demonstrated in Step 2 of the EC&R procedure. Similarly, λ and μ record the weights for $1 - \sum_{j \in M} y_j$ when multiplied with the bound constraints in Ξ in Step 3 of the EC&R procedure. Finally, the relaxation step in the EC&R procedure determines the components $\beta^\pm$ as the weights of the bilinear constraints in K_l , as well as the components η^j and ρ^j as the weights for y_j when multiplied with the bound constraints in Ξ to cancel the remaining bilinear terms. Then, it is shown in theorem 2.7 of Davarnia et al. [12] that criteria (C1) and (C2) of the EC&R procedure provide necessary conditions for the weight vector π^l to be an extreme point of

$$\mathcal{C}^l = \left\{ \pi^l \in \mathbb{R}_+^{2(\kappa-1)+(m+1)(\tau+2n)} \left| \sum_{k \in K_l} A_{ji}^k (\beta_k^+ - \beta_k^-) + \sum_{t \in T} E_{it} (\gamma_t^j - \theta_t) \right. \right. \\ \left. \left. + \eta_i^j - \rho_i^j - \lambda_i + \mu_i = \pm A_{ji}^l, \forall (i, j) \in N \times M \right\}. \quad (1)$$

This set plays a central role in our analysis because the collection of the EC&R inequalities corresponding to the extreme points of $\mathcal{C}^l$, for all $l \in K$, contains all *nontrivial* facet-defining inequalities in $\text{conv}(S)$, where a nontrivial inequality is one that cannot be implied by the linear constraints in the description of Ξ and Δ_m .

3. Network Polytopes

In this section, we study the set $\mathcal{S}$ where Ξ represents a network polytope. In particular, the constraint set $Ex \geq f$ is composed of the flow-balance constraints after being separated into two inequalities of opposite signs, that is, E is an augmented node-arc incidence matrix where each row is duplicated with a negative sign, and f represents the extended supply/demand vector. In this description, u denotes the arc-capacity vector.

3.1. The Case with $m = 1$

In this section, we consider the case where $m = 1$, whose corresponding bilinear set is denoted by $\mathcal{S}^1$. This special case is important as it provides a natural setting to improve the McCormick results as the most common relaxation for bilinear programs, which considers $m = 1$, even when there are multiple y variables that interact with each other through side constraints. In fact, the McCormick relaxation of bilinear terms with $m = 1$ is a building block of the factorable decomposition technique at the core of MINLP solvers (McCormick et al. [24]). Therefore, the convexification results for the case where each y variable is treated separately (i.e., $m = 1$) is critical in determining the quality of the relaxation bound obtained for the entire model. For this reason, improving the McCormick relaxation, even for the simplest case with $m = 1$, could lead to substantial bound improvement as corroborated by the computational results in the Section 4.

In this special case, we can simplify notation by matching the indices of z and x variables such that $y_1 x_k = z_k$ for all $k \in K = N$. We next show that, to generate an EC&R inequality for $\mathcal{S}^1$, it is sufficient to use aggregation weight 1 for all constraints used in the aggregation.

Proposition 1. Let $\bar{\pi}^l$ be an extreme point of the projection cone $\mathcal{C}^l$, for some $l \in K$, corresponding to a nontrivial facet-defining inequality of $\text{conv}(\mathcal{S}^1)$. Then, it can be scaled in such a way that all of its components are 0 or 1.

Proof. When $m = 1$, we can write $\mathcal{C}^l$ in (1) as

$$\mathcal{C}^l = \left\{ \pi^l \in \mathbb{R}_+^{2(\kappa-1)+2(\tau+2n)} \left| \sum_{t \in T} E_{ti}(\gamma_t^1 - \theta_t) + \mu_i - \lambda_i + \eta_i^1 - \rho_i^1 + \sum_{k \in K_l} A_{1i}^k(\beta_k^+ - \beta_k^-) = \pm A_{1i}^l, \forall i \in N \right. \right\}.$$

We can rearrange the columns of the coefficient matrix of the system defining $\mathcal{C}^l$ to obtain

$$[E^T | -E^T | I | -I | I | -I | \bar{I} | -\bar{I}]. \quad (2)$$

In the above matrix, the rows correspond to bilinear terms $y_1 x_i$ for $i \in N$. The first and second column blocks correspond to the weights of the nonbound constraints in Ξ multiplied by y_1 and $1 - y_1$, which are denoted by γ_t^1 and θ_t , respectively, for all $t \in T$. The third and fourth column blocks correspond to the weights of the lower and upper bound constraints on variables in Ξ multiplied by $1 - y_1$, which are captured by μ_i and λ_i , respectively, for all $i \in N$. Similarly, the fifth and sixth column blocks correspond to the weights of the lower and upper bound constraints on variables in Ξ multiplied by y_1 , which are recorded by η_i^1 and ρ_i^1 , respectively, for all $i \in N$. In these columns, I represents the identity matrix of appropriate size. Lastly, the seventh and eighth column blocks correspond to the weights of the bilinear constraints in $\mathcal{S}^1$, which are represented by β_k^+ and β_k^- , respectively, for all $k \in K_l$. In particular, the element at column $k \in K_l$ and row $i \in N$ of $\bar{I}$ is equal to 1 if $i = k$, and it is equal to zero otherwise. Based on these column values, it can be easily verified that (2) is totally unimodular (TU). In $\mathcal{C}^l$, the right-hand-side vector is $\pm e^l \in \mathbb{R}^{m+n}$, where e^l is the unit vector whose components are all zero except for that corresponding to row l representing $y_1 x_l$, which is equal to 1. Because $\bar{\pi}^l$ is an extreme point of $\mathcal{C}^l$, it is associated with a basic feasible solution for its system of equations. Let B be the corresponding basis for (2). It follows from Cramer's rule that all elements of B^{-1} belong to $\{0, -1, 1\}$ since (2) is TU. Therefore, the components of $\pm B^{-1}e^l$ belong to $\{0, -1, 1\}$. We conclude that the components of basic feasible solutions to $\mathcal{C}^l$ are equal to 0 or 1 due to nonnegativity of all variables in its description. $\square$

Remark 1. When $m = 1$, multiplying the bound constraints with $1 - y_1$ in Step 3 of the EC&R procedure produces two of the standard McCormick bounds. As a result, we can skip Step 3 in the EC&R procedure and merge it into the relaxation step, in which the other two McCormick bounds and the bilinear constraints are used for relaxing the remaining bilinear terms. In specific, any remaining bilinear term in the aggregated inequality can be relaxed into either of the two McCormick lower bounds or the two McCormick upper bounds or the $\pm z$ variable corresponding to that term depending on its sign. In this case, the characterization of EC&R assignment can be reduced further to $[\mathcal{I}_1, \bar{\mathcal{I}}]$.

Remark 2. As described in Remark 1, each remaining bilinear term in the aggregated inequality of the EC&R procedure can be relaxed into three different linear terms. While this can lead to an exponential growth in the number of resulting linear EC&R inequalities for each EC&R assignment, we can use an efficient separation procedure to find the most violated inequality among the resulting EC&R inequalities as follows. Assume that we aim to separate a given solution $(\bar{x}; \bar{y}_1; \bar{z})$ from $\text{conv}(S^1)$ through the EC&R inequalities obtained from the aggregated inequality $g(x; y_1; z) \geq 0$ associated with the EC&R assignment $[\mathcal{I}_1, \tilde{\mathcal{I}}]$. For each bilinear term $y_1 x_i$, we choose the relaxation option that provides the minimum value among $u_i \bar{y}_1$ obtained from using $y_1(u_i - x_i) \geq 0$, $\bar{x}_i$ obtained from using $(1 - y_1)x_i \geq 0$, and $\bar{z}_i$ obtained from using $-y_1 x_i + z_i \geq 0$. Similarly, for each bilinear term $-y_1 x_i$, we choose the relaxation option that provides the minimum value among 0 obtained from using $y_1 x_i \geq 0$, $u_i - \bar{x}_i - u_i \bar{y}_1$ obtained from using $(1 - y_1)(u_i - x_i) \geq 0$, and $-\bar{z}_i$ obtained from using $y_1 x_i - z_i \geq 0$. This approach provides the most violated EC&R inequality in the time linear in the number of remaining bilinear terms in the aggregated inequality.

Considering the relation between the extreme points of the projection cone $\mathcal{C}^l$ for $l \in K$ and the aggregation weights in the EC&R procedure, Proposition 1 and Remark 1 imply that generating class- $l^\pm$ EC&R inequalities reduces to identifying the assignment $[\mathcal{I}_1, \tilde{\mathcal{I}}]$ as the aggregation weights are readily determined. In particular, the constraints in $\mathcal{I}_1$ are multiplied with y_1 , and those in $\tilde{\mathcal{I}}$ are multiplied with $1 - y_1$. We next show that, for set S^1 , identifying all the EC&R assignments that satisfy the EC&R conditions (C1) and (C2) can be achieved by considering a special graphical structure in the underlying network.

Given a network $G = (V, A)$ with a node set V and arc set A , assume that the index k of variables z_k in the description of S^1 refers to the arc whose flow variable x_k appears in that bilinear constraint, that is, $y_1 x_k = z_k$ for $k \in A = N = K$. We define $t(k)$ and $h(k)$ to be the tail and head nodes of arc $k \in A$, respectively. Further, for any node $i \in V$, we define $\delta^+(i)$ and $\delta^-(i)$ to be the set of outgoing and incoming arcs at that node, respectively. We refer to the flow-balance inequality $\sum_{k \in \delta^+(i)} x_k - \sum_{k \in \delta^-(i)} x_k \geq f_i$ (resp. $-\sum_{k \in \delta^+(i)} x_k + \sum_{k \in \delta^-(i)} x_k \geq -f_i$) corresponding to node i as the *positive* (resp. *negative*) flow-balance inequality, and refer to its index in the description of Ξ by i^+ (resp. i^-) to be recorded in the EC&R assignment. For example, an EC&R assignment $\{i^+, \{j^-\}\}$ implies that, in the aggregation, the positive flow-balance inequality corresponding to the node $i \in V$ is multiplied with y_1 , and the negative flow-balance inequality corresponding to the node $j \in V$ is multiplied with $1 - y_1$. In the sequel, we denote the undirected variant of a subnetwork P of G by $\bar{P}$, and conversely, we denote the directed variant of an undirected subnetwork $\bar{P}$ of $\bar{G}$ by P according to the arc directions in G .

Proposition 2. Consider set S^1 with Ξ that represents the network polytope corresponding to the network $G = (V, A)$. Let $[\mathcal{I}_1, \tilde{\mathcal{I}}]$ be an EC&R assignment for class- $l^\pm$, for some $l \in A$, that leads to a nontrivial facet-defining inequality of $\text{conv}(S^1)$. Define $\tilde{\mathcal{I}} = \{i \in V \mid i^\pm \in \mathcal{I}_1 \cup \tilde{\mathcal{I}}\}$ to be the subset of nodes whose flow-balance inequalities are used in the aggregation. Then, there exists a tree $\bar{T}$ of $\bar{G}$ composed of the nodes in $\tilde{\mathcal{I}}$ such that arc l is incident to exactly one node of $\bar{T}$.

Proof. First, we observe that for each node $i \in V$, both of its positive and negative flow-balance inequalities cannot be selected for the aggregation, since otherwise, the columns representing the positive and negative inequalities in the basis of the coefficient matrix (2) associated with the extreme point of $\mathcal{C}^l$ would be linearly dependent, which would be a contradiction to the fact that the selected EC&R assignment leads to a facet-defining inequality of $\text{conv}(S^1)$; see the proof of Proposition 1 for details. As a result, considering that $\mathcal{I}_1 \cap \tilde{\mathcal{I}} = \emptyset$ by the EC&R requirement, at most one of the following possibilities can occur in the EC&R assignment: $i^+ \in \mathcal{I}_1$, $i^+ \in \tilde{\mathcal{I}}$, $i^- \in \mathcal{I}_1$, and $i^- \in \tilde{\mathcal{I}}$. Therefore, each node in $\tilde{\mathcal{I}}$ corresponds to a unique flow-balance constraint in the EC&R assignment. Next, we show that arc l is incident to exactly one node of $\tilde{\mathcal{I}}$. It follows from condition (C2) of the EC&R procedure that the bilinear term $y_1 x_l$ for arc l in the base equality must be canceled during the aggregation. The constraints of Ξ that can produce the bilinear term $y_1 x_l$ during the aggregation are the flow-balance constraint corresponding to the tail node $t(l)$ of arc l , and the flow-balance constraint corresponding to the head node $h(l)$ of arc l . Since the aggregation weight for all the constraints in the EC&R assignment are 1 according to Proposition 1, and considering that each flow-balance constraint can appear once in the aggregation as noted above, the only possibility to cancel the term $y_1 x_l$ is to pick exactly one of the above constraints in the EC&R assignment. As a result, exactly one of the head and the tail nodes of arc l must be in $\tilde{\mathcal{I}}$. Next, we show that there exists a tree $\bar{T}$ of $\bar{G}$ whose node set is $\tilde{\mathcal{I}}$. Assume by contradiction that there is no such tree composed of the nodes in $\tilde{\mathcal{I}}$. Therefore, $\tilde{\mathcal{I}}$ can be partitioned into two subsets $\tilde{\mathcal{I}}_1$ and $\tilde{\mathcal{I}}_2$, where the nodes in $\tilde{\mathcal{I}}_1$ are not adjacent to any nodes in $\tilde{\mathcal{I}}_2$. It is clear that arc l cannot be incident to the nodes in both $\tilde{\mathcal{I}}_1$ and $\tilde{\mathcal{I}}_2$, since otherwise $\tilde{\mathcal{I}}_1$ and $\tilde{\mathcal{I}}_2$ would have adjacent nodes. Assume without the loss of generality that arc l is incident to a node in $\tilde{\mathcal{I}}_1$. Since the given EC&R assignment leads to a facet-defining inequality after applying the relaxation step, its aggregation weights correspond to

an extreme point of C^l as described in the proof of Proposition 1. The resulting system of equations for the associated basic feasible solution can be written as

$$\begin{bmatrix} \pm \bar{E}_1 & \pm I_1 & \pm \bar{I}_1 & 0 & 0 & 0 & C_1 \\ 0 & 0 & 0 & \pm \bar{E}_2 & \pm I_2 & \pm \bar{I}_2 & C_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & C_3 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \pm e^l \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (3)$$

where the columns and rows of the basis matrix have been suitably reordered. In (3), the first (resp. second) row block corresponds to bilinear terms $y_1 x_i$ for arcs $i \in A$ that are incident to the nodes in $\tilde{T}_1$ (resp. $\tilde{T}_2$), and the last row block corresponds to all the other bilinear terms that do not appear during aggregation. The first (resp. fourth) column block denotes the transpose of the node-arc incidence matrix for nodes in $\tilde{T}_1$ (resp. $\tilde{T}_2$). The second (resp. fifth) column block contains positive or negative multiples of columns of the identity matrix representing the weights used in the relaxation step of the EC&R procedure corresponding to the McCormick bounds. The third (resp. sixth) column block represents positive or negative multiples of the bilinear constraints in the description of S^1 used in the relaxation step corresponding to the arcs appearing in the first (resp. second) row blocks. All these columns have weights equal to 1 according to Proposition 1 as denoted in the first two row blocks of the solution vector multiplied with this matrix. The last column in the basis corresponds to the constraints that have weights 0 in the basic feasible solution and are added to complete the basis. Lastly, e^l is a unit vector whose elements are all zeros except that corresponding to $y_1 x_l$, which is equal to 1. It follows that the linear combination of the columns in the column blocks 4, 5, and 6 of the basis matrix with weights 1 yields the zero vector. This shows that these columns are linearly dependent, a contradiction. $\square$

Proposition 2 implies that each nontrivial EC&R inequality can be obtained as an aggregation of constraints corresponding to a special tree structures. The next theorem provides the converse result that aggregating constraints associated with each special tree structure can produce EC&R inequalities.

Theorem 2. Consider set S^1 with Ξ that represents the network polytope corresponding to the network $G = (V, A)$. Let $\bar{T}$ be a tree in $\bar{G}$ with the node set $\tilde{T} \subseteq V$, and let $l \in A$ be an arc that is incident to exactly one node of $\bar{T}$. Then, for any partition $\tilde{T}_1$ and $\tilde{T}_2$ of $\tilde{T}$ (i.e., $\tilde{T}_1 \cap \tilde{T}_2 = \emptyset$ and $\tilde{T}_1 \cup \tilde{T}_2 = \tilde{T}$), we have that

- (i) if $h(l) \in \tilde{T}$, then $[\{i^+\}_{i \in \tilde{T}_1}, \{i^-\}_{i \in \tilde{T}_2}]$ is an EC&R assignment for class- l^+ ,
- (ii) if $h(l) \in \tilde{T}$, then $[\{i^-\}_{i \in \tilde{T}_1}, \{i^+\}_{i \in \tilde{T}_2}]$ is an EC&R assignment for class- l^- ,
- (iii) if $t(l) \in \tilde{T}$, then $[\{i^-\}_{i \in \tilde{T}_1}, \{i^+\}_{i \in \tilde{T}_2}]$ is an EC&R assignment for class- l^+ ,
- (iv) if $t(l) \in \tilde{T}$, then $[\{i^+\}_{i \in \tilde{T}_1}, \{i^-\}_{i \in \tilde{T}_2}]$ is an EC&R assignment for class- l^- .

Proof. We show the result for part (i), as the proof for parts (ii)–(iv) follows from similar arguments. It suffices to show that the aggregation procedure performed on the constraints in the proposed assignment satisfies the EC&R conditions (C1) and (C2). For condition (C1), we need to show that at least $|\tilde{T}_1| + |\tilde{T}_2|$ bilinear terms are canceled during aggregation. Let R be the set of arcs in T , which is the directed variant of $\bar{T}$ obtained by replacing each edge with its corresponding arc in G . It is clear from the definition that $l \notin R$. As a result, for each $r \in R$, the only constraints in the aggregation that contain x_r are the flow-balance constraints corresponding to the head node $h(r)$ and tail node $t(r)$ of r since both of these nodes are included in $\tilde{T}$ as r is an arc in T . There are four cases for the partitions of $\tilde{T}$ that these head and tail nodes can belong to. For the first case, assume that $h(r) \in \tilde{T}_1$ and $t(r) \in \tilde{T}_1$. It follows from the EC&R assignment in case (i) that the positive flow-balance constraints $h(r)^+$ and $t(r)^+$ are used in the aggregation with weights y_1 . In particular, we have $y_1(\sum_{k \in \delta^+(h(r))} x_k - \sum_{k \in \delta^-(h(r)) \setminus \{r\}} x_k - x_r \geq f_{h(r)})$ added with $y_1(\sum_{k \in \delta^+(t(r)) \setminus \{r\}} x_k - \sum_{k \in \delta^-(t(r))} x_k + x_r \geq f_{t(r)})$, which results in the cancellation of $y_1 x_r$. For the second case, assume that $h(r) \in \tilde{T}_1$ and $t(r) \in \tilde{T}_2$. It follows from the EC&R assignment that the positive flow-balance constraints $h(r)^+$ and the negative flow-balance constraint $t(r)^+$ are used in the aggregation with weights y_1 and $(1 - y_1)$, respectively. In particular, we have $y_1(\sum_{k \in \delta^+(h(r))} x_k - \sum_{k \in \delta^-(h(r)) \setminus \{r\}} x_k - x_r \geq f_{h(r)})$ added with $(1 - y_1)(-\sum_{k \in \delta^+(t(r)) \setminus \{r\}} x_k + \sum_{k \in \delta^-(t(r))} x_k - x_r \geq -f_{t(r)})$, which results in the cancellation of $y_1 x_r$. For the remaining two cases, we can use similar arguments by changing the inequality signs to show that the term $y_1 x_r$ will be canceled during aggregation. As a result, we obtain at least $|R|$ cancellations during the aggregation corresponding to the arcs of T . Since $\bar{T}$ is a tree, we have that $|R| = |\tilde{T}_1| + |\tilde{T}_2| - 1$. Finally, for arc l , it follows from the assumption of case (i) in the problem statement that $h(l) \in \tilde{T}$ and $t(l) \notin \tilde{T}$. If $h(l) \in \tilde{T}_1$, then according to the EC&R procedure for class- l^+ , we aggregate $y_1 x_l - z_l = 0$ with $y_1(\sum_{k \in \delta^+(h(l))} x_k - \sum_{k \in \delta^-(h(l)) \setminus \{l\}} x_k - x_l \geq f_{h(l)})$, which results in the cancellation of $y_1 x_l$. If $h(l) \in \tilde{T}_2$, we aggregate $y_1 x_l - z_l = 0$ with $(1 - y_1)(-\sum_{k \in \delta^+(h(l))} x_k + \sum_{k \in \delta^-(h(l)) \setminus \{l\}} x_k + x_l \geq -f_{h(l)})$, which

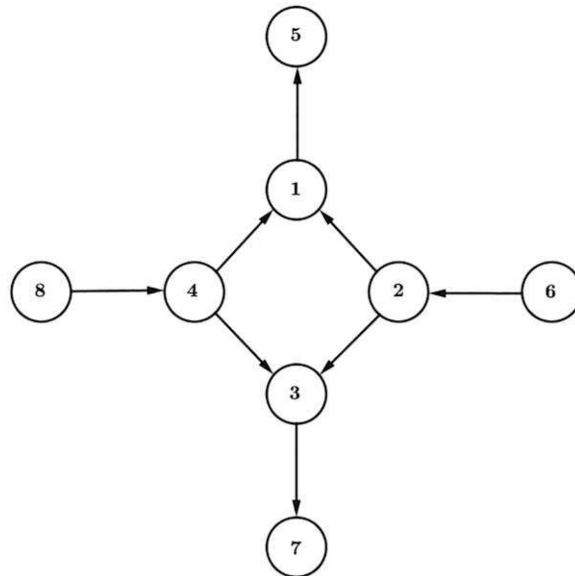
results in the cancellation of $y_1 x_l$. As a result, in total we have at least $|R| + 1 = |\tilde{T}_1| + |\tilde{T}_2|$ cancellations during the aggregation of the constraints in the EC&R assignment, showing the satisfaction of condition (C1) of the EC&R procedure. For condition (C2) of the EC&R procedure, we need to show that for each constraint used in the aggregation, including the base equality, at least one bilinear term among those created after multiplication of that constraint with its corresponding weight is canceled. There are two types of constraints to consider. The first type is the flow-balance constraints in $\tilde{T}_1$ and $\tilde{T}_2$, which correspond to the nodes of $\bar{T}$. It follows from the previous discussion that for each node $i \in \tilde{T}_1 \cup \tilde{T}_2$, the bilinear term $y_1 x_r$, where r is an arc in T that is incident to i , that is, $h(r) = i$ or $t(r) = i$, is canceled during aggregation. This proves that at least one bilinear term is canceled in the inequality obtained after multiplying the corresponding flow-balance constraint at node i with y_1 or $1 - y_1$. The second type of constraints used in the aggregation is the base equality l . The proof follows from an argument similar to that given above where we showed that the bilinear term $y_1 x_l$ that appears in the base constraint $y_1 x_l - z_l = 0$ is canceled. We conclude that condition (C2) of the EC&R procedure is satisfied for all constraints used in the aggregation. $\square$

In view of Theorem 2, note that for the simplest choice of the tree $\bar{T}$, that is, an empty set, the resulting EC&R inequalities recover the classical McCormick bounds. Therefore, considering any nonempty tree structure can potentially improve the McCormick results by adding new valid inequalities for the bilinear set.

Proposition 2 and Theorem 2 suggest that the EC&R assignments have a simple graphical interpretation for S^1 , which can be used to generate all nontrivial EC&R inequalities to describe $\text{conv}(S^1)$ without the need to search for all possible constraints and their aggregation weights that satisfy the EC&R conditions as is common for general bilinear sets. This attribute can significantly mitigate cut-generation efforts when used systematically to produce cutting plane. Such a systematic procedure can be designed by identifying a tree of a given network and then following the result of Theorem 2 to obtain the corresponding EC&R assignments. We illustrate this procedure in the following example.

Example 1. Consider set S^1 where Ξ represents the network model corresponding to a *spiked cycle* graph $G = (V, A)$ shown in Figure 1. We refer to each arc in this network as a pair (i, j) of its tail node i and its head node j , and denote its corresponding flow variable as $x_{i,j}$. Assume that we are interested in finding EC&R assignments for class- $(1, 5)^+$. According to Theorem 2, we need to identify the trees that contain exactly one of the tail and head nodes of arc $(1, 5)$. For instance, we may select the tree $\bar{T}$ composed of the nodes $\tilde{T} = \{8, 4, 1, 2, 6\}$ that contain the tail node of arc $(1, 5)$. Consider the partitions $\tilde{T}_1 = \{8, 2\}$ and $\tilde{T}_2 = \{4, 1, 6\}$. Following case (iii) in Theorem 2, we can obtain the EC&R assignment $[\{8^-, 2^-\}, \{4^+, 1^+, 6^+\}]$ for class- $(1, 5)^+$. As a result, we multiply the negative flow-balance constraints at nodes 8 and 2 with y_1 , and we multiply the positive flow-balance constraints at nodes 4, 1, and 6 with $1 - y_1$, and aggregate them with the base bilinear equality corresponding to arc $(1, 5)$ with

Figure 1. Graph G of Example 1.



weight 1 to obtain the aggregated inequality

$$\begin{aligned} & -z_{1,5} - y_1 x_{2,3} - y_1 x_{4,3} + (f_8 + f_2 + f_1 + f_4 + f_6) y_1 \\ & + x_{1,5} - x_{2,1} + x_{4,3} - x_{8,4} + x_{6,2} - f_1 - f_4 - f_6 \geq 0 \end{aligned}$$

where f_i denotes the supply/demand value at node i . Following Remark 1, we may relax each remaining bilinear term $-y_1 x_{2,3}$ and $-y_1 x_{4,3}$ into three possible linear expressions, leading to 9 total EC&R inequalities. If implemented inside of a separation oracle, we can use Remark 2 to find the most violated inequality among these 9 efficiently. $\square$

3.1.1. Generalization of the Coefficient Matrix for Bilinear Constraints. Consider a generalization of S^1 where the bilinear constraints may contain multiple bilinear terms:

$$\tilde{S}^1 = \{(x; y; z) \in \Xi \times \Delta_1 \times \mathbb{R}^K \mid y_1 \tilde{A}^k x = z_k, \quad \forall k \in K\},$$

where $\tilde{A}^k$ is a matrix of appropriate size with potentially multiple nonzero elements. For instance, $\tilde{S}^1$ may include the constraint $2y_1 x_i - 5y_1 x_j = z_k$ for some $i, j \in N$. In this case, the coefficient matrix (2) of $\mathcal{C}^l$ will be modified as follows after rearranging columns and rows.

$$[E^\top \mid -E^\top |I| - I|I| - I|\tilde{A}| - \tilde{A}]. \quad (4)$$

In the above matrix, the row and column blocks are defined similarly to those of (2) with a difference that the seventh and eighth column blocks correspond to the weights of the bilinear constraints $y_1 \tilde{A}^k x = z_k$, which are represented by β_k^+ and β_k^- in the dual weight vector, respectively, for all $k \in K_l$. In particular, the element at column $k \in K_l$ and row $i \in N$ of $\tilde{A}$ is equal to $\tilde{A}_{1i}^k$. It is clear from the structure of (4) that this matrix may lose the TU property when $\tilde{A}^k$ contains multiple nonzero elements for some $k \in K_l$. In fact, this property may not hold even if $\tilde{A}_{1i}^k \in \{0, 1, -1\}$ for all $k \in K_l$ and $i \in N$. As a result, there is no guarantee that the aggregation weights for all the EC&R inequalities corresponding to nontrivial facets of $\text{conv}(\tilde{S}^1)$ will be 1. While an explicit derivation of the convex hull description through identifying special network structures, such as those presented for S^1 , may not be attainable for this problem in its original space of variables, we can use the following ancillary result to apply Theorem 2 and obtain a convex hull description for $\tilde{S}^1$ in a higher-dimensional space.

Proposition 3. Consider sets

$$S^1 = \{(x; y; w) \in \Xi \times \Delta_1 \times \mathbb{R}^n \mid y_1 x_i = w_i, \quad \forall i \in N\},$$

and

$$\mathcal{D} = \{(x; y; w; z) \in \Xi \times \Delta_1 \times \mathbb{R}^n \times \mathbb{R}^K \mid \tilde{A}^k w = z_k, \quad \forall k \in K\},$$

Then,

$$\text{conv}((S^1 \times \mathbb{R}^K) \cap \mathcal{D}) = (\text{conv}(S^1) \times \mathbb{R}^K) \cap \mathcal{D}. \quad (5)$$

Proof. We prove the result by showing both directions of inclusion for the equality. The direct inclusion follows from the fact that the convex hull of intersection of two sets is a subset of the intersection of the convex hulls of those sets. For the reverse inclusion, we need to show that $\text{conv}((S^1 \times \mathbb{R}^K) \cap \mathcal{D}) \supseteq (\text{conv}(S^1) \times \mathbb{R}^K) \cap \mathcal{D}$. Consider a point $\bar{\phi} = (\bar{x}; \bar{y}; \bar{w}; \bar{z}) \in (\text{conv}(S^1) \times \mathbb{R}^K) \cap \mathcal{D}$. We show that $\bar{\phi} \in \text{conv}((S^1 \times \mathbb{R}^K) \cap \mathcal{D})$. It follows from the assumption that $\bar{z}_k = \tilde{A}^k \bar{w}$ for all $k \in K$. Further, there must exist a finite collection of points $\hat{\phi}^j = (\hat{x}^j; \hat{y}^j; \hat{w}^j) \in S^1$ for $j \in J$ such that $\bar{x} = \sum_{j \in J} \omega_j \hat{x}^j$, $\bar{y} = \sum_{j \in J} \omega_j \hat{y}^j$, and $\bar{w} = \sum_{j \in J} \omega_j \hat{w}^j$ for some nonnegative weights ω_j such that $\sum_{j \in J} \omega_j = 1$. Consider the set of points $\hat{\phi}^j = (\hat{x}^j; \hat{y}^j; \hat{w}^j; \hat{z}^j)$ for $j \in J$ such that $\hat{x}^j = \hat{x}^j$, $\hat{y}^j = \hat{y}^j$, $\hat{w}^j = \hat{w}^j$, and $\hat{z}_k^j = \tilde{A}^k \hat{w}^j$ for all $k \in K$. It is clear that $\hat{\phi}^j \in (S^1 \times \mathbb{R}^K) \cap \mathcal{D}$ for all $j \in J$ by definition of the components of these points. It follows that $\bar{\phi} = \sum_{j \in J} \omega_j \hat{\phi}^j$, since $\bar{x} = \sum_{j \in J} \omega_j \hat{x}^j$, $\bar{y} = \sum_{j \in J} \omega_j \hat{y}^j$, and $\bar{w} = \sum_{j \in J} \omega_j \hat{w}^j$ by definition, and since $\bar{z}_k = \tilde{A}^k \bar{w} = \tilde{A}^k (\sum_{j \in J} \omega_j \hat{w}^j) = \sum_{j \in J} \omega_j \tilde{A}^k \hat{w}^j = \sum_{j \in J} \omega_j \hat{z}_k^j$ for all $k \in K$. This proves that $\bar{\phi} \in \text{conv}((S^1 \times \mathbb{R}^K) \cap \mathcal{D})$. $\square$

The result of Proposition 3 shows that we can obtain a convex hull description for $\tilde{S}^1$ in a higher dimension, which is expressed on the left-hand-side of (5), by finding the convex hull of S^1 through application of Theorem 2 and then intersecting it with the linear constraints in $\mathcal{D}$ as indicated on the right-hand-side of (5).

3.2. The Case with $m > 1$

In this section, we consider the general case where $m > 1$ in $\mathcal{S}$. The coefficient matrix of $\mathcal{C}^l$ in (1) can be written as follows after a suitable rearrangement of columns and rows.

$$\begin{bmatrix} E^T & 0 & \dots & 0 & -E^T & I & -I & I & -I & 0 & 0 & \dots & 0 & 0 & \bar{I}^1 & -\bar{I}^1 & 0 & 0 & \dots & 0 & 0 \\ 0 & E^T & \dots & 0 & -E^T & I & -I & 0 & 0 & I & -I & \dots & 0 & 0 & 0 & 0 & \bar{I}^2 & -\bar{I}^2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & E^T & -E^T & I & -I & 0 & 0 & 0 & 0 & \dots & I & -I & 0 & 0 & 0 & 0 & \dots & \bar{I}^m & -\bar{I}^m \end{bmatrix}. \quad (6)$$

In the above matrix, each row block $j \in M$ represents the bilinear terms $y_j x_i$ for all $i \in N$. The first m column blocks correspond to the weights of the flow-balance constraints in Ξ multiplied by y_j for $j \in M$, which are denoted by γ_t^j for all $t \in T$ in the dual vector. The next column block represents the weights of the flow-balance constraints in Ξ multiplied by $1 - \sum_{j \in M} y_j$, which are denoted by θ_t for all $t \in T$ in the dual vector. The next two column blocks indicate the lower and upper bound constraints on variables in Ξ multiplied by $1 - \sum_{j \in M} y_j$, which are recorded by λ_i and μ_i , respectively, for all $i \in N$. The next $2m$ columns blocks correspond to the weights of the lower and upper bound constraints on variables in Ξ multiplied by y_j for $j \in M$, which are recorded by η_i^j and ρ_i^j , respectively, for all $i \in N$. The last $2m$ column blocks correspond to the weights of the positive and negative bilinear constraints in $\mathcal{S}$, which are represented by β_k^+ and β_k^- for all $k \in K_l$. For instance, for constraint $y_j x_i - z_k \geq 0$ with $(i, j) \in N \times M$ and $k \in K_l$, the elements of column k in $\bar{I}^j$ are all zero except the one in the row corresponding to the bilinear term $y_j x_i$ which is equal to one.

It is clear from the structure of (6) that this matrix does not have the TU property, implying that a result similar to that of Proposition 1 does not necessarily hold. Therefore, there is no guarantee that the aggregation weights for all the EC&R assignments obtained from the extreme points of $\mathcal{C}^l$ are 1. In fact, Example 2 shows that there exists EC&R assignments with aggregation weights that map to extreme points of $\mathcal{C}^l$ with components that are not all 0 or 1.

Example 2. Consider set $\mathcal{S}$ where Ξ describes the network polytope corresponding to network $G = (V, A)$ in Figure 1, and $\Delta = \{(y_1, y_2) \in \mathbb{R}_+^2 \mid y_1 + y_2 \leq 1\}$. Select class- l^+ corresponding to the base equality $y_1 x_{8,4} - z_l = 0$. Let $[\mathcal{I}_1, \mathcal{I}_2, \bar{\mathcal{I}} \mid \mathcal{J}, \bar{\mathcal{J}}]$ be an assignment for class- l^+ where $\mathcal{I}_1 = \{4^+, 3^+\}$, $\mathcal{I}_2 = \{2^-\}$, $\bar{\mathcal{I}} = \{1^+\}$, $\mathcal{J} = \{(4, 1)\}$, and $\bar{\mathcal{J}} = \{(2, 3)\}$. Next, we show that the above assignment is an EC&R assignment for class- l^+ when considering the dual weight 1 for all constraints except the bound constraint in $\mathcal{J}$ which has a dual weight of 2 in the aggregation. Specifically, we aggregate the base constraint $y_1 x_{8,4} - z_l \geq 0$ with weight 1, the positive flow-balance constraints at node 4 that is $x_{4,1} + x_{4,3} - x_{8,4} \geq f_4$ with weight y_1 , the positive flow-balance constraints at node 3 that is $x_{3,7} - x_{4,3} - x_{2,3} \geq f_3$ with weight y_1 , the negative flow-balance constraints at node 2 that is $x_{6,2} - x_{2,1} - x_{2,3} \geq -f_2$ with weight y_2 , the positive flow-balance constraints at node 1 that is $x_{1,5} - x_{4,1} - x_{2,1} \geq f_1$ with weight $1 - y_1 - y_2$, the lower bound constraint for arc (4, 1) that is $x_{4,1} \geq 0$ with weight $2(1 - y_1 - y_2)$, and the upper bound constraint for arc (2, 3) that is $x_{2,3} \leq 0$ with weight $1 - y_1 - y_2$. During this aggregation, six bilinear terms will be canceled, satisfying condition (C1) of the EC&R procedure. Further, at least one bilinear term for each constraint involved in the aggregation is canceled, satisfying condition (C2) of the EC&R procedure. Therefore, the above assignment is a valid EC&R assignment for class- l^+ . Next, we argue that the dual weight vector for this assignment corresponds to an extreme point of $\mathcal{C}^l$ in (1). For $\mathcal{S}$ in this example, the coefficient matrix of $\mathcal{C}^l$, as depicted in (6), has 16 rows corresponding to bilinear terms $y_j x_i$ for all $j = 1, 2$ and $i \in A$. It is easy to verify that the columns corresponding to the six constraints in the above EC&R assignment are linearly independent. As a result, we can form a basis by adding to the above six columns 10 more linearly independent columns corresponding to the constraints used in the relaxation step for the bilinear terms remaining in the aggregated inequality together with the columns that complete the basis. The resulting basis corresponds to a basic feasible solution where all variables (interpreted as the dual weights for constraints involved in the EC&R procedure) are 0 or 1, except the one associated with the column representing the lower bound constraint for arc (4, 1) which has a dual weight equal to 2. Therefore, there exists extreme points of $\mathcal{C}^l$ with components that are not all 0 or 1. $\square$

In light of the above observation, even though identifying the aggregation weights for the EC&R procedure for $\mathcal{S}$ is not as straightforward compared with that of $\mathcal{S}^1$, we next show that a generalization of the previously discussed tree structures can still be detected for a given EC&R assignment. First, we give a few definitions that will be used to derive these results.

Definition 1. Consider set $\mathcal{S}$ where Ξ describes the network polytope corresponding to network $G = (V, A)$. We define a *parallel network* $G^j = (V^j, A^j)$ for $j \in M$ to be a replica of G that represents the multiplication of flow variables x with y_j during the aggregation procedure. For instance, the selected nodes in the parallel network j represent the multiplication of their corresponding flow-balance constraints with variable y_j .

In view of Definition 1, to simplify presentation, we use the same node and arc notation across all parallel networks. In particular, given the network $G = (V, A)$, we refer to the replica of node $v \in V$ (resp. arc $a \in A$) in the parallel network G^j by v (resp. a) for all $j \in M$. Following this rule, given a subnetwork $\dot{G}$ of the parallel network G^j for some $j \in M$, we say that a node $v \in V$ (resp. arc $a \in A$) is adjacent to (resp. incident to) a node of G^j , if the replica of v in the parallel network G^j is adjacent to (resp. incident to) a node of G^j .

Definition 2. Consider set $\mathcal{S}$ where Ξ describes the network polytope corresponding to network $G = (V, A)$. Consider a collection of networks $\dot{G}^k = (\dot{V}^k, \dot{A}^k)$, for $k = 1, \dots, r$, where each $\dot{G}^k$ is a subnetwork of a parallel networks G^j for some $j \in M$. We say that this collection is *vertically connected through the connection nodes* $C_v \subseteq V$ and *connection arcs* $C_a \subseteq A$ if there exists an ordering $s_1, s_2, \dots, s_r$ of indices $1, \dots, r$ such that for each $i = 1, \dots, r-1$, there exists either (i) an arc $a \in C_a$ such that a is incident to a node of $\dot{G}^{s_{i+1}}$ and it is incident to a node of some subnetworks among $\dot{G}^{s_1}, \dots, \dot{G}^{s_i}$, or (ii) a set of nodes $v_1, \dots, v_p \in C_v$, each adjacent to the previous one in the original network G , such that v_1 is adjacent to a node of $\dot{G}^{s_{i+1}}$ and v_p is adjacent to a node of some subnetworks among $\dot{G}^{s_1}, \dots, \dot{G}^{s_i}$. In this definition, if node v_1 is in $\dot{G}^{s_{i+1}}$, it counts as an adjacent node to the connection node v_1 .

The next example illustrates the concepts introduced in Definitions 1 and 2.

Example 3. Consider set $\mathcal{S}$ where Ξ describes the network polytope corresponding to network $G = (V, A)$ in Figure 1, and $\Delta = \{(y_1, y_2) \in \mathbb{R}_+^2 \mid y_1 + y_2 \leq 1\}$. According to Definition 1, since $m=2$, we create two parallel networks $G^j = (V^j, A^j)$ for $j=1, 2$. Consider subnetworks $\dot{G}^1 = (\dot{V}^1, \dot{A}^1)$ and $\dot{G}^2 = (\dot{V}^2, \dot{A}^2)$ of the parallel network $G^1 = (V^1, A^1)$, and the subnetwork $\dot{G}^3 = (\dot{V}^3, \dot{A}^3)$ of the parallel network $G^2 = (V^2, A^2)$, where $\dot{V}^1 = \{1, 5\}$, $\dot{A}^1 = \{(1, 5)\}$, $\dot{V}^2 = \{7\}$, $\dot{A}^2 = \emptyset$, $\dot{V}^3 = \{2, 6\}$, $\dot{A}^3 = \{(6, 2)\}$. Further, define $C_v = \{3\}$ and $C_a = \{(2, 1)\}$. According to Definition 2, the subnetworks $\dot{G}^k$, for $k=1, 2, 3$, are vertically connected through the connection nodes C_v and connection arcs C_a due to the following observations. Consider the ordering $s_1 = 1, s_2 = 3$, and $s_3 = 2$ of the indices 1, 2, 3 of the subnetworks. It follows that $\dot{G}^{s_2}$ is vertically connected to $\dot{G}^{s_1}$ through arc $(2, 1) \in C_a$ as it satisfies the condition (i) of Definition 2 because it is incident to node 1 of $\dot{G}^{s_1}$ and node 2 of $\dot{G}^{s_2}$. Similarly, $\dot{G}^{s_3}$ is vertically connected to $\dot{G}^{s_2}$ through node $3 \in C_v$ as it satisfies the condition (ii) of Definition 2 because it is adjacent to node 2 of $\dot{G}^{s_2}$ and node 7 of $\dot{G}^{s_3}$. $\square$

Now, we are ready to show the relationship between an EC&R assignment and a special graphical structure in the underlying network.

Proposition 4. Consider set $\mathcal{S}$ with Ξ that represents the network polytope corresponding to the network $G = (V, A)$. Let $[\mathcal{I}_1, \dots, \mathcal{I}_m, \tilde{\mathcal{I}} \mid \mathcal{J}, \tilde{\mathcal{J}}]$ be an EC&R assignment that leads to a nontrivial facet-defining inequality of $\text{conv}(\mathcal{S})$. Assume that $\cup_{j \in M} \mathcal{I}_j \neq \emptyset$. For each $j \in M$, define $\tilde{\mathcal{I}}^j = \{i \in V \mid i^\pm \in \mathcal{I}_j\}$, $\tilde{\mathcal{I}} = \{i \in V \mid i^\pm \in \tilde{\mathcal{I}}\}$, and $\tilde{\mathcal{J}} = \{i \in A \mid i \in \mathcal{J} \cup \tilde{\mathcal{J}}\}$. Then, there exist forests $\tilde{F}^j$ in the parallel network G^j for $j \in M$, each composed of trees $\tilde{T}_k^j$ for $k \in \Gamma_j$, where Γ_j is an index set, such that

- (i) the forest $\tilde{F}^j$ is composed of the nodes in $\tilde{\mathcal{I}}^j$ for all $j \in M$,
- (ii) the collection of the trees $\tilde{T}_k^j$ for all $k \in \Gamma_j$ and all $j \in M$ are vertically connected through connection nodes $\tilde{\mathcal{I}}$ and connection arcs $\tilde{\mathcal{J}}$,
- (iii) the collection of all nodes in $\tilde{F}^j$ for all $j \in M$ together with the connection nodes $\tilde{\mathcal{I}}$ form a tree in G , which has at least one incident node to each connection arc in $\tilde{\mathcal{J}}$.

Proof. We show the result by proving conditions (i)–(iii). First, we may assume that the given EC&R assignment corresponds to class- $l^\pm$ for some $l \in K$. Since $[\mathcal{I}_1, \dots, \mathcal{I}_m, \tilde{\mathcal{I}} \mid \mathcal{J}, \tilde{\mathcal{J}}]$ leads to a nontrivial facet-defining inequality of $\text{conv}(\mathcal{S})$, its corresponding dual weights in the aggregation should represent an extreme point of $\mathcal{C}^l$ defined in (1). This extreme point is associated with a basis in the coefficient matrix (6). In this basis, the subset of the column block that contains E^T in the row block j represents the flow-balance constraints (multiplied with y_j) for the nodes $i \in \tilde{\mathcal{I}}^j$, which can be viewed as the selected nodes in the parallel network G^j for $j \in M$. Further, the rows in the row block j represent the flow variables (multiplied with y_j) for each arc in G , which can be viewed as an arc in the parallel network G^j . We may reorder the columns and rows of this basis corresponding to each parallel network G^j to obtain a diagonal formation composed of diagonal blocks $E_{j,k}^T$ for k in an index set Γ_j . It follows from this structure that the nodes corresponding to the columns of $E_{j,k}^T$ in the parallel network G^j are connected via arcs of G^j represented by the matrix rows. Therefore, these nodes can form a tree $\tilde{T}_k^j$ for $k \in \Gamma_j$, the collection of which represents a forest $\tilde{F}^j$ for all $j \in M$, satisfying condition (i) of the proposition statement.

For condition (ii), considering the aforementioned diagonal block structure and representing the subset of column blocks of (6) containing $\tilde{I}$ and I in the basis by one block with $\pm$ sign (as only one of them can be selected in

the basis), we can write the resulting system of equations for the associated basic feasible solution as follows

$$\left[\begin{array}{cccccccccccccccccccc} E_{1,1}^\top & 0 & 0 & & & & & & & & & & & & & & & & \\ 0 & \ddots & 0 & & 0 & \dots & 0 & -E^\top & \pm I & \pm I & 0 & \dots & 0 & 0 & \pm \bar{I}^1 & 0 & \dots & 0 & 0 & C_1 \\ 0 & 0 & E_{1,|\Gamma_1|}^\top & & & & & & & & & & & & & & & & \\ & & & E_{2,1}^\top & 0 & 0 & & & & & & & & & & & & & \\ & & 0 & 0 & \ddots & 0 & \dots & 0 & -E^\top & \pm I & 0 & \pm I & \dots & 0 & 0 & 0 & \pm \bar{I}^2 & \dots & 0 & 0 & C_2 \\ & & 0 & 0 & 0 & E_{2,|\Gamma_2|}^\top & & & & & & & & & & & & & \\ \vdots & & \vdots & & \ddots & & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ & & & & & E_{m,1}^\top & 0 & 0 & & & & & & & & & & & \\ 0 & & 0 & & \dots & 0 & \ddots & 0 & -E^\top & \pm I & 0 & 0 & \dots & \pm I & 0 & 0 & 0 & \dots & \pm \bar{I}^m & 0 & C_m \\ & & & & 0 & 0 & E_{m,|\Gamma_m|}^\top & & & & & & & & & & & & \\ 0 & & 0 & & \dots & 0 & & -E^\top & \pm I & 0 & 0 & \dots & 0 & \pm I & 0 & 0 & \dots & 0 & \pm \bar{I}^{1,\dots,m} & C_{m+1} \\ 0 & & 0 & & \dots & 0 & & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & C_{m+2} \end{array} \right], \quad (7)$$

where the last column in the basis corresponds to the constraints that have weights 0 in the basic feasible solution and are added to complete the basis, and where the last row represents all bilinear terms that do not appear in any constraints during aggregation. Further, the row block next to the last row corresponds to the bilinear terms that appear during aggregation but not in any of the selected flow-balance constraints; the matrix $\pm \bar{I}^{1,\dots,m}$ denotes the bilinear constraints in $\mathcal{S}$ that contain these bilinear terms and could be used during the relaxation step. In the above basis, the column block that contains $-E^\top$ represents the flow-balance constraints (multiplied with $1 - \sum_{j \in M} y_j$) corresponding to the nodes in $\tilde{\mathcal{I}}$. Similarly, the column block that contains $\pm I$ in all rows represents the bound constraints on the flow variables (multiplied with $1 - \sum_{j \in M} y_j$) corresponding to the arcs in $\tilde{\mathcal{J}}$. We refer to the column group composed of the columns of $E_{j,k}^\top$ for any $j \in M$ and $k \in \Gamma_j$ as the column group representing the nodes of the tree $\tilde{T}_k^j$. It is clear from the diagonal structure of the submatrix containing $E_{j,k}^\top$ that the column groups representing the nodes of $\tilde{T}_k^j$ are all *arc disjoint*, that is, there are no two columns from different groups that have a nonzero element in the same row. We claim that there exists an ordering $(s_1, r_1), (s_2, r_2), \dots, (s_h, r_h)$ of the pairs (j, k) for all $j \in M$ and $k \in \Gamma_j$ such that for each column group representing the nodes of $\tilde{T}_{r_i}^{s_i}$, for all $i = 2, \dots, h$, there exists either (i) a column from the column blocks corresponding to $\tilde{\mathcal{J}}$ that has a nonzero element in a row corresponding to a row of $E_{s_i, r_i}^\top$ and a nonzero element in a row corresponding to a row of $E_{s_i, r_i}^\top$ for some $t \in \{1, \dots, i-1\}$, or (ii) a sequence of columns in the column block corresponding to $\tilde{\mathcal{I}}$, each sharing a nonzero element in a common row with the previous one, such that the first column has a nonzero element in a row corresponding to a row of $E_{s_i, r_i}^\top$ and the last column has a nonzero element in a row corresponding to a row of $E_{s_i, r_i}^\top$ for some $t \in \{1, \dots, i-1\}$. Assume by contradiction that no such ordering exists. Therefore, we can partition the rows in the first $m+1$ row blocks of (7) in such a way that no two rows in all columns composed of the columns in $E_{j,k}^\top$ for all $j \in M$ and $k \in \Gamma_j$ and those corresponding to the columns of $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{J}}$ have all their nonzero elements in the rows of one of these partitions. In this case, considering that the column blocks composed of $\pm I$ and those composed of $\pm \bar{I}^j$ have exactly one nonzero element in the basis, we can compactly rewrite the system of equations for the basic feasible solution as follows:

$$\left[\begin{array}{ccccccc} \pm \bar{E}_1 & \pm I_1 & \pm \bar{I}_1 & 0 & 0 & 0 & \bar{C}_1 \\ 0 & 0 & 0 & \pm \bar{E}_2 & \pm I_2 & \pm \bar{I}_2 & \bar{C}_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & C_{m+2} \end{array} \right] \begin{bmatrix} + \\ + \\ 0 \end{bmatrix} = \begin{bmatrix} \pm e^l \\ 0 \\ 0 \end{bmatrix}, \quad (8)$$

where the first and second row blocks respectively correspond to the first and second partitions discussed above. In (8), e^l is a unit vector whose elements are all zero except that corresponding to the row representing $y_{j'} x_{i'}$ for some i', j' that satisfy $A_{j'i'}^l = 1$, which is equal to 1. We may assume without the loss of generality that the row containing this nonzero element in e^l belongs to the first row block. All these columns except the ones in the last column block have positive weights because the associated constraints are assumed to be used in the aggregation. These weights are denoted by $+$ in the first two row blocks of the vector multiplied with this matrix. It follows that the linear combination of the columns in the second column block of the basis matrix with positive weights yields the zero vector. This shows that the columns are linearly dependent, a contradiction. Now, consider the ordering $(s_1, r_1), (s_2, r_2), \dots, (s_h, r_h)$ described above. It follows that for each $i = 2, \dots, h$, there exists either (i) a column from the column block

corresponding to $\tilde{\mathcal{J}}$ that has a nonzero element in a row corresponding to a row of $E_{s_i, r_i}^\top$ and a nonzero element in a row corresponding to a row of $E_{s_t, r_t}^\top$ for some $t \in \{1, \dots, i-1\}$, or (ii) a sequence of columns in the column block corresponding to $\tilde{\mathcal{I}}$, each sharing a nonzero element in a common row with the previous one, such that the first column has a nonzero element in a row corresponding to a row of $E_{s_i, r_i}^\top$ and the last column has a nonzero element in a row corresponding to a row of $E_{s_t, r_t}^\top$ for some $t \in \{1, \dots, i-1\}$. First, consider the case (i) in the above either-or argument holds for a column $k \in \tilde{\mathcal{J}}$. This column has nonzero elements in the rows representing arc k in all subnetworks G^j for $j \in M$. Matrix $E_{s_i, r_i}^\top$ has a nonzero element in row k if the tree $\bar{T}_{r_i}^{s_i}$ has an incident node to arc k . We conclude that $\bar{T}_{r_i}^{s_i}$ and $\bar{T}_{r_t}^{s_t}$ have at least one node incident to k , satisfying criterion (i) in Definition 2. Second, consider the case (ii) in the above either-or argument holds for a sequence $k_1, \dots, k_p$ of the nodes corresponding to $\tilde{\mathcal{I}}$ where $p \leq |\tilde{\mathcal{I}}|$. Any such column, say k_1 , has nonzero elements in the rows representing the arcs that are incident to node k_1 in all parallel networks G^j for $j \in M$. Therefore, since each column contains a nonzero element in a common row with the previous one, the nodes corresponding to these columns must be adjacent to one another in G . Further, since the column corresponding to k_1 has a nonzero element in a row corresponding to a row of $E_{s_i, r_i}^\top$, we conclude that k_1 is adjacent to a node in $\bar{T}_{r_i}^{s_i}$, which means that either k_1 belongs to this tree, or it is adjacent to a node of this tree. A similar argument can be made about node k_p and the tree $\bar{T}_{r_t}^{s_t}$. This satisfies criterion (ii) of Definition 2. This proves condition (ii) of the proposition statement due to Definition 2.

For condition (iii), we show there exists a sequence $v_1, \dots, v_q$ of all the nodes in $\cup_{j \in M} \tilde{\mathcal{I}}^j \cup \tilde{\mathcal{I}}$, where $q = |\cup_{j \in M} \tilde{\mathcal{I}}^j \cup \tilde{\mathcal{I}}|$ and $v_1 \in \cup_{j \in M} \tilde{\mathcal{I}}^j$, such that every node v_i is adjacent to at least one node in $v_1, \dots, v_{i-1}$ for every $i = 2, \dots, q$. We may assume that v_1 is incident to arc i' defined previously that is associated with the base equality l . For other cases, where i' is not incident to any nodes in $\cup_{j \in M} \tilde{\mathcal{I}}^j$, the argument will be similar with an adjustment of the partitions described below. It follows from the previously proven conditions (i) and (ii) of the problem statement, as well as Definition 2 that there exists a sequence $\bar{v}_1, \dots, \bar{v}_p$ of all the nodes in $\cup_{j \in M} \tilde{\mathcal{I}}^j \cup \hat{\mathcal{I}}$ for some $\hat{\mathcal{I}} \subseteq \tilde{\mathcal{I}}$ such that $\bar{v}_i$ is adjacent to at least one node in $\bar{v}_1, \dots, \bar{v}_{i-1}$ for every $i = 2, \dots, p$. We claim that there exists a sequence $\hat{v}_1, \dots, \hat{v}_{q-p}$ of all the nodes in $\tilde{\mathcal{I}} \setminus \hat{\mathcal{I}}$ such that $\hat{v}_i$ is adjacent to at least one node in $\bar{v}_1, \dots, \bar{v}_p, \hat{v}_1, \dots, \hat{v}_{i-1}$ for every $i = 1, \dots, q-p$. Assume by contradiction that no such sequence exists. Then, we can use an argument similar to that of case (ii) above to partition the rows of (7) in such a way that all columns corresponding to the nodes in $\bar{v}_1, \dots, \bar{v}_p, \hat{v}_1, \dots, \hat{v}_t$ have all their nonzero elements in the first partition, and the columns corresponding to all the remaining nodes $\hat{v}_{t+1}, \dots, \hat{v}_{q-p}$ have all their nonzero elements in the second partition. Then, a similar argument to that following (8) will show that the columns in the second group are linearly dependent, a contradiction. The case for the second part of the statement regarding the connection arcs in $\tilde{\mathcal{J}}$ can be shown similarly. $\square$

Although Proposition 4 shows that an EC&R assignment corresponds to a forest structure in the parallel networks created from the underlying network in $\mathcal{S}$, the converse result—similar to the one presented for set $\mathcal{S}^1$ —does not hold here. More specifically, a forest structure that satisfies the conditions of Proposition 4 does not necessarily lead to a valid EC&R assignment, and even when it does, the calculation of the aggregation weights to satisfy the EC&R conditions is not as straightforward; see Example 4 below.

Example 4. Consider set $\mathcal{S}$ where Ξ describes the network polytope corresponding to network $G = (V, A)$ in Figure 1, and $\Delta = \{(y_1, y_2) \in \mathbb{R}_+^2 \mid y_1 + y_2 \leq 1\}$. Select class- l^+ corresponding to the base equality $y_1 x_{1,5} - z_l = 0$. In the parallel network G^1 , we select the forest $\bar{F}^1$ composed of the tree $\bar{T}_1^1$ with the node set $\tilde{\mathcal{I}}^1 = \{1, 2\}$. In the parallel network G^2 , we select the forest $\bar{F}^2$ composed of the tree $\bar{T}_1^2$ with the node set $\tilde{\mathcal{I}}^2 = \{2, 6\}$. We select the connection node set $\tilde{\mathcal{I}} = \{6\}$, and the connection arc set $\tilde{\mathcal{J}} = \emptyset$. It is easy to verify that these sets satisfy the conditions (i)–(iii) of Propositions 4. However, we cannot find an aggregation weight for the flow-balance constraints corresponding to the nodes in the above sets that yields a cancellation of at least 5 bilinear terms. As a result, there is no EC&R assignment that matches the considered forest structure. $\square$

A common way to circumvent the above-mentioned difficulty in obtaining valid EC&R assignments and their aggregation weights is to aim at a special class of EC&R assignments with more specific attributes that can be used to strengthen the connection between an EC&R assignment and its corresponding network structure. An important example of such class is the class of EC&R assignments that are obtained through *pairwise cancellation*. In this procedure, each cancellation of bilinear terms is obtained by aggregating two constraints. This definition includes the bilinear terms that are *canceled* during the relaxation step, that is, the constraint used to relax the remaining bilinear terms counts as one of the two constraints in the preceding statement. Following this procedure, the aggregation weight for each constraint can be determined successively as the constraint is added to the assignment to ensure the satisfaction of the EC&R conditions. The next result shows that the aggregation weights for all constraints used in the EC&R assignments obtained through pairwise cancellation are 1.

Proposition 5. Consider set $\mathcal{S}$ where Ξ describes the network polytope corresponding to network $G = (V, A)$. Let $[\mathcal{I}_1, \dots, \mathcal{I}_m, \bar{\mathcal{I}} | \mathcal{J}, \bar{\mathcal{J}}]$ be an EC&R assignment for class- $l^\pm$ for some $l \in K$ corresponding to pairwise cancellation. Then, the aggregation weights for all constraints used in this assignment are 1.

Proof. Let $\bar{\pi}^l$ be the solution vector for the system of Equation (1) of $\mathcal{C}^l$ corresponding to the aggregation weights of the given EC&R assignment. We may rewrite this system of equations as follows by rearranging its rows and columns.

$$\begin{bmatrix} P_1 & 0 & C_1 \\ P_2 & \pm I & C_2 \\ 0 & 0 & C_3 \end{bmatrix} \begin{bmatrix} \bar{\pi}_1^l \\ \bar{\pi}_2^l \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \pm e^l \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (9)$$

In the coefficient matrix of (9), the first row block represents the bilinear terms that are canceled during aggregation. The second row block corresponds to the remaining bilinear terms in the aggregated inequality that are relaxed in the last step of the EC&R procedure. The last row block represents all the bilinear terms that are not involved in the aggregation procedure. Further, in this matrix, the first column block corresponds to the constraints used in the aggregation, whose aggregation weights in the solution vector $\bar{\pi}^l$ are denoted by $\bar{\pi}_1^l$. The second column block corresponds to the variable bound constraints in Ξ as well as the bilinear constraints in $\mathcal{S}$ used in the EC&R procedure to relax the remaining bilinear terms in the aggregated inequality, whose weights in the solution vector $\bar{\pi}^l$ are denoted by $\bar{\pi}_2^l$. The last column block represents all other constraints that are not used during the EC&R procedure and their weights in the solution vector $\bar{\pi}^l$ are zero. Finally, e^l on the right-hand-side of this system is a unit vector whose elements are all zeros except that corresponding to the row representing $y_{i'}x_{i'}$ for some i', j' that satisfy $A_{j'i'}^l = 1$, which is equal to 1. It is clear that this row belongs to the first row block since according to the EC&R condition (C2), the bilinear term in the base equality l must be canceled during the aggregation procedure when the assignments are not empty. It follows from the Equation (9) that $P_1 \bar{\pi}_1^l = \pm e^l$. Next, we analyze the structure of P_1 . Note that all elements of P_1 belong to $\{0, -1, 1\}$ because it is a submatrix of (6) that represents the coefficients of the constraints in $\mathcal{S}$. Considering that the columns of P_1 represent the constraints used in the aggregation except the base equality (as that constraint has been moved to the right-hand-side to form $\mathcal{C}^l$), and that the rows of P_1 correspond to the canceled bilinear terms during aggregation, according to condition (C1) of EC&R, we conclude that the number of rows of P_1 is no smaller than the number of columns of P_1 . Further, it follows from condition (C2) of EC&R that each constraint used in the aggregation (after being multiplied with its corresponding weight) will have at least one bilinear term canceled, which implies that each column of P_1 has at least one nonzero element. The assumption of pairwise cancellation for the given EC&R assignment implies that each canceled bilinear term corresponding to the rows of P_1 are obtained through aggregation of exactly two constraints. As a result, each row of P_1 must contain exactly two nonzero elements, except for the row corresponding to the bilinear term $y_{i'}x_{i'}$ that appears in the base equality $y_{i'}x_{i'} - z_l = 0$ which must have only one nonzero element because the weight of the base equality has been fixed at ± 1 and its column has been moved to the right-hand-side of the equation captured by $\pm e^l$; see the derivation of (1). Therefore, we may rearrange the rows and columns of the matrices in this equation to obtain the form:

$$\begin{bmatrix} \pm 1 & 0 & \dots & 0 \\ \{0, \pm 1\} & \pm 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \{0, \pm 1\} & \{0, \pm 1\} & \dots & \pm 1 \\ \{0, \pm 1\} & \{0, \pm 1\} & \dots & \{0, \pm 1\} \\ \vdots & \vdots & \ddots & \vdots \\ \{0, \pm 1\} & \{0, \pm 1\} & \dots & \{0, \pm 1\} \end{bmatrix} \bar{\bar{\pi}}_1^l = \pm e^1, \quad (10)$$

where $\bar{\bar{\pi}}_1^l$ is composed of the elements of $\bar{\pi}_1^l$ that are rearranged to match the rearrangement of columns of P_1 in the above form, and where the first row corresponds to the bilinear term $y_{i'}x_{i'}$ so that the right-hand-side vector becomes $\pm e^1$. It follows from the above discussion about the structure of P_1 and Equation (10) that all components of $\bar{\bar{\pi}}_1^l$ must be equal to 1 as they need to be nonnegative. Finally, for the equations in the second row block of (9), we have that $P_2 \bar{\pi}_1^l \pm I \bar{\pi}_2^l = \mathbf{0}$. It follows from the pairwise cancellation assumption that each row of P_2 contains exactly one nonzero element as it corresponds to a remaining bilinear term in the aggregation inequality. Since all of the elements in P_2 belong to $\{0, -1, 1\}$, and all the components in $\bar{\pi}_1^l$ are equal to 1, it must hold that $\bar{\pi}_2^l = I$. $\square$

Remark 3. Consider the special case with $m=1$. As described in the proof of Proposition 2, for any $i \in N \setminus \{l\}$ (resp. for $i=l$), there are two (resp. three) possible scenarios for constraints that could be used in the aggregation to cancel a bilinear term $y_l x_i$. Since the aggregation weights for all constraints are 1 in this case (see Proposition 1), we conclude that each cancellation is obtained through aggregation of exactly two constraints. Further, any remaining bilinear term in the aggregated inequality corresponds to an arc that is incident to exactly one node of the tree associated with the EC&R assignment (see Proposition 2), which implies that each such bilinear term appears in exactly one constraint during aggregation. As a result, all EC&R inequalities for the case with $m=1$ can be obtained through pairwise cancellation.

Although the EC&R inequalities obtained through pairwise cancellation do not necessarily produce a full convex hull description for $\mathcal{S}$, the result of Proposition 5 provides three important advantages: (i) it generalizes the convexification results for the case with $m=1$ as described in Remark 3; (ii) it can produce inequalities stronger than those obtained by applying Theorem 2 to relaxations of $\mathcal{S}$ that contain one y variable at a time, because it considers all the y variables in their original simplex set Δ_m ; and (iii) it enables us to derive explicit EC&R inequalities cognizant of the underlying network structure without the need to search for the aggregation weights that satisfy the EC&R conditions, as will be shown in the sequel. These advantages are corroborated by the computational experiments presented in Section 4. The next proposition shows that the pairwise cancellation property provides more information about the forest structure presented in Proposition 4.

Proposition 6. Consider the setting of Proposition 4, and assume that the EC&R assignment $[\mathcal{I}_1, \dots, \mathcal{I}_m, \tilde{\mathcal{I}} | \mathcal{J}, \tilde{\mathcal{J}}]$ has the pairwise cancellation property. Further, let this assignment correspond to a class- $l^\pm$ for some $l \in K$ such that $A_{l'j'}^l = 1$ for some $(i', j') \in N \times M$. Then, in addition to the outcome of Proposition 4, we have that

- (i) arc i' is either in $\tilde{\mathcal{J}}$ or incident to exactly one node in $\tilde{\mathcal{I}} \cup \tilde{\mathcal{I}}^{j'}$, but not both,
- (ii) each arc in $\tilde{\mathcal{J}}$ is incident to at most one node in $\tilde{\mathcal{I}} \cup \tilde{\mathcal{I}}^j$ for each $j \in M$,
- (iii) each node in $\tilde{\mathcal{I}} \cap \tilde{\mathcal{I}}^j$, for $j \in M \setminus \{j'\}$ (resp. $j = j'$), is adjacent to no other nodes in that set and no arcs in $\tilde{\mathcal{J}}$ (resp. $\tilde{\mathcal{J}} \cup \{i'\}$).

Proof. For case (i), it follows from condition (C2) of the EC&R procedure that the bilinear term $y_{j'} x_{i'}$ in the base equality must be canceled during aggregation. Further, according to the pairwise cancellation property, there must be exactly one constraint in the aggregation in addition to the base equality that would contain $y_{j'} x_{i'}$ after multiplication with the corresponding dual weight. There are two possible scenarios. The first possibility is that the bound constraints for $x_{i'}$ are used in the aggregation, which implies that arc i' is a connection arc and belongs to $\tilde{\mathcal{J}}$. The second possibility is that the flow-balance constraint at either node $t(i')$ or $h(i')$, but not both, is used in the aggregation, which implies that i' is incident to exactly one node in $\tilde{\mathcal{I}} \cup \tilde{\mathcal{I}}^{j'}$.

For case (ii), consider an arc $i \in \tilde{\mathcal{J}}$. Therefore, either of the bound constraints $x_i \geq 0$ or $u_i - x_i \geq 0$ is used in the aggregation with weight $1 - \sum_{j \in M} y_j$. It follows from the pairwise cancellation property that, for each $j \in M$, there can be at most one additional constraint in the aggregation that contains a term $y_j x_i$. The only possibility for such a constraint is the flow-balance constraint at either node $t(i')$ or $h(i')$, but not both. We conclude that i is incident to at most one node in $\tilde{\mathcal{I}} \cup \tilde{\mathcal{I}}^j$ for each $j \in M$.

For case (iii), consider a node $i \in \tilde{\mathcal{I}} \cap \tilde{\mathcal{I}}^j$ for some $j \in M \setminus \{j'\}$. Therefore, the aggregation contains the (positive or negative) flow-balance constraint at node i multiplied with $1 - \sum_{j \in M} y_j$ due to $i \in \tilde{\mathcal{I}}$, together with the (positive or negative) flow-balance constraint at node i multiplied with y_j due to $i \in \tilde{\mathcal{I}}^j$. Therefore, the bilinear terms $y_j x_k$ for all $k \in \delta^+(i) \cup \delta^-(i)$ already appear in two constraints, which implies that they cannot appear in any other constraints during aggregation. As a result, the bound constraints for each variable x_k corresponding to arc k cannot be included in $\tilde{\mathcal{J}}$. Similarly, the flow-balance constraint at node $h(k)$ for any $k \in \delta^+(i)$ and at node $t(k)$ for any $k \in \delta^-(i)$ cannot be included in the aggregation, which implies that i cannot be adjacent to any other nodes in $\tilde{\mathcal{I}} \cap \tilde{\mathcal{I}}^j$. The proof for the case where $j = j'$ follows from a similar argument. $\square$

As noted earlier, an important consequence of the pairwise cancellation property is providing the ability to derive the converse statement to those of Propositions 4 and 6, which identifies EC&R assignments based on a special forest structure in the underlying network. The procedure to obtain such an EC&R assignment is given in Theorem 3, which makes use of Algorithm 1 to determine the aggregation weights for deriving the corresponding EC&R inequalities.

Theorem 3. Consider set $\mathcal{S}$ with Ξ that represents the network polytope corresponding to the network $G = (V, A)$. Let $\bar{F}^j$, for each $j \in M$, be a forest in the parallel network G^j , composed of trees $\bar{T}_k^j$ for $k \in \Gamma_j$, where Γ_j is an index set, that satisfies the conditions (i)–(iii) of Propositions 4 and 6 with the corresponding node sets $\tilde{\mathcal{I}}^j$, the connection node set $\tilde{\mathcal{I}}$, the connection arc set $\tilde{\mathcal{J}}$, and the class l . Then, the assignment $[\mathcal{I}_1, \dots, \mathcal{I}_m, \tilde{\mathcal{I}} | \mathcal{J}, \tilde{\mathcal{J}}]$ obtained from Algorithm 1 is an EC&R assignment for class- $l^\pm$.

Proof. First, we argue that conditions (i)–(iii) of Proposition 4 imply that each member of the sets $\tilde{\mathcal{I}}$, $\tilde{\mathcal{J}}$, and $\tilde{\mathcal{T}}^j$ for $j \in M$ receives a label assignment through the steps of Algorithm 1, that is, the member is added to set $\mathcal{D}$ defined in that algorithm. It follows from condition (i) of Proposition 4 that once a member of the node subset in $\tilde{\mathcal{T}}^j$, for each $j \in M$, that represents a tree $\tilde{T}_k^j$ with $k \in \Gamma_j$ is added to $\mathcal{D}$, all the remaining nodes in $\tilde{T}_k^j$ are eventually added to $\mathcal{D}$ because of the loop in lines 11–13 in the algorithm, as all nodes of the tree are connected. Condition (ii) of Proposition 4 implies that all trees $\tilde{T}_k^j$ for $k \in \Gamma_j$ and $j \in M$ are connected through an appropriate sequence of the tree nodes, the connection nodes in $\tilde{\mathcal{I}}$, and the connection arcs in $\tilde{\mathcal{J}}$. Consequently, the loops in lines 10–44 of the algorithm ensure that each member of these sets is visited following that sequence and becomes added to $\mathcal{D}$. Further, condition (iii) of Proposition 4 suggests that each member in the sets $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{J}}$ is connected to the subgraph composed of the set of all tree nodes in $\tilde{\mathcal{T}}^j$ and their associated connection nodes and connection arcs. As a result, there exists a sequence of adjacent nodes and arcs that lead to each member of $\tilde{\mathcal{I}}$ and $\tilde{\mathcal{J}}$, thereby getting added to $\mathcal{D}$.

Second, we show that each bilinear term created during the aggregation can appear in at most two constraints. There are four cases. In case 1, consider the bilinear term $y_{j'}x_{i'}$ that appears in the base equality l . Condition (i) of Proposition 6 implies that this bilinear term can appear in exactly one other constraint, which could be either the bound constraint on variable $x_{i'}$ (which would be included in $\tilde{\mathcal{J}}$) or the flow-balance constraint at one of the incident nodes to i' (which would be included in $\tilde{\mathcal{T}}^j \cup \tilde{\mathcal{I}}$). In case 2, consider a bilinear term y_jx_i , for some $j \in M$, that appears in the bound constraint on variable x_i for any arc $i \in \tilde{\mathcal{J}}$. Condition (ii) of Proposition 6 implies that this bilinear term can appear in at most one other constraint, which could be the flow-balance constraint at one of the incident nodes to i (which would be included in $\tilde{\mathcal{T}}^j \cup \tilde{\mathcal{I}}$). In case 3, consider a bilinear term y_jx_i , for some $j \in M$, that appears in the flow-balance constraint at an incident node of arc i after being multiplied with both y_j (i.e., the node being in $\tilde{\mathcal{T}}^j$) and $1 - \sum_{j \in M} y_j$ (i.e., the node being in $\tilde{\mathcal{I}}$). Condition (iii) of Proposition 6 implies that this bilinear term cannot appear in any other constraints during aggregation. In case 4, consider a bilinear term y_jx_i , for some $j \in M$, that appears in the flow-balance constraint at an incident node of arc i that is not in $\tilde{\mathcal{T}}^j \cap \tilde{\mathcal{I}}$. It follows from condition (iii) of Proposition 4 that this bilinear term can appear in at most one other constraint because of the tree structure of all the nodes in $\cup_{j \in M} \tilde{\mathcal{T}}^j \cup \tilde{\mathcal{I}}$.

Third, we discuss that, for any $k \in \mathcal{D}$ that has been newly added to this set, its label value has been determined through lines 10–44 of Algorithm 1 in such a way that, for a member $i \in \cup_{j \in M} \tilde{\mathcal{T}}^j \cup \tilde{\mathcal{I}} \cup \tilde{\mathcal{J}}$ that has been previously added to $\mathcal{D}$ and is adjacent/incident to i , the bilinear term that commonly appears in the weighted constraints corresponding to both i and k is canceled. For instance, consider the case where $i \in \tilde{\mathcal{I}}$ (line 22 of the algorithm) and $k \in \tilde{\mathcal{T}}^j$ for some $j \in M$ is an adjacent node to i (line 26 of the algorithm). Assume that $l(i) = +$, and that arc $a \in A$ is such that $t(a) = i$ and $h(a) = k$. It follows from line 27 of the algorithm that $l(k) = -$. Considering the assignment rule in lines 48 and 52 of the algorithm, we should aggregate the constraint $\sum_{r \in \delta^+(i) \setminus \{p\}} x_r - \sum_{r \in \delta^-(i)} x_r + x_p \geq f_i$ with weight $1 - \sum_{j \in M} y_j$, together with the constraint $-\sum_{r \in \delta^+(k)} x_r + \sum_{r \in \delta^-(k) \setminus \{p\}} x_r + x_p \geq -f_k$ with weight y_j , which results in the cancellation of the bilinear term y_jx_p . A similar argument can be made for any other possible case in Algorithm 1.

Combining all the results shown in the previous parts, that is, (I) each member of the sets $\tilde{\mathcal{I}}$, $\tilde{\mathcal{J}}$, and $\tilde{\mathcal{T}}^j$ for $j \in M$ receives a label assignment and is added to $\mathcal{D}$; (II) each bilinear term created during the aggregation can appear in at most two constraints; and (III) for any $k \in \mathcal{D}$, its label value is determined in such a way that the bilinear term that is common between the weighted constraints corresponding to i and a previously added member k in $\mathcal{D}$ is canceled, we conclude that at least $|\tilde{\mathcal{I}}| + |\tilde{\mathcal{J}}| + \sum_{j \in M} |\tilde{\mathcal{T}}^j|$ bilinear terms will be canceled during aggregation in the desired assignment $[\mathcal{I}_1, \dots, \mathcal{I}_m, \tilde{\mathcal{I}} | \mathcal{J}, \tilde{\mathcal{J}}]$. This satisfies the EC&R conditions (C1). Finally, the above argument also implies that each flow-balance constraint at the nodes in $\cup_{j \in M} \tilde{\mathcal{T}}^j \cup \tilde{\mathcal{I}}$, and each variable bound constraint for the arcs in $\tilde{\mathcal{I}}$ will have at least one of their bilinear terms (after being multiplied with appropriate weights) canceled because each such node or arc will eventually be added to $\mathcal{D}$ when it receives a label for the desired cancellation. This satisfies the EC&R condition (C2). We conclude that $[\mathcal{I}_1, \dots, \mathcal{I}_m, \tilde{\mathcal{I}} | \mathcal{J}, \tilde{\mathcal{J}}]$ is an EC&R assignment. $\square$

Algorithm 1 (Derive an EC&R Assignment Associated with a Forest Structure)

Require: network $G = (V, A)$, forest node sets $\tilde{\mathcal{T}}^j$ for $j \in M$, connection node set $\tilde{\mathcal{I}}$, connection arc set $\tilde{\mathcal{J}}$, class l with $(i', j') \in N \times M$ such that $A_{i'j'}^l = 1$, and a class sign indicator $\pm$, that satisfy the conditions of Theorem 3

Ensure: the EC&R assignment $[\mathcal{I}_1, \dots, \mathcal{I}_m, \tilde{\mathcal{I}} | \mathcal{J}, \tilde{\mathcal{J}}]$ for class- $l^\pm$

- 1: assign an empty label denoted by $l(i)$ to each node $i \in \tilde{\mathcal{I}} \cup \cup_{j \in M} \tilde{\mathcal{T}}^j$ and each arc $i \in \tilde{\mathcal{J}} \cup \{l\}$, and define set $\mathcal{D} = \emptyset$
- 2: set $l(i') =$ the class sign indicator, and let $k \in \tilde{\mathcal{T}}^{j'} \cup \tilde{\mathcal{I}}$ either be an incident node to i' or be the arc $i' \in \tilde{\mathcal{J}}$
- 3: **if** $(k = h(i') \in \tilde{\mathcal{T}}^{j'})$ or $(k = t(i') \in \tilde{\mathcal{I}})$ or $(k = i' \in \tilde{\mathcal{J}})$ **then**
- 4: set $l(k) = l(i')$, and add k to $\mathcal{D}$

```

5: else if  $(k = t(i') \in \tilde{\mathcal{I}}^{j'})$  or  $(k = h(i') \in \tilde{\mathcal{I}})$  then
6:   set  $l(k) = \neg l(i')$ , and add  $k$  to  $\mathcal{D}$  ( $\neg$  represents the negation symbol)
7: end if
8: while  $\mathcal{D} \neq \emptyset$  do
9:   select  $i \in \mathcal{D}$ 
10:  if  $i \in \tilde{\mathcal{I}}^j$  for some  $j \in M$  then
11:    for each unlabeled node  $k \in \tilde{\mathcal{I}}^j$  and  $\bar{k} \in \tilde{\mathcal{I}}$  that is adjacent to  $i$  do
12:      set  $l(k) = l(i)$ , set  $l(\bar{k}) = \neg l(i)$ , and add  $k$  and  $\bar{k}$  to  $\mathcal{D}$ 
13:    end for
14:    for each unlabeled arc  $k \in \tilde{\mathcal{J}}$  that is incident to  $i$  do
15:      if  $i = t(k)$  then
16:        set  $l(k) = l(i)$ , and add  $k$  to  $\mathcal{D}$ 
17:      else
18:        set  $l(k) = \neg l(i)$ , and add  $k$  to  $\mathcal{D}$ 
19:      end if
20:    end for
21:  end if
22:  if  $i \in \tilde{\mathcal{I}}$  then
23:    for each unlabeled node  $k \in \tilde{\mathcal{I}}^j$  for  $j \in M$  such that  $k = i$  do
24:      set  $l(k) = l(i)$ , and add  $k$  to  $\mathcal{D}$ 
25:    end for
26:    for each unlabeled node  $k \in \tilde{\mathcal{I}}^j$  for  $j \in M$ , and  $\bar{k} \in \tilde{\mathcal{I}}$  that is adjacent to  $i$  do
27:      set  $l(k) = \neg l(i)$ , set  $l(\bar{k}) = l(i)$ , and add  $k$  and  $\bar{k}$  to  $\mathcal{D}$ 
28:    end for
29:    for each unlabeled arc  $k \in \tilde{\mathcal{J}}$  that is incident to  $i$  do
30:      if  $i = t(k)$  then
31:        set  $l(k) = \neg l(i)$ , and add  $k$  to  $\mathcal{D}$ 
32:      else
33:        set  $l(k) = l(i)$ , and add  $k$  to  $\mathcal{D}$ 
34:      end if
35:    end for
36:  end if
37:  if  $i \in \tilde{\mathcal{J}}$  then
38:    for each unlabeled node  $k = t(i) \in \tilde{\mathcal{I}}^j$  and  $\bar{k} = h(i) \in \tilde{\mathcal{I}}^j$  for  $j \in M$  do
39:      set  $l(k) = l(i)$ , set  $l(\bar{k}) = \neg l(i)$ , and add  $k$  and  $\bar{k}$  to  $\mathcal{D}$ 
40:    end for
41:    for each unlabeled node  $k = t(i) \in \tilde{\mathcal{I}}$  and  $\bar{k} = h(i) \in \tilde{\mathcal{I}}$  do
42:      set  $l(k) = \neg l(i)$ , set  $l(\bar{k}) = l(i)$ , and add  $k$  and  $\bar{k}$  to  $\mathcal{D}$ 
43:    end for
44:  end if
45:  Remove  $i$  from  $\mathcal{D}$ 
46: end while
47: for each  $i \in \tilde{\mathcal{I}}^j$  for each  $j \in M$  do
48:   add  $i^{l(i)}$  to  $\mathcal{I}_j$ 
49: end for
50: for each  $i \in \tilde{\mathcal{I}}$  do
51:   add  $i^{l(i)}$  to  $\tilde{\mathcal{I}}$ 
52: end for
53: for each  $i \in \tilde{\mathcal{J}}$  do
54:   if  $i^{l(i)} = +$  then
55:     add  $i$  to  $\mathcal{J}$ 
56:   else
57:     add  $i$  to  $\tilde{\mathcal{J}}$ 
58:   end if
59: end for

```

In view of Theorem 3, once we identify a forest structure with the desired conditions, we can use the steps in Algorithm 1 to determine the weight of each constraint in the corresponding EC&R assignment by following a path that starts from the arc associated with the base equality and reaches the node or arc associated with that constraint. We illustrate this approach in the following example.

Example 5. Consider set $\mathcal{S}$ with $m=2$ and Ξ that represents the primal network model corresponding to the graph $G = (V, A)$ shown in Figure 1. Similarly to Example 1, we refer to each arc in this network as a pair (i, j) of its tail node i and its head node j , and denote its corresponding flow variable as $x_{i,j}$. Assume that we are interested in finding EC&R assignments for class- l^+ where the base equality l contains the bilinear term $y_1 x_{1,5}$, that is, $i' = (1, 5)$ and $j' = 1$. According to Theorem 2, we need to identify a forest structure that satisfies the conditions (i)–(iii) of Propositions 4 and 6. In the parallel network G^1 , we select the forest $\bar{F}^1$ composed of the tree $\bar{T}_1^1$ with the node set $\{1, 2, 6\}$ and the tree $\bar{T}_2^1$ with the node set $\{8\}$. In the parallel network G^2 , we select the forest $\bar{F}^2$ composed of the tree $\bar{T}_1^2$ with the node set $\{1, 4\}$. Therefore, we can form the set $\tilde{\mathcal{I}}^1 = \{1, 2, 6, 8\}$ and $\tilde{\mathcal{I}}^2 = \{1, 4\}$. We select the connection node set $\tilde{\mathcal{I}} = \{3\}$, and the connection arc set $\tilde{\mathcal{J}} = \{(8, 4)\}$. It is easy to verify that these sets satisfy the conditions (i)–(iii) of Propositions 4 and 6. Next, we determine the label of each node and arc in the above sets through applying Algorithm 1. According to line 2 of this algorithm, we set $l(1, 5) = +$ in parallel network G^1 , and select $k = t(1, 5) = 1 \in \tilde{\mathcal{I}}^1$. It follows from line 5 of the algorithm that $l(1) = -$ and k is added to $\mathcal{D}$. Following lines 10–13, we obtain for $\tilde{\mathcal{I}}^1$ that $l(2) = l(6) = -$, and for $\tilde{\mathcal{I}}$ that $l(3) = +$. Then, from lines 26–28 of Algorithm 1, we deduce for $\tilde{\mathcal{I}}^2$ that $l(4) = -$, and from lines 11–13 for $\tilde{\mathcal{I}}^2$, we obtain that $l(1) = -$. Lines 32–34 imply that $l(8, 4) = -$ for $\tilde{\mathcal{J}}$. Lastly, we conclude from lines 38–40 that $l(8) = -$ for $\tilde{\mathcal{I}}^1$. As a result, following lines 47–59 of the algorithm, we obtain the EC&R assignment $[\{1^-, 2^-, 6^-, 8^-\}, \{1^-, 4^-\}, \{3^+\} | \emptyset, \{(8, 4)\}]$ for class- l^+ . Based on this assignment, we multiply the negative flow-balance constraints at nodes 1, 2, 6, 8 with y_1 , we multiply the negative flow-balance constraints at nodes 1, 4 with y_2 , we multiply the positive flow-balance constraint at node 3 with $1 - y_1 - y_2$, and we multiply the upper bound constraint on variable $x_{8,4}$ with $1 - y_1 - y_2$, and aggregate them with the base bilinear equality corresponding to arc $(1, 5)$ with weight 1 to obtain the aggregated inequality

$$\begin{aligned} & -z_{1,5} - y_1 x_{4,5} - y_1 x_{3,7} + y_1 x_{4,3} - y_2 x_{1,5} + y_2 x_{4,5} + y_2 x_{2,3} - y_2 x_{3,7} \\ & + (f_1 + f_2 + f_3 + f_6 + f_8 - u_{8,4})y_1 + (f_1 + f_3 + f_4 - u_{8,4})y_2 \\ & + x_{3,7} - x_{2,3} - x_{4,3} - x_{8,4} - f_3 + u_{8,4} \geq 0, \end{aligned}$$

where f_i denotes the supply/demand value at node i , and $u_{i,j}$ denotes the upper bound for variable $x_{i,j}$. Following the relaxation step in the EC&R procedure, we may relax each of the seven remaining bilinear terms into two possible linear expressions, leading to 128 total EC&R inequalities. If implemented inside of a separation oracle, we can use Remark 2 to find the most violated inequality among these 128 inequalities efficiently in linear time. $\square$

We conclude this section with a remark on the practical implementation of the proposed EC&R inequalities. While there is an efficient separation algorithm to find a separating EC&R inequality among those created from a given EC&R assignment as noted in Remark 2, the choice of the class of an EC&R assignment and its possible forest structure in the underlying network can lead to a large pool of candidates to consider during a branch-and-cut approach. Note that each EC&R inequality is obtained through an aggregation of the constraints of $\mathcal{S}$ with proper weights. In particular, given an EC&R assignment $[\mathcal{I}_1, \dots, \mathcal{I}_m, \tilde{\mathcal{I}} | \mathcal{J}, \tilde{\mathcal{J}}]$ for class- $l^\pm$, we aggregate the base inequality of the form $f_l(x, y, z) \geq 0$ with constraints of the general form $h(y)g(x) \geq 0$, where $h(y)$ represents the aggregation weight that could be y_j or $1 - \sum_{j \in M} y_j$, and where $g(x) \geq 0$ denotes a linear network flow constraint that could be the flow-balance or variable bound constraints. In most branch-and-cut approaches, the starting relaxation of the problem contains all linear side constraints on x and y . It follows that an optimal solution $(\bar{x}; \bar{y}; \bar{z})$ of such relaxation that is to be separated satisfies $h(\bar{y})g(\bar{x}) \geq 0$ for all valid choices of function $h(y)$ and constraint $g(x) \geq 0$. Therefore, for the resulting aggregated inequality to be violated at a point $(\bar{x}; \bar{y}; \bar{z})$, we must have the base inequality violated at that point, that is, $f_l(\bar{x}, \bar{y}, \bar{z}) < 0$. This observation can be used to select the class and sign of the EC&R assignment to be generated during a separation process. To this end, we may sort the *residual* values defined by $\Psi_k = |\bar{y}_j \bar{x}_i - \bar{z}_k|$ for all (i, j, k) such that $A_{ji}^k = 1$, and choose class k as that associated with largest Ψ_k with the class sign $+$ if $\bar{y}_j \bar{x}_i - \bar{z}_k < 0$, and class sign $-$ otherwise. This perspective can shed light on explaining the observation that the EC&R inequalities obtained from *fewer* aggregations tend to be more effective in practice as noted in Davarnia et al. [12] and also observed in our experiments in Section 4. Specifically, the addition of constraints $h(y)g(x) \geq 0$ in the aggregation can increase the left-hand-side value in the aggregated inequality when $h(\bar{y})g(\bar{x}) > 0$, which could reduce the odds of obtaining a violated aggregated inequality.

Another observation that can be helpful for choosing the forest structures is considering the relaxation step in the EC&R procedure. As described therein, each remaining bilinear term $y_j x_i$ can be relaxed using either the bound constraints or the bilinear constraints. The former case is equivalent to aggregating the inequality with a constraint of the form $h(\mathbf{y})g(\mathbf{x}) \geq 0$ where $h(\mathbf{y}) = y_j$ and $g(\mathbf{x}) \in \{x_i \geq 0, u_i - x_i \geq 0\}$, for which the previous argument holds about achieving a violation. For the latter case, on the other hand, we aggregate the inequality with a bilinear constraint of the form $\pm (y_j x_i - z_k) \geq 0$ for (i, j, k) such that $A_{j,i}^k = 1$, which can potentially lead to a violation depending on the value of $\Psi_k = |\bar{y}_j \bar{x}_i - \bar{z}_k|$. As a result, we might choose forest structures that contain the nodes incident to arcs $i \in A$ corresponding to the most violated values in $|\bar{y}_j \bar{x}_i - \bar{z}_k|$. In our computational experiments presented in Section 4, we use the above-mentioned approaches in our separation oracle to select the class of EC&R assignments and their forest structures, which show promising results.

4. Computational Experiments

In this section, we present preliminary computational results to evaluate the impact of the cutting planes generated through the results of Section 3 on improving the relaxation gap for two network flow applications that contain bilinear constraints. For these experiments, the codes are written in Python 3.8.8. and the optimization problems are solved using Gurobi 9.5.2 at its default settings.

4.1. Fixed-Charge Network Flow Problem

We study the bilinear formulation for the fixed-charge network flow problem proposed in Rebennack et al. [26]. Consider a directed bipartite network structure where the nodes in the first partition are supply nodes with supply s_i for $i \in S$, and the nodes in the second partition are demand nodes with demand d_j for $j \in D$. For each arc (i, j) , let x_{ij} represent its flow and u_{ij} denote its capacity. We assume that the cost of sending one unit flow through arc (i, j) is 0 when $x_{ij} = 0$, and it is $c_{i,j} + t_{ij}x_{ij}$ when $0 < x_{ij} \leq u_{ij}$ for some constant $c_{i,j} > 0$ and slope $t_{ij} > 0$. In Rebennack et al. [26], a piecewise concave underestimator of this cost function is considered, which calculates the cost as $c_{i,j} + \frac{t_{ij}}{\epsilon_{ij}}x_{ij}$ when $0 \leq x_{ij} \leq \epsilon_{ij}$ for some $0 \leq \epsilon_{ij} \leq u_{ij}$, and $c_{i,j} + t_{ij}x_{ij}$ when $\epsilon_{ij} \leq x_{ij} \leq u_{ij}$. Through introducing a binary variable $y_{ij} \in \{0, 1\}$ that indicates the line segment in the piecewise function above, Rebennack et al. [26] proposes the following bilinear formulation for the corresponding fixed-charge problem.

$$\min \sum_{i \in S} \sum_{j \in D} \left(\left(c_{ij} + \frac{t_{ij}}{\epsilon_{ij}} \right) x_{ij} + t_{ij} y_{ij} - \frac{t_{ij}}{\epsilon_{ij}} z_{ij} \right) \quad (11)$$

$$\sum_{j \in D} x_{ij} \leq s_i, \quad \forall i \in S \quad (12)$$

$$\sum_{i \in S} x_{ij} \geq d_j, \quad \forall j \in D \quad (13)$$

$$x_{ij} y_{ij} = z_{ij}, \quad \forall i \in S, j \in D \quad (14)$$

$$\sum_{i \in S} \sum_{j \in D} y_{ij} \leq b \quad (15)$$

$$0 \leq x_{ij} \leq u_{ij}, \quad \forall i \in S, j \in D \quad (16)$$

$$y_{ij} \in \{0, 1\}, \quad \forall i \in S, j \in D \quad (17)$$

where the objective function computes the total cost of transportation. Constraints (12) and (13) represent the flow-balance equations for the supply and demand nodes, respectively. Constraint (14) defines the bilinear terms in the objective function. Further, Constraint (15) imposes a budget requirement on the y variables. The base relaxation for this problem, which we refer to as the *McCormick relaxation*, is obtained by replacing Constraints (14) with their McCormick relaxations and relaxing y variables to be continuous. For each variable y_{ij} , the structure of the corresponding relaxation (12), (13), (14), (16) conforms to that of S with $m = 1$. Therefore, our goal is to assess the effectiveness of the EC&R inequalities associated with the tree structures in the above network using the results of Section 3.1.

For our computational experiments, we generate 10 random instances for each size category based on the following data specifications. We consider four different size categories by choosing the number of nodes in the complete bipartite network from $\{50, 100\}$ and by selecting the fraction value $\frac{\epsilon_{ij}}{u_{ij}}$ for arcs (i, j) that have fixed-charge structure (which determines the position of the break point in the piecewise linear cost function) from $\{0.2, 0.5\}$. The number of y variables is equal to 20% of the total number of arcs, where each such variable is associated with a randomly selected arc. Supply and demand parameters are chosen from a discrete uniform distribution between (20, 50). The arc capacity is produced from a discrete uniform distribution between (1, 50), and the budget b is set at 20% of the total number of y

variables. The slope t for each cost function is randomly generated from a discrete uniform distribution between (50, 100). If an arc (i, j) follows the fixed-charge structure, the value of c_{ij} is selected from a uniform distribution between (1, 5). Otherwise, this value is generated from a uniform distribution between (10, 20).

Table 1 shows the results of adding the EC&R cutting planes for the single-variable relaxations of (12)–(17) as described above. The first column contains the network size (the number of nodes), the second column shows the fraction $\frac{\epsilon_{ij}}{u_{ij}}$ of arcs (i, j) that have fixed-charge structure (which determines the position of the break point in the piecewise linear cost function), and the third column indicates the instance number. The fourth column shows the optimal value of the McCormick relaxation of (11)–(17). The fifth and sixth columns, respectively, represent the optimal value and the solution time (in seconds) of (11)–(17) obtained by the Gurobi solver at its default setting. If an optimal solution is not found within the time limit of 1,000 seconds, the best solution found at termination is reported, which is indicated by “ $\geq 1,000$ ” in the “Time” column. The percentage of the gap improvement obtained by the Gurobi solver at the root node (compared with the McCormick relaxation), before the start of branch-and-bound, is given in the seventh column. The next two columns under “Full EC&R” contain the result of adding all violated EC&R inequalities obtained

Table 1. Evaluating EC&R cutting planes for the fixed-charge network flow problem.

Node #	Frac.	#	LP	Solver			Tree EC&R				RLT	
				Opt.	Time	Root	Full	Time	Sep.	Time	Full	Time
50	0.2	1	4,437.82	5,322.36	735.83	0.58	0.72	94.42	0.66	2.19	0.72	859.62
		2	3,815.63	4,617.38	128.63	0.69	0.82	90.74	0.8	2.58	0.83	1,878.66
		3	4,769.89	5,654.0	92.98	0.57	0.77	91.15	0.72	2.44	0.77	824.01
		4	5,361.71	5,935.23	378.29	0.44	0.69	93.58	0.69	2.54	0.69	1,913.67
		5	4,424.93	5,446.63	$\geq 1,000$	0.65	0.76	104.65	0.73	2.66	0.77	2,714.41
		6	4,508.74	5,302.18	322.14	0.62	0.83	103.06	0.82	3.11	0.83	1,186.32
		7	4,693.38	5,590.65	$\geq 1,000$	0.56	0.73	72.07	0.7	1.97	0.74	2,452.11
		8	4,441.68	5,145.72	283.13	0.58	0.74	94.69	0.7	2.45	0.75	1,009.55
		9	4,233.48	5,233.15	22.76	0.84	0.93	90.82	0.88	1.81	0.93	1,116.21
		10	4,897.29	5,636.04	710.44	0.64	0.77	97.04	0.76	2.58	0.77	838.35
avg					467.42	0.62	0.78	93.22	0.75	2.44	0.78	1,479.29
50	0.5	1	4,735.93	5,347.03	254.85	0.55	0.77	90.53	0.75	2.1	0.78	197.58
		2	4,274.4	4,856.78	10.23	0.61	0.87	72.98	0.83	2.24	0.89	293.13
		3	5,026.89	5,610.48	7.87	0.75	0.86	71.24	0.84	1.93	0.88	199.79
		4	4,630.15	5,246.89	98.49	0.65	0.85	67.95	0.82	2.45	0.86	198.27
		5	4,036.38	4,669.68	106.82	0.71	0.85	91.06	0.81	1.9	0.87	236.8
		6	4,474.54	5,151.84	198.48	0.59	0.81	87.82	0.75	1.72	0.83	211.01
		7	5,176.22	5,646.79	80.32	0.63	0.82	67.16	0.8	2.41	0.83	316.09
		8	5,084.2	5,721.84	$\geq 1,000$	0.54	0.79	67.33	0.77	2.53	0.8	254.97
		9	4,523.52	5,088.3	105.66	0.63	0.79	87.54	0.77	2.64	0.8	241.79
		10	4,735.93	5,347.03	254.85	0.55	0.77	90.53	0.75	2.1	0.78	197.58
avg					186.7	0.63	0.83	79.6	0.8	2.26	0.84	238.89
100	0.2	1	7,088.91	9,213.18	$\geq 1,000$	0.36	0.51	1,162.19	0.46	8.4	—	$\geq 5,000$
		2	7,146.18	8,960.31	$\geq 1,000$	0.48	0.63	1,142.41	0.57	8.74	—	$\geq 5,000$
		3	7,248.75	9,040.84	$\geq 1,000$	0.34	0.47	882.9	0.42	9.28	—	$\geq 5,000$
		4	7,692.17	9,836.81	$\geq 1,000$	0.3	0.43	902.66	0.39	8.51	—	$\geq 5,000$
		5	6,802.61	8,709.36	$\geq 1,000$	0.4	0.55	1,144.94	0.48	8.57	—	$\geq 5,000$
		6	6,950.35	8,862.86	$\geq 1,000$	0.47	0.61	1,272.74	0.57	8.58	—	$\geq 5,000$
		7	7,052.04	8,963.63	$\geq 1,000$	0.43	0.59	1,151.14	0.52	8.37	—	$\geq 5,000$
		8	7,018.5	8,894.89	$\geq 1,000$	0.49	0.62	1,169.81	0.57	10.64	—	$\geq 5,000$
		9	6,732.16	8,723.6	$\geq 1,000$	0.41	0.53	1,132.3	0.48	8.49	—	$\geq 5,000$
		10	7,088.91	9,213.18	$\geq 1,000$	0.36	0.51	1,162.19	0.46	8.4	—	$\geq 5,000$
avg					$\geq 1,000$	0.41	0.55	1,111.73	0.49	8.87	—	$\geq 5,000$
100	0.5	1	7,277.28	8,572.76	$\geq 1,000$	0.6	0.78	1,537.78	0.7	8.52	—	$\geq 5,000$
		2	6,991.21	8,398.58	$\geq 1,000$	0.63	0.75	1,388.98	0.68	8.1	—	$\geq 5,000$
		3	7,486.13	8,604.12	$\geq 1,000$	0.52	0.72	1,331.14	0.63	7.76	—	$\geq 5,000$
		4	6,768.0	8,145.13	$\geq 1,000$	0.52	0.7	1,370.79	0.61	7.93	—	$\geq 5,000$
		5	7,018.42	8,413.21	$\geq 1,000$	0.58	0.73	1,439.65	0.67	8.33	—	$\geq 5,000$
		6	7,139.52	8,558.94	$\geq 1,000$	0.51	0.67	1,470.46	0.6	7.89	—	$\geq 5,000$
		7	7,362.48	8,822.47	$\geq 1,000$	0.51	0.65	1,519.48	0.58	8.3	—	$\geq 5,000$
		8	7,232.95	8,613.9	$\geq 1,000$	0.6	0.76	1,493.97	0.72	10.38	—	$\geq 5,000$
		9	6,881.73	8,112.89	$\geq 1,000$	0.59	0.77	1,510.83	0.72	10.47	—	$\geq 5,000$
		10	7,277.28	8,572.76	$\geq 1,000$	0.6	0.78	1,537.78	0.7	8.52	—	$\geq 5,000$
avg					$\geq 1,000$	0.57	0.73	1,445.3	0.66	8.89	—	$\geq 5,000$

from tree structures according to Theorem 2 with up to two cancellations (i.e., three aggregations). These cuts are added in loops after the LP relaxation is solved to separate the current optimal solution until the improvement in the optimal value is less than 1%. To find the most violated EC&R inequalities produced from an EC&R assignment, we use the technique in Remark 2. The column “Gap” contains the gap improvement obtained by adding these EC&R inequalities compared with the optimal value of the McCormick relaxation reported in column 6. The next column shows the total solution time to add these inequalities. The column “Gap” under “Separation EC&R” includes the result of adding the above EC&R inequalities through a separation oracle according to that discussed in Section 3. In particular, for a current optimal solution $(\bar{x}; \bar{y}; \bar{z})$, we consider the EC&R assignment class and sign associated with the 35 largest values for $\Psi_{ij} = |\bar{x}_{ij}\bar{y}_{ij} - \bar{z}_{ij}|$ for all (i, j) . We add the resulting EC&R inequalities in loops as discussed above. The next column shows the solution time when using this separation method. The last two columns under “RLT” contain the results of using the reformulation-linearization technique (RLT) of level 1 (Sherali and Adams [29], Sherali et al. [31]). Specifically, the column “Full RLT” shows the gap closure achieved by augmenting the McCormick relaxation with the linearized constraints as a result of multiplying each constraint of the formulation with y_{ij} and $(1 - y_{ij})$ for all $i \in S$ and $j \in D$. The next two columns show the gap improvement obtained by this method compared with the McCormick relaxation, and the solution time for this approach. The symbol “-” indicates that the model has not been solved to optimality within the time limit of 5,000 seconds. The last row for each problem size reports the average values over the 10 random instances.

The results in Table 1 indicate the effectiveness of the EC&R approach in five areas. First, the gap improvement values show the effectiveness of the proposed EC&R inequalities based on the tree structures in improving the gap closure and strengthening the classical McCormick relaxation. Second, the gap improvement obtained through addition of the EC&R inequalities exceeds that obtained by the Gurobi solver at the root node. Third, these results also imply that the EC&R approach is more effective in improving the dual bounds compared with the RLT application; the EC&R cuts provide similar gap improvement values in a much smaller time. Fourth, they support the general observation for aggregation-based methods, such as the EC&R, that the inequalities obtained from fewer aggregations (up to three in these experiments) tend to be the most impactful, as they account for a considerable portion of the total gap closure for most instances. This is evident from comparing the improvement achieved by the EC&R cuts with that of the RLT, which provides an upper bound for the possible gap improvement that can be obtained from single-variable disjunctive programming (Balas [2]) relaxations of the problem. Fifth, these experiments demonstrate the effectiveness of the proposed separation method, which achieves similar gap improvement values in much smaller time compared with the case without separation. These observations show promise for an efficient implementation of the EC&R technique to solve practical problems.

4.2. Transportation Problem with Conflicts

In this section, we study a generalization of the Red-Blue Transportation Problem—which is a class of the Transportation Problem with Conflicts—presented in Vancroonenburg et al. [34]. In particular, we consider a balanced set of supply and demand nodes that are arranged in a bipartite network structure, where there is a route (arc) between each supply node to each demand node. Each supply node $i \in S$ has a supply s_i , and each demand node $j \in D$ has a demand d_j . For each arc (i, j) , we refer to the flow sent through that arc as x_{ij} and to the capacity of that arc as u_{ij} . The default cost of transportation using arc (i, j) is c_{ij} . In addition to the regular transportation means, there are multiple transportation services that could be used to facilitate the transfer of goods via different routes. Binary variable y_k represents whether service $k \in K$ is used. The relative incentive/cost of using service $k \in K$ for each arc (i, j) is denoted by r_{ij}^k . In the Red-Blue Transportation Problem, due to certain logistic conflicts, some pairs of services could not be used simultaneously, that is, $y_k y_l = 0$ for each pair of services k and l that are in the conflict set C . This problem can be formulated as

$$\min \sum_{i \in S} \sum_{j \in D} \left(c_{ij} x_{ij} + \sum_{k \in K} r_{ij}^k z_{ij}^k \right) \quad (18)$$

$$\sum_{j \in D} x_{ij} \leq s_i, \quad \forall i \in S \quad (19)$$

$$\sum_{i \in S} x_{ij} \geq d_j, \quad \forall j \in D \quad (20)$$

$$x_{ij} y_k = z_{ij}^k, \quad \forall i \in S, j \in D, k \in K \quad (21)$$

$$y_k + y_l \leq 1, \quad \forall (k, l) \in C \quad (22)$$

$$0 \leq x_{ij} \leq u_{ij}, \quad \forall i \in S, j \in D \quad (23)$$

$$y_k \in \{0, 1\}, \quad \forall k \in K \quad (24)$$

where the objective function calculates the total cost of transportation. Constraints (19) and (20) represent the flow-balance equations for the supply and demand nodes, respectively. Constraint (21) defines the bilinear terms in the objective function that capture the incentive collected through using different transportation services. The conflicts between the pair of transportation services in C are imposed by Constraint (22). The base relaxation for this problem, which we refer to as the *McCormick relaxation*, is obtained by replacing Constraints (21) with their McCormick relaxations and relaxing y variables to be continuous.

For our experiments, we generate random generated instances with the following specifications. We consider four size categories by choosing the number of nodes in the complete bipartite network from $\{50, 100\}$, and selecting the number of transportation services from $\{20, 30\}$. Supply and demand parameters are generated from a discrete uniform distribution between $(100, 200)$, and the arc capacities are generated from a discrete uniform distribution between $(1, 25)$. The number of conflicts is set at 10% of all possible pairwise conflicts between the transportation services. The cost c_{ij} for each arc (i, j) is chosen randomly from a uniform distribution between $(20, 40)$ multiplied by the number of

Table 2. Evaluating EC&R cutting planes for the transportation problem with conflicts.

			Solver				EC&R							RLT	
Node #	K	#	LP	Opt.	Time	Root	Tree Full	Tree Time	Forest Full	Forest Time	Sep.	Time	Full	Time	
50	20	1	1,816,461.35	1,845,306.7	58.71	0.14	0.24	106.05	0.54	424.88	0.5	42.55	0.63	41.39	
		2	1,911,383.7	1,932,617.96	61.13	0.13	0.3	92.53	0.6	407.09	0.56	48.2	0.68	44.42	
		3	1,791,541.33	1,815,825.43	66.51	0.03	0.27	107.84	0.56	449.59	0.53	50.07	0.62	37.26	
		4	1,916,558.28	1,938,151.82	53.96	0.12	0.25	102.08	0.53	605.55	0.5	46.53	0.7	37.42	
		5	1,883,973.93	1,907,566.87	102.68	0.13	0.25	105.91	0.54	443.17	0.52	43.61	0.65	36.89	
		6	1,754,821.67	1,779,059.14	45.84	0.06	0.29	104.15	0.6	555.97	0.56	43.03	0.67	39.77	
		7	1,859,891.92	1,882,935.97	61.43	0.07	0.33	90.97	0.62	609.47	0.58	44.93	0.7	37.78	
		8	1,878,222.67	1,907,158.85	71.31	0.04	0.24	102.66	0.54	423.88	0.51	44.37	0.56	32.62	
		9	1,888,464.09	1,917,359.49	104.42	0.13	0.27	110.78	0.54	633.7	0.51	49.51	0.63	34.44	
		10	1,828,942.4	1,853,959.15	81.38	0.13	0.31	104.1	0.56	427.72	0.52	47.27	0.67	34.09	
avg					70.74	0.1	0.28	102.71	0.56	498.1	0.53	46.01	0.65	37.61	
50	30	1	2,735,130.47	2,769,326.44	416.31	0.18	0.29	173.18	0.57	1,382.33	0.56	87.34	0.69	107.66	
		2	2,758,928.32	2,797,673.91	219.19	0.19	0.24	139.81	0.55	854.01	0.52	76.51	0.67	98.79	
		3	2,790,718.79	2,830,074.49	152.08	0.16	0.22	141.19	0.56	1,179.99	0.52	76.41	0.65	98.27	
		4	2,948,401.96	2,977,573.06	75.43	0.22	0.37	148.31	0.69	1,215.35	0.66	75.41	0.85	117.27	
		5	2,848,728.26	2,884,380.56	189.93	0.16	0.27	153.52	0.58	1,010.3	0.55	74.65	0.7	93.62	
		6	2,913,904.34	2,960,520.37	510.74	0.24	0.2	177.89	0.53	1,067.57	0.51	92.51	0.63	98.26	
		7	2,600,592.36	2,643,997.19	367.13	0.16	0.26	184.94	0.57	1,055.12	0.55	83.96	0.63	113.51	
		8	2,798,975.64	2,841,262.04	324.84	0.2	0.25	176.37	0.53	1,467.63	0.49	84.44	0.7	105.82	
		9	2,909,192.25	2,948,513.71	219.4	0.2	0.28	187.26	0.59	950.87	0.56	93.77	0.72	106.4	
		10	2,611,547.24	2,656,632.4	936.96	0.16	0.23	150.52	0.54	878.4	0.52	82.67	0.62	91.72	
avg					341.2	0.19	0.26	163.3	0.57	1,106.15	0.54	82.77	0.69	103.13	
100	20	1	3,160,875.04	3,261,525.49	2,901.52	0.11	0.14	983.68	0.44	3,932.78	0.42	409.97	0.41	1,296.16	
		2	3,322,767.42	3,418,349.48	1,388.27	0.11	0.15	986.57	0.46	3,821.98	0.44	328.39	0.44	1,935.97	
		3	3,264,069.46	3,356,160.8	1,194.99	0.13	0.16	1,032.29	0.49	3,932.34	0.46	359.18	0.46	1,344.05	
		4	3,301,911.59	3,398,167.09	1,182.85	0.11	0.13	910.19	0.47	3,275.66	0.45	314.41	0.41	2,110.84	
		5	3,334,562.36	3,435,820.55	2,770.33	0.11	0.14	974.61	0.44	3,806.47	0.42	318.75	0.41	2,052.78	
		6	3,316,661.88	3,414,631.77	2,088.83	0.09	0.16	960.33	0.46	3,912.27	0.44	324.51	0.41	2,661.02	
		7	3,341,341.4	3,438,435.11	1,519.65	0.03	0.16	1,019.78	0.5	3,956.82	0.48	347.23	0.37	1,227.38	
		8	3,252,602.14	3,351,139.62	2,477.65	0.12	0.14	788.56	0.46	3,174.85	0.43	313.43	0.42	1,082.16	
		9	3,273,121.61	3,369,045.93	1,647.25	0.12	0.14	745.21	0.44	3,025.82	0.42	287.02	0.43	1,201.9	
		10	3,231,657.29	3,339,983.21	2,644.37	0.03	0.13	791.88	0.4	3,242.4	0.38	333.17	0.33	993.24	
avg					1,981.57	0.1	0.15	919.31	0.45	3,608.14	0.44	333.6	0.41	1,590.55	
100	30	1	4,834,376.78	4,968,183.0	≥5,000	0.19	0.17	1,866.58	0.55	10,207.78	0.54	1,179.07	0.51	2,243.39	
		2	4,803,153.48	4,955,801.77	≥5,000	0.19	0.12	1,642.88	0.51	11,944.64	0.49	994.29	0.46	2,236.58	
		3	4,995,871.11	5,147,567.66	≥5,000	0.13	0.14	1,838.84	0.5	11,362.07	0.5	1,091.86	0.43	2,322.86	
		4	4,961,244.87	5,143,262.24	≥5,000	0.17	0.12	1,976.34	0.46	12,195.03	0.44	1,308.95	0.42	2,062.33	
		5	4,863,008.91	5,024,866.15	≥5,000	0.17	0.13	1,833.8	0.49	11,701.45	0.48	984.07	0.45	2,238.44	
		6	4,902,278.58	5,060,625.13	≥5,000	0.17	0.13	1,498.19	0.49	8,708.22	0.49	1,099.36	0.45	1,983.21	
		7	4,866,695.51	5,017,000.38	≥5,000	0.2	0.14	1,517.02	0.52	8,596.32	0.51	1,331.52	0.49	1,886.76	
		8	5,023,334.69	5,184,894.21	≥5,000	0.22	0.13	1,437.16	0.51	9,807.23	0.5	1,086.44	0.48	1,701.46	
		9	5,043,560.92	5,198,406.71	≥5,000	0.15	0.14	1,420.03	0.49	8,983.96	0.48	1,382.81	0.45	1,784.4	
		10	4,969,879.76	5,117,213.58	≥5,000	0.18	0.14	1,379.89	0.5	8,421.21	0.49	1,189.98	0.47	1,911.2	
avg					≥5,000	0.18	0.14	1,641.07	0.5	10,192.79	0.49	1,164.84	0.46	2,037.06	

transportation services for each problem set, and the relative incentive/cost r_{ij}^k of using service k is randomly chosen from a uniform distribution between $(-10, 10)$.

Table 2 shows the results of adding the EC&R cutting planes to the McCormick relaxation described above. The first column contains the network size (the number of nodes), the second column represents the number of available transportation services (i.e., the number of y variables), and the third column indicates the instance number. The values in the next four columns are defined similarly to those of Table 1. The time limit to solve the MIP reformulation of the problem by Gurobi is set to 5,000 seconds. If the problem is not solved to optimality within the time limit, its best solution found at termination is reported, which is indicated by “ $\geq 5,000$ ” in the “Time” column. The columns under “EC&R” contain the gap improvement results obtained by adding different types of EC&R inequalities to the McCormick relaxation. In particular, “Tree Full” represents the gap improvement obtained by adding EC&R inequalities with up to three aggregations (two cancellations) generated from the one-variable relaxations of (19)–(24) where only one y variable is considered. For this approach, we use the EC&R results of Theorem 2 to identify the tree structures for each one-variable relaxation and add the resulting cutting planes for each relaxation separately through loops as previously described in Section 4.1. The next column contains the time to implement these tree inequalities. The column “Forest EC&R” includes the gap improvement obtained by adding EC&R inequalities with up to three aggregations obtained for S where two y variables in a conflict are considered in their original simplex. To add EC&R cutting planes, we consider the forest structures according to Theorem 3. The next column shows the time it takes to use this approach. The next two columns “Sep.” and “Time” indicate the gap closure and the solution time to implement these EC&R cutting planes using the separation oracle introduced in Section 3.2, respectively. The values in the columns with the header “RLT” are defined similarly to those of Table 1.

It is evident from the results of Table 2 that the EC&R cuts obtained from the forest structures that simultaneously consider multiple y variables significantly outperform those obtained from the tree structures that consider y variables individually, showing the effectiveness of the introduced class of EC&R inequalities with the pairwise cancellation property. Further, these results show the remarkable impact of using a separation oracle to produce the EC&R inequalities on reducing the solution time, especially for larger size problems where the gap improvement by the EC&R inequalities based on forest structures outperform the RLT approach in both the gap improvement and solution time.

5. Conclusion

We study a bipartite bilinear set, where the variables in one partition belong to a network flow model, and the variables in the other partition belong to a simplex. We design a convexification technique based on the aggregation of side constraints with appropriate weights, which produces an important class of facet-defining inequalities for the convex hull of the bilinear set, describing the convex hull for the special case where the simplex contains a single variable. We show that each such inequality can be obtained by considering the constraints corresponding to the nodes of the underlying network that form a special tree or forest structure. This property leads to an explicit derivation of strong inequalities through identifying special graphical structures in the network model. These inequalities can be added to the classical McCormick relaxation to strengthen the relaxation and improve the dual bounds, as corroborated in the preliminary computational experiments conducted on network problems in different application areas.

Acknowledgments

The authors thank the associate editor and the two anonymous referees for suggestions that helped improve the paper.

References

- [1] Al-Khayyal FA, Falk JE (1983) Jointly constrained biconvex programming. *Math. Oper. Res.* 8:273–286.
- [2] Balas E (1998) Disjunctive programming: Properties of the convex hull of the feasible points. *Discrete Appl. Math.* 89:3–44.
- [3] Beale EML, Forrest JHH (1976) Global optimization using special ordered sets. *Math. Programming* 10:52–69.
- [4] Ben-Ayed O, Blair CE (1990) Computational difficulties of bilevel linear programming. *Oper. Res.* 38:556–560.
- [5] Boland N, Dey SS, Kalinowski T, Molinaro M, Rigterink F (2017) Bounding the gap between the McCormick relaxation and the convex hull for bilinear functions. *Math. Programming* 162:523–532.
- [6] Bonami P, Günlük O, Linderoth J (2018) Globally solving nonconvex quadratic programming problems with box constraints via integer programming methods. *Math. Programming Comput.* 10(3):333–382.
- [7] Buchheim C, Crama Y, Rodríguez-Heck E (2019) Berge-acyclic multilinear 0-1 optimization problems. *Eur. J. Oper. Res.* 273(1):102–107.
- [8] Chiu SW (2005) Bilevel programming for the continuous transport network design problem. *Transportation Res. Part B Methodol.* 39(4):361–383.
- [9] Davarnia D (2016) Convexification techniques for bilinear and complementarity constraints with application to network interdiction. PhD thesis, University of Florida, Gainesville, FL.
- [10] Davarnia D (2021) Strong relaxations for continuous nonlinear programs based on decision diagrams. *Oper. Res. Lett.* 49(2):239–245.

- [11] Davarnia D, Van Hoeve WJ (2020) Outer approximation for integer nonlinear programs via decision diagrams. *Math. Programming* 187: 111–150.
- [12] Davarnia D, Richard JPP, Tawarmalani M (2017) Simultaneous convexification of bilinear functions over polytopes with application to network interdiction. *SIAM J. Optim.* 27:1801–1833.
- [13] Del Pia A, Di Gregorio S (2021) Chvátal rank in binary polynomial optimization. *INFORMS J. Optim.* 3(4):315–349.
- [14] Del Pia A, Khajavirad A (2018) The multilinear polytope for acyclic hypergraphs. *SIAM J. Optim.* 28(2):1049–1076.
- [15] Del Pia A, Khajavirad A (2021) The running intersection relaxation of the multilinear polytope. *Math. Oper. Res.* 46:1008–1037.
- [16] Del Pia A, Walter M (2022) Simple odd β -cycle inequalities for binary polynomial optimization. Aardal K, Sanità L, eds. *Integer Programming and Combinatorial Optimization* (Springer International Publishing, Cham, Switzerland), 181–194.
- [17] Fampa M, Lee J (2021) Convexification of bilinear forms through non-symmetric lifting. *J. Global Optim.* 80:287–305.
- [18] Ficker A, Spieksma F, Woeginger G (2021) The transportation problem with conflicts. *Ann. Oper. Res.* 298:207–227.
- [19] Gupte A, Ahmed S, Cheon MS, Dey S (2013) Solving mixed integer bilinear problems using MILP formulations. *SIAM J. Optim.* 23: 721–744.
- [20] Gupte A, Kalinowski T, Rigterink F, Waterer H (2020) Extended formulations for convex hulls of some bilinear functions. *Discrete Optim.* 36:100569.
- [21] Khajavirad A (2023) On the strength of recursive McCormick relaxations for binary polynomial optimization. *Oper. Res. Lett.* 51(2): 146–152.
- [22] Locatelli M, Schoen F (2014) On convex envelopes for bivariate functions over polytopes. *Math. Programming* 144:65–91.
- [23] Luedtke J, Namazifar M, Linderoth J (2012) Some results on the strength of relaxations of multilinear functions. *Math. Programming* 136:325–351.
- [24] McCormick GP (1976) Computability of global solutions to factorable nonconvex programs: Part I. Convex underestimating problems. *Math. Programming* 10:145–175.
- [25] Muller B, Serrano F, Gleixner A (2020) Using two-dimensional projections for stronger separation and propagation of bilinear terms. *SIAM J. Optim.* 30(2):1339–1365.
- [26] Rebennack S, Nahapetyan A, Pardalos P (2009) Bilinear modeling solution approach for fixed charge network flow problems. *Optim. Lett.* 3:347–355.
- [27] Salemi H, Davarnia D (2022) On the structure of decision diagram-representable mixed integer programs with application to unit commitment. *Oper. Res.* 71:1943–1959.
- [28] Salemi H, Davarnia D (2023) Solving unsplittable network flow problems with decision diagrams. *Transportation Sci.* 57(4):937–953.
- [29] Sherali HD, Adams WP (1990) A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.* 3:411–430.
- [30] Sherali HD, Alameddine A (1992) A new reformulation-linearization technique for bilinear programming problems. *J. Global Optim.* 2:379–410.
- [31] Sherali HD, Adams WP, Driscoll PJ (1998) Exploiting special structures in constructing a hierarchy of relaxations for 0-1 mixed integer problems. *Oper. Res.* 46:396–405.
- [32] Smith JC, Lim C (2008) Algorithms for network interdiction and fortification games. Chinchuluun A, Pardalos PM, Migdalas A, Pitsoulis L, eds. *Pareto Optimality, Game Theory And Equilibria*, vol. 17 (Springer Optimization and Its Applications, New York), 609–644.
- [33] Tawarmalani M, Richard JPP, Xiong C (2012) Explicit convex and concave envelopes through polyhedral subdivisions. *Math. Programming* 138:531–577.
- [34] Vancroonenburg W, Della Croce F, Goossens D, Spieksma FC (2014) The red-blue transportation problem. *Eur. J. Oper. Res.* 237(3): 814–823.