Enhancing the Solvability of Network Optimization Problems through Model Augmentations

Amr Nabil
ECE Dept.
Virginia Tech
Email: anabil@vt.edu

Hanif Sherali
ISE Dept.
Virginia Tech
Email: hanifs@vt.edu

Mustafa ElNainay
CSE Dept.
Alexandria University, Egypt
Email: ymustafa@alexu.edu.eg

***Abstract*—Intensive research effort has been dedicated to tackle multi-hop network problems. Joint consideration across multiple layers is required to achieve optimal performance. The general trend in solving these problems is to develop strong mathematical programming formulations that are capable of providing near-optimal solutions to practical-sized problems. For the class of problems studied, we show that a traditionally formulated model turns out to be insufficient from a problem-solving perspective. When the size of the problem increases, even state-of-the-art optimizers cannot obtain an optimal solution because of running out of memory. In this work, we show that augmenting the model with suitable additional constraints and structure enables the optimizer to derive optimal solutions, or significantly reduce the optimality gap, which were previously elusive given available memory restrictions.**

I. Introduction

Optimal performance of wireless networks requires joint consideration and optimization across multiple layers. Typically, these problems in the most complex form involve integer, binary, and continuous variables. At the network layer, rates of data sessions can be represented using continuous variables. At the MAC layer, scheduling can be done in either frequency or time domain if the available spectrum/time frame is fragmented into number of small divisions. In either case, a binary variable is needed to model the link activity between two nodes on specific frequency band or time slot. At the physical layer, adopting discrete power levels within power control strategies, and exploiting different technologies such as Multiple-Input-Multiple-Output (MIMO) and Interference Alignment (IA) mandates the use of binary and integer variables to correctly model their behavior. In most cases, this leads to an overall formulation of a Mixed-Integer-Linear Program (MILP) [1], [6], [7]. Moreover, if non-linear terms appear in the formulation, advanced optimization techniques, such as the Reformulation-Linearization Technique (RLT) [2], enable the linearization of such terms, resulting in an equivalently reformulated MILP, which is more convenient for solution using powerful, robust available software.

State-of-the-art optimizers (such as CPLEX [3]) implement a wide range of techniques and methods to solve MILPs. Most of these algorithms are based on the well-known Branch-and-Bound (B&B) method [4]. In this method, a search tree is constructed by fixing one or more binary variable to the value of zero or one. For reasonable network sizes, the number of binary variables is relatively large. Consequently, the search tree of the optimizer eventually explodes if the problem instance is sufficiently challenging so that an optimal solution is not found during the early steps. Maintaining a large search tree requires huge amount of memory, which can be beyond traditional desktop machine capabilities. In such cases, the optimizer runs out-of-memory and fails to obtain an optimal/near-optimal solution. On the other hand, most (if not all) optimizers are designed as general-purpose tools to tackle optimization problems. That is, they are not tailored to efficiently solve specifically structured problems such as the class of wireless network problems at hand. When the network size is small, any optimizer can easily provide the optimal solution for the problem under study. However, for large networks, the optimization tools are unable to handle the problem (e.g., in [5], the solver could only solve the relaxed version of the MILP problem while in [6] and [7] simulations were limited to network size of 20 and 25 nodes, respectively).

One way to tackle this problem is to use distributed procedure. Newton's method [8], among other efficient algorithms, can be adopted to solve linear and, more generally, convex optimization problems in a distributed way. However, the integrality restrictions on some variables in the MILP problem preclude a straightforward extension of this algorithm. An optimization software does not understand the networking problem itself; it recognizes the problem as an objective function with a set of variables and constraints. Although it generates different kinds of generic cuts, these cuts do not fully exploit the inherent physical structure of the problem.

This paper makes the following contributions. We demonstrate how the structure of the networking problem can be exploited to generate effective specialized cuts (constraints). The basic idea behind these cuts is to associate flows with the link activity variables based on the inherent nature of the problem. Moreover, we develop an effective strategy of introducing auxiliary binary variables to induce a specialized disjunctive constraint-based branching process. In the following sections, a case study is considered for an MILP problem and different strategies are introduced to implement this idea.

ISBN 978-3-901882-88-3

It is worth mentioning here that the proposed special cuts and strategies are also applicable to any multi-hop network problem formulated as (or reduced to) an MILP problem. The introduced cuts in Section III-C are applicable to any multi-hop network problem having minimum rate requirements on some of its data flows.

The remainder of this paper is organized as follows. Section II reviews the basic components of mathematically modeling a multi-hop network. Section III introduces our specialized techniques in details. In Section IV, we introduce a case study concerning an MILP formulation of a multi-hop network-based problem. In Section V, we present our results. Section VI concludes our work and indicates directions for future research.

II. Basic mathematical model for a multi-hop network

In this section, we review how we can mathematically model data flow balance and enforce data rate requirements of data sessions in multi-hop networks. Consider a multi-hop network where a set $\mathcal{N}$ of wireless nodes are placed randomly in bounded area. A node i is capable of directly transmit and relay signals to a subset of the surrounding nodes $\mathcal{T}_i$ in its transmission range. Also, a subset of nodes $\mathcal{I}_j$ can overhear (being interfered) by a transmit node i if they fall inside the latter's interference range. A link $(i,j) \in \mathcal{L}$ from i to j is defined if and only if $j \in \mathcal{T}_i$, where $\mathcal{L}$ is the set of all links in the network. We assume that the time frame is divided into finite number of equal sized time slots T. Each link activity at any time slot t is represented using a binary variable $x_{ij}[t]$. That is, $x_{ij}[t] = 1$ if node i transmits to node j during time slot t, and $x_{ij}[t] = 0$, otherwise. We assume a static environment where the wireless channel remains constant. Also, the interference in the network follows the protocol model [9]. In this model, a transmission is considered successful if the receive node is inside the transmit node's transmission range, and outside the interference ranges of other non-intended simultaneous transmit nodes. Therefore, any two interfering links cannot be activated simultaneously. Given these two assumptions, we can simply consider that each link (i,j) has a constant capacity C_{ij} when activated. There is a set of end-to-end sessions $\mathcal{M}$ to transfer data through the network. A session $m \in \mathcal{M}$ is defined by its source-destination pair $(s(m), d(m))$, and its data rate $r(m)$. Data flows of all sessions in the network are assumed to be steady and infinite. Without loss of generality, we also assume that each node has infinite buffer to temporarily store the relayed data traffic. Table I lists the relevant notation used in the paper. The different constraints described below model the basic behavior for a wireless multi-hop network.

Symbol	Definition
$\mathcal{N}_s$	Set of source nodes in the network
$\mathcal{N}_m$	Set of intermediate nodes in the network
$\mathcal{N}_d$	Set of destination nodes in the network
$\mathcal{N}$	$\mathcal{N}_s \bigcup \mathcal{N}_m \bigcup \mathcal{N}_d$, the set of all nodes in the network
$\mathcal{L}$	Set of all links in the network
$\mathcal{M}$	Set of all sessions in the network
$\mathcal{T}_i$	Set of nodes within the transmission range of node i
$\mathcal{I}_i$	Set of nodes within the interference range of node i
T	Total number of available time slots
$r(m)$	Data rate of session $m \in \mathcal{M}$
$s(m), d(m)$	Source and destination nodes of session $m \in \mathcal{M}$
$x_{ij}[t]$	Link activity indicator for the link (i,j) in time slot t
$f_{ij}(m)$	Data rate attributed to session $m \in \mathcal{M}$ on link (i,j)
C_{ij}	Capacity of link (i,j)

TABLE I: Notation.

Avoiding self- and mutual-interference: At any time slot t, if node i transmits signal to node j, it cannot neither transmit to nor receive from any other node. This can be expressed as follows:

$$\sum_{j \in \mathcal{T}_i} x_{ij}[t] + \sum_{k \in \mathcal{T}_i} x_{ki}[t] \leq 1, \qquad i \in \mathcal{N}, t \in \{1, 2, \cdots, T\}.$$

To avoid mutual interference, when a node j receives signal from i at a time slot t, every node $p \neq i$, where $j \in \mathcal{I}_p$ should not be transmitting in the same time. The following constraint models this behavior.

$$\sum_{i \in \mathcal{T}_j} x_{ij}[t] + \sum_{q \in \mathcal{T}_p} x_{pq}[t] \leq 1,$$
$$p : j \in \mathcal{I}_p, p \neq i, j \in \mathcal{N}, t \in \{1, 2, \cdots, T\}.$$

Maintaining network flow balance: Denote $f_{ij}(m)$ as the data rate that is attributed to session m on link (i,j). We assume that flow splitting is allowed inside the network. This means that a data flow can split or merge at any node inside the network at the bit level. Then, the flow balance at each node can be maintained using the following constraints:

$$\sum_{j \in \mathcal{T}_i}^{j \neq s(m)} f_{ij}(m) = \sum_{k \in \mathcal{T}_i}^{k \neq d(m)} f_{ki}(m),$$
$$m \in \mathcal{M}, i \in \mathcal{N}, i \neq s(m), d(m),$$

$$\sum_{j \in \mathcal{T}_i} f_{ij}(m) = r(m), \qquad m \in \mathcal{M}, i = s(m),$$

$$\sum_{j : i \in \mathcal{T}_j} f_{ji}(m) = r(m), \qquad m \in \mathcal{M}, i = d(m).$$

It is easy to show that the third constraint above can be derived from the former two. When a session is data rate requirement-restricted, $r(m)$ becomes a predetermined constant.

Link capacity: The total amount of data rate of different flows on link (i,j) cannot exceed its capacity C_{ij}. This can be represented using the following constraint:

$$\sum_{m \in \mathcal{M}}^{j \neq s(m), i \neq d(m)} f_{ij}(m) \leq \frac{1}{T} \sum_{t=1}^{T} C_{ij}.x_{ij}[t], \qquad i \in \mathcal{N}, j \in \mathcal{T}_i.$$

Objective functions: In designing multi-hop wireless networks, several objectives can be considered. Maximizing the total data flow rates of all sessions in the network is one

example. Another example is to maximize the minimum data flow rate in order to achieve fairness between sessions and avoid starvation. We assume a general utility function U to be maximized/minimized in order to express the complete problem formulation.

Traditional Formulation: A general formulation of a multi-hop wireless network can be expressed as follows:

OPT
max/min U
s.t.
Self- and mutual-interference constraints;
Network flow balance constraints;
Link capacity constraints;

In this formulation, $f_{ij}(m)$ and $r(m)$ are continuous variables, $x_{ij}[t]$ are binary variables, T and C_{ij} are constants. The problem is in the form of MILP.

III. Proposed network structure-based techniques

In this section, we derive our specialized valid inequalities (VIs) by considering the particular inherent special structures of a multi-hop network. We first examine the relationship between the sets of incoming/outgoing links associated with each node in the network. Second, we tie up the activation of links associated with each source-destination pair in the network. The last set of constraints relates the sessions with data rate requirements by generating suitable lower bounds on the number of active links associated with their source and destination nodes.

A. Relationships between nodes' incoming and outgoing links

Each physical link in the network between two nodes can be active during one or more time slots as long as this link activation does not interfere with other active links. The number of time slots during which this link needs to be active (having corresponding x-variables set at one) depends on its capacity and the amount of data flow passing through. If we consider every single node in the network, explicit constraints can be derived by simply exploiting the relationship between the number of time slots during which the incoming and outgoing links associated with this node, are active. These constraints are presented in two different ways in the following subsections.

1) VIs based on nodes' incoming and outgoing links: Let $\mathcal{N}_s$ the set of source nodes, $\mathcal{N}_m$ the set of intermediate nodes excluding any node that acts as a source or destination to any session, $\mathcal{N}_d$ the set of destination nodes. Note that the data flows are infinite. To keep the network stable, all data accumulated at any intermediate node should be forwarded within one time frame. Therefore, to simplify our analysis, we look into one time frame because the link activation schedule in other frames will be the same. Also, the time slot indices ignore the order of incoming and outgoing link activation. For example, consider a time frame length of ten time slots. If the outgoing link is activated during time slot 5, the incoming link can be activated at any other time slot other than 5 including the slots from 6 to 10. This is because what actually happens is that the data may arrive the node at time slot 7 and it will be forwarded to the next node in time slot 5 of the next frame.

For any multi-hop network, the following are implied:

1) For all intermediate nodes (excluding source and destination nodes of other sessions), if one of the incoming links is active during one of the available time slots, at least one of the outgoing links must be active during at least one of the available time slots, and vise versa. This can be expressed as follows:

$$\sum_{i:j\in\mathcal{T}_i}\sum_{t=1}^{T} x_{ij}[t] \geq 1 \Leftrightarrow \sum_{k\in\mathcal{T}_j}\sum_{t=1}^{T} x_{jk}[t] \geq 1, \quad \forall j \in \mathcal{N}_m, \quad (1)$$

and can be formulated as shown below.
For each $j \in \mathcal{N}_m$:

$$\sum_{i:j\in\mathcal{T}_i} x_{ij}[t] \leq \sum_{k\in\mathcal{T}_j}\sum_{\acute{t}=1}^{T} x_{jk}[\acute{t}], \quad \forall t \in \{1,2,\cdots,T\}, \quad (2)$$

$$\sum_{k\in\mathcal{T}_j} x_{jk}[t] \leq \sum_{i:j\in\mathcal{T}_i}\sum_{\acute{t}=1}^{T} x_{ij}[\acute{t}], \quad \forall t \in \{1,2,\cdots,T\}. \quad (3)$$

2) For all source nodes, if one of the incoming links is active during one of the available time slots, at least one of the outgoing links must be active during at least one of the available time slots, but not vice versa. In this case, the source node acts as an intermediate node for another session. This can be expressed as follows:

$$\sum_{i:s\in\mathcal{T}_i}\sum_{t=1}^{T} x_{is}[t] \geq 1 \Rightarrow \sum_{k\in\mathcal{T}_s}\sum_{t=1}^{T} x_{sk}[t] \geq 1, \quad \forall s \in \mathcal{N}_s, \quad (4)$$

which leads to the formulation given below.
For each $s \in \mathcal{N}_s$:

$$\sum_{i:s\in\mathcal{T}_i} x_{is}[t] \leq \sum_{k\in\mathcal{T}_s}\sum_{\acute{t}=1}^{T} x_{sk}[\acute{t}], \quad \forall t \in \{1,2,\cdots,T\}. \quad (5)$$

3) For all destination nodes, if one of the outgoing links is active during one of the available time slots, at least one of the incoming links must be active during at least one of the available time slots, but not vice versa. In this case, the destination node acts as an intermediate node

for another session. This can be expressed as follows:

$$\sum_{k\in\mathcal{T}_d}\sum_{t=1}^{T} x_{dk}[t] \geq 1 \Rightarrow \sum_{i:d\in\mathcal{T}_i}\sum_{t=1}^{T} x_{id}[t] \geq 1, \quad \forall d \in \mathcal{N}_d, \quad (6)$$

and can be formulated as specified below.
For each $d \in \mathcal{N}_d$:

$$\sum_{k\in\mathcal{T}_d} x_{dk}[t] \leq \sum_{i:d\in\mathcal{T}_i}\sum_{\acute{t}=1}^{T} x_{id}[\acute{t}], \quad \forall t \in \{1,2,\cdots,T\}. \quad (7)$$

These constraints, although simple and can be derived from the original constraints, are very unlikely to be automatically generated by the standard, generic methods implemented within the optimization software. However, augmenting the original formulation with these constraints may cause a slight degradation in the performance since the number of added constraints is $(|\mathcal{N}_s| + 2|\mathcal{N}_m| + |\mathcal{N}_d|) * T$, where $|\mathcal{N}_i|$ is the total number of nodes in the set $\mathcal{N}_i$. As such, when they do not offer significant reduction in the search space of the problem's feasible region, they can cause an overhead on the optimizer and negatively affect its performance. However, this occurs only in a few cases.

2) Introducing additional binary variables to facilitate special branching strategies: Although the derived formulation in Section III-A1 offers a tight model formulation, it sometimes does not provide satisfactory performance improvement. We therefore propose a new strategy of introducing certain binary variables to induce a disjunctive constraint-based branching using the developed set of cuts, which affords improved performance for some difficult instances. We motivate this strategy below and provide results in Section V to demonstrate its utility.

First, consider the set of constraints (2). By introducing additional binary variables, a revised set of constraints can be modeled as follows:

1) For each $j \in \mathcal{N}_m$:

$$\sum_{i:j\in\mathcal{T}_i} x_{ij}[t] \leq z_j, \quad \forall t \in \{1,2,\cdots,T\}, \qquad \sum_{k\in\mathcal{T}_j}\sum_{t=1}^{T} x_{jk}[t] - z_j \geq 0, \quad (8)$$

where $z_j \in \{0,1\}$. It is straightforward to check the validity of (8) by considering the cases of $z_j = 0, 1$. That is, when $z_j = 0$, these constraints reduces to $\sum_{i:j\in\mathcal{T}_i} x_{ij}[t] \leq 0, \forall t \in \{1,2,\cdots,T\}$, and $\sum_{k\in\mathcal{T}_j}\sum_{t=1}^{T} x_{jk}[t] \geq 0$. The second set of constraints becomes redundant but the first set enforces all $x_{ij}[t], \forall t \in \{1,2,\cdots,T\}$, to have the value of zero. On the other hand, when $z_j = 1$, the constraints reduce to $\sum_{i:j\in\mathcal{T}_i} x_{ij}[t] \leq 1, \forall t \in \{1,2,\cdots,T\}$, and $\sum_{k\in\mathcal{T}_j}\sum_{t=1}^{T} x_{jk}[t] \geq 1$. Here, the first set is redundant but the second set enforces that at least one of $x_{jk}[t]$ is equal to one.

Note that the addition of such superfluous binary variables to a model is atypical from a modeling perspective. However, this strategy turns out to be advantageous when done in the proposed fashion because it affords the opportunity for the solver to branch on certain key constraints (as opposed to just branching on variables as in the standard branch-and-bound/cut procedure) by virtue of the usual branching on the auxiliary binary variable. Indeed, this is evident by examining the effect of the disjunctive constraints imposed by (8) when considering the cases of z_j equal to zero and one.

Similarly, we can modify (3) as follows:
For each $j \in \mathcal{N}_m$:

$$\sum_{k\in\mathcal{T}_j} x_{jk}[t] \leq y_j, \quad \forall t \in \{1,2,\cdots,T\}, \qquad \sum_{i:j\in\mathcal{T}_i}\sum_{t=1}^{T} x_{ij}[t] - y_j \geq 0, \quad (9)$$

where $y_j \in \{0,1\}$.
Likewise, we can derive similar constraints for the source and destination nodes:

2) For each $s \in \mathcal{N}_s$:

$$\sum_{i:s\in\mathcal{T}_i} x_{is}[t] \leq z_s, \quad \forall t \in \{1,2,\cdots,T\}, \qquad \sum_{k\in\mathcal{T}_s}\sum_{t=1}^{T} x_{sk}[t] - z_s \geq 0, \quad (10)$$

where $z_s \in \{0,1\}$.

3) For each $d \in \mathcal{N}_d$:

$$\sum_{k\in\mathcal{T}_d} x_{dk}[t] \leq y_d, \quad \forall t \in \{1,2,\cdots,T\}, \qquad \sum_{i:d\in\mathcal{T}_i}\sum_{t=1}^{T} x_{id}[t] - y_d \geq 0, \quad (11)$$

where $y_d \in \{0,1\}$.

The proposed auxiliary binary variables help the optimizer improve the partitioning process in the search tree. However, if the optimization tool does not benefit from such branching opportunities due to its internal heuristics, the increased dimension of the problem might slightly negatively impact its performance. In our experience, this deterioration in performance for certain instances is outweighed by the improvement achieved for other challenging instances.

B. VIs based on links of source and destination nodes

In this section, we jointly consider the activation of links associated with each session's source-destination node pair in the network. This consideration is under two conditions. The first condition is that at least one of the outgoing links of the session's source node is active during any time slot. The

second condition is that this session's source node is not an intermediate node for any other session. Then, at least one of the incoming links to the session's destination node must be active during at least one time slot, and vice versa. Then, for each session, if these two conditions on the links associated with the source node are met, we can derive a restriction on the incoming links associated with the destination node. This can be mathematically expressed as follows:

Defining (s, d) as the source-destination pair of the session under consideration:

$$\begin{aligned}&\left\{\sum_{k\in\mathcal{T}_s}\sum_{t=1}^{T}x_{sk}[t]\geq 1\&\sum_{i:s\in\mathcal{T}_i}\sum_{t=1}^{T}x_{is}[t]\leq 0\right\}\\&\Rightarrow\left\{\sum_{i:d\in\mathcal{T}_i}\sum_{t=1}^{T}x_{id}[t]\geq 1\right\}.\end{aligned}\tag{12}$$

$$\begin{aligned}&\left\{\sum_{i:d\in\mathcal{T}_i}\sum_{t=1}^{T}x_{id}[t]\geq 1\&\sum_{k\in\mathcal{T}_d}\sum_{t=1}^{T}x_{dk}[t]\leq 0\right\}\\&\Rightarrow\left\{\sum_{k\in\mathcal{T}_s}\sum_{t=1}^{T}x_{sk}[t]\geq 1\right\}.\end{aligned}\tag{13}$$

Focusing on (12), since both expressions are linear, non-negative and integer valued, $\sum_{k\in\mathcal{T}_s}\sum_{t=1}^{T}x_{sk}[t]\leq T$, and $\sum_{i:s\in\mathcal{T}_i}\sum_{t=1}^{T}x_{is}[t]\leq T$, we get that (12) is equivalent to the following:

$$\begin{aligned}&\left\{\sum_{k\in\mathcal{T}_s}\sum_{t=1}^{T}x_{sk}[t]>0\&\sum_{i:s\in\mathcal{T}_i}\sum_{t=1}^{T}x_{is}[t]<1\right\}\\&\Rightarrow\left\{\sum_{i:d\in\mathcal{T}_i}\sum_{t=1}^{T}x_{id}[t]\geq 1\right\}.\end{aligned}$$

This in turn is equivalent to:

$$\begin{aligned}&\left\{\sum_{i:d\in\mathcal{T}_i}\sum_{t=1}^{T}x_{id}[t]\geq 1\right\}\\\text{OR}&\left\{\sum_{k\in\mathcal{T}_s}\sum_{t=1}^{T}x_{sk}[t]\leq 0\right\}\\\text{OR}&\left\{\sum_{i:s\in\mathcal{T}_i}\sum_{t=1}^{T}x_{is}[t]\geq 1\right\}\end{aligned}$$

which can be modeled as follows:

$$\begin{aligned}&h_1+h_2+h_3=1, h\in\{0,1\},\\&\sum_{i:d\in\mathcal{T}_i}\sum_{t=1}^{T}x_{id}[t]\geq h_1,\\&\sum_{k\in\mathcal{T}_s}\sum_{t=1}^{T}x_{sk}[t]\leq(1-h_2)*T,\\&\sum_{i:s\in\mathcal{T}_i}\sum_{t=1}^{T}x_{is}[t]\geq h_3.\end{aligned}\tag{14}$$

Similarly, (13) can be modeled as follows:

$$\begin{aligned}&g_1+g_2+g_3=1, g\in\{0,1\},\\&\sum_{k\in\mathcal{T}_s}\sum_{t=1}^{T}x_{sk}[t]\geq g_1,\\&\sum_{i:d\in\mathcal{T}_i}\sum_{t=1}^{T}x_{id}[t]\leq(1-g_2)*T,\\&\sum_{k\in\mathcal{T}_d}\sum_{t=1}^{T}x_{dk}[t]\geq g_3.\end{aligned}\tag{15}$$

The benefit from these constraints occurs when the source and/or destination node of a data session are not participating in other data sessions as an intermediate node. Otherwise, the added constraints may result in overhead on the overall formulation and might cause a slight degradation in the performance.

C. VIs for data rate requirement-restricted sessions

A data session is usually defined by its source-destination nodes pair in the network. We focus here on a data session with a minimum data requirement. The source node of such session can be a source of one or more other data rate requirement-restricted sessions. Similarly, its destination node can be a destination of one or more other data rate requirement-restricted sessions. As a traditional node in the multi-hop network, these source/destination nodes may relay other sessions' traffic in the network. Consequently, the amount of data transmitted from a source node is lower-bounded by the summation of the rates of sessions for which this is the source node. Similarly, the amount of data received by a destination node is lower-bounded by the summation of the rates of sessions for which this is the destination node. We can exploit this simple fact to derive special cuts as explained below.

Denote $\hat{\mathcal{M}}$ as the set of data rate requirement-restricted sessions (note that $\hat{\mathcal{M}}\bigcap\mathcal{M}=\phi$). Consequently, denote $r(\hat{m})$, $s(\hat{m})$ and $d(\hat{m})$ as the data rate, source and destination nodes of session $\hat{m}\in\hat{\mathcal{M}}$, respectively. Also, denote $f_{ij}(\hat{m})$ as the data rate that is attributed to data rate requirement-restricted session $\hat{m}$ on link (i,j). The capacity constraint for any source node $s\in\mathcal{N}_s$ is given as follows:

$$\begin{aligned}&\sum_{\hat{m}\in\hat{\mathcal{M}}}^{k\neq s(\hat{m}),s\neq d(\hat{m})}f_{sk}(\hat{m})+\sum_{m\in\mathcal{M}}^{k\neq s(m),s\neq d(m)}f_{sk}(m)\\&\leq\frac{1}{T}\sum_{t=1}^{T}C_{sk}.x_{sk}[t],\qquad(k\in\mathcal{T}_s).\end{aligned}$$

By summing both sides over $k\in\mathcal{T}_s$,

$$\begin{aligned}&\sum_{k\in\mathcal{T}_s}\left(\sum_{\hat{m}\in\hat{\mathcal{M}}}^{k\neq s(\hat{m}),s\neq d(\hat{m})}f_{sk}(\hat{m})+\sum_{m\in\mathcal{M}}^{k\neq s(m),s\neq d(m)}f_{sk}(m)\right)\\&\leq\frac{1}{T}\sum_{t=1}^{T}\sum_{k\in\mathcal{T}_s}C_{sk}.x_{sk}[t].\end{aligned}$$

As mentioned earlier, a lower bound on the LHS of the last inequality is given by the sum of $r(\hat{m})$ over all sessions $\hat{m} \in \hat{\mathcal{M}}$ for which s is the source.
Denoting this lower bound by H_s, we have:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{k\in\mathcal{T}_s} C_{sk}.x_{sk}[t] \geq H_s = \sum_{\hat{m}\in\hat{\mathcal{M}}:s=s(\hat{m})} r(\hat{m}).$$

Multiplying both sides by T and dividing both sides by $C_s^{\max} = \max\{C_{sk} : k \in \mathcal{T}_s\}$, we get:

$$\sum_{t=1}^{T}\sum_{k\in\mathcal{T}_s} \frac{C_{sk}}{C_s^{\max}}.x_{sk}[t] \geq \frac{T * H_s}{C_s^{\max}}.$$

Because $\frac{C_{sk}}{C_s^{\max}} \leq 1$, a Chvatal inequality [4] is given as follows:

$$\sum_{t=1}^{T}\sum_{k\in\mathcal{T}_s} x_{sk}[t] \geq \left\lceil \frac{T * H_s}{C_s^{\max}} \right\rceil. \quad (16)$$

Similarly, for any destination node $d \in \mathcal{N}_d$, defining $H_d = \sum_{\hat{m}\in\hat{\mathcal{M}}:d=d(\hat{m})} r(\hat{m})$, and $C_d^{\max} = \max\{C_{kd} : d \in \mathcal{T}_k\}$, we derive

$$\sum_{t=1}^{T}\sum_{k:d\in\mathcal{T}_k} x_{kd}[t] \geq \left\lceil \frac{T * H_d}{C_d^{\max}} \right\rceil. \quad (17)$$

Note that these constraints apply only to sources and destinations of data rate requirement-restricted sessions. Particularly, in cases when the number of such sessions passing through the same source/destination node increases, the proposed constraint becomes tighter and can thereby assist in enhancing performance.

IV. A CASE STUDY

In this work, we consider a cognitive radio (CR) network as a case study to evaluate the effectiveness of the proposed strategies. CR is an enabling technology for spectrum sharing in wireless networks [10]. That is, the nodes of a primary network usually do not fully utilize the available spectrum all the time. Hence, secondary CR nodes communicate by exploiting the available opportunities in time, frequency, and space domains. The prevailing paradigm is to have completely uncooperative primary and secondary networks. When the primary and secondary networks are co-located geographically, a more cooperative paradigm is to let the secondary nodes help relaying the primary nodes' traffic but not vice versa [11]. Yuan *et al.* introduced the concept of transparent coexistence of primary and secondary multi-hop networks in [12]. In that work, primary and secondary networks are completely coordinating. That is, each node in both networks may relay data from any node that belongs to the other network. The data rate requirement-restricted sessions in the context of this paper are the primary sessions. The objective in that work was to maximize the minimum rate of the secondary sessions while maintaining all data rate requirements of the primary sessions. For the details of the model and description of the constraints, see [12].

Form. Index	Description
1	OPT_Maxisum/OPT_Maximin
2	OPT_Maxisum/OPT_Maximin;2;3;5;7
3	OPT_Maxisum/OPT_Maximin;8;9;10;11
4	OPT_Maxisum/OPT_Maximin;14;15
5	OPT_Maxisum/OPT_Maximin;16;17
6	OPT_Maxisum/OPT_Maximin;2;3;5;7;14;15
7	OPT_Maxisum/OPT_Maximin;2;3;5;7;16;17
8	OPT_Maxisum/OPT_Maximin;2;3;5;7;14;15;16;17
9	OPT_Maxisum/OPT_Maximin;8;9;10;11;14;15
10	OPT_Maxisum/OPT_Maximin;8;9;10;11;16;17
11	OPT_Maxisum/OPT_Maximin;8;9;10;11;14;15;16;17

TABLE II: A summary of formulations.

V. PERFORMANCE EVALUATION

In this section, we present the performance of CPLEX (v12.6) [3] in solving the cut-enhanced optimization problem (augmented with the proposed cuts discussed in Section III) compared to its performance when solving the original problem. The set of test cases consists of 55 randomly generated instances (combinations of Maximin and Maxisum[1] versions of the original problem), with 11 instances each of 30, 35, 40, 45, and 50-node networks. Each network has four active sessions: two primary and two secondary sessions where the source and destination of each were randomly selected. We used a cluster at VirginiaTech, called BlueRidge [13], to run our experiments. Each experiment was executed on a single node of BlueRidge that has 16 processors (utilized by CPLEX when possible) and 64GB memory. This hardware configuration is very similar to a traditional desktop machine so that any practitioner can use the proposed algorithms to run similar experiments without the need of state-of-the-art cluster capabilities. Each experiment is terminated when its run-time reaches 144 hours (this limitation comes from the rules enforced by the BlueRidge administration), reaches optimal solution, or runs out of memory, whatever happens first. For the sake of clarity, Table II summarizes all problem formulations tested in our experiments. As shown in the table, each formulation represents either one of the two versions of the original problem or one of the versions augmented with one or more of the proposed sets of cuts described in Section III.

A. Recognizing hard instances

We define "hard" instances as the ones that CPLEX could not solve to optimality within the enforced computational limits. Consequently, these instances were run using formulations augmented with different combinations of the proposed cuts to test their relative effectiveness and performance improvement. In order to distinguish the hard instances from others in the test set, each of the instance's statistics (number of binary variables, number of constraints, etc.) was

[1]The Maxisum version is very similar to the Maximin version except that the objective function in the former is to maximize the sum of the flow rates of secondary sessions instead of maximizing the minimum flow rate as in the latter.

correlated with the level of the instance's hardness. However, we could not derive a clear relationship using these statistics. On the other hand, we noticed that CPLEX significantly reduces the number of binary variables for some instances during the preprocessing step. We found a high correlation between the "reduced" number of binary variables and the difficulty of the instance. That is, if the resulting reduced number of binary variables is above a certain threshold (2000 in the problem under consideration), CPLEX could not solve it to optimality for almost all cases because it runs out of memory. As a result, this serves as a good test for deciding whether an instance should be augmented with one or more of the proposed techniques, or not (before attempting to solve it). In the following sections, we will focus only on the instances that CPLEX could not solve to optimality when implementing the original formulation.

B. Potential of the proposed formulations

Augmenting the original problem with only one set of cuts in Section III-B or III-C (Formulations 4 and 5) did not result in considerable improvement over the original Formulation 1. Consequently, all the results presented below will focus on the comparison between the performance of different representations of the logical implications in Sections III-A1 or III-A2 augmented by one or more of the proposed sets of cuts in Sections III-B and III-C. Due to space limitations, we show detailed results for a few key formulations followed by comparative results for all formulations. We define the optimality gap for any maximization problem as follows:

$$\text{Optimality gap} = \frac{\text{UB-LB}}{\text{LB}} * 100\%$$

where LB (lower bound) is determined by calculating the objective value of the best obtained solution, i.e., the incumbent solution, and UB is the value of the LP-relaxation, which is an "upper bound" for the optimal solution of the MILP problem. We consider improvement/degradation in the performance if the optimality gap is decreased/increased by at least 5%, respectively.

C. Detailed results

We discuss here the effect of introducing additional binary variables to the constraints in Section III-A1 (see Section III-A2). Table III shows some of the instances of different network sizes where the auxiliary binary variables significantly enhanced the performance. Formulation 2 helped CPLEX significantly reduces the optimality gap (Instances III-1, III-3, III-4, and III-5) or even reach the optimal solution (Instance III-2). When introducing binary variables to the added cuts (Formulation 3), CPLEX attained optimality for most instances (and further reduced the optimality gap for others). More interestingly, for Instance III-2, CPLEX obtained the optimal solution in 82.65 hours using Formulation 2. With Formulation 3, it attained optimality much quicker (7.8 hours). As discussed in Section III-A2, this improved performance when using the extra binary variables is likely because of the different type of branching opportunities that are afforded by these extra variables. We also show the significance of adding the cuts in Section III-B to Formulations 3 (Formulation 9). As shown in Table IV, the added cuts caused the optimality gap to further shrink and/or reach the optimal solution in shorter time. Similar results were obtained when the cuts in Section III-C were used (Formulation 10). Note that longer running time does not always mean worse performance. It can also mean that CPLEX has more opportunity (before running out of memory) to further reduce or close the optimality gap.

Instance	Formulation 1		Formulation 2		Formulation 3	
	opt. gap	time (h)	opt. gap	time (h)	opt. gap	time (h)
III-1	118.03%	6.41	101.77%	12.5	optimal	2.63
III-2	156.86%	3.39	optimal	82.65	optimal	7.8
III-3	26.54%	12.03	13.67%	16.96	optimal	33.86
III-4	62.63%	5.70	40.93%	13.23	21.07%	24.63
III-5	290.18%	24.35	127.68%	39.64	95.1%	40.1

TABLE III: Effect of augmenting the original formulation with the VIs w/ and w/o superfluous binary variables.

Instance	Formulation 1		Formulation 3		Formulation 9	
	opt. gap	time (h)	opt. gap	time (h)	opt. gap	time (h)
IV-1	77.98%	9.08	optimal	19.56	optimal	5.89
IV-2	98.79%	7.57	70.8%	29.02	50.4%	23.34
IV-3	25.76%	9.55	16.62%	17.98	optimal	6.00
IV-4	19.84%	11.12	optimal	47.74	optimal	34.93
IV-5	23.19%	11.28	12.96%	26.50	8.19%	20.36

TABLE IV: CPLEX's performance with Formulations 3 and 9.

D. Comparative results

Table V summarizes statistics for the overall performance of different formulations. Here, "p1", "p2" and "p3" in the table refer to the logical implications in Sections III-A, III-B and III-C, respectively. The "+ve effect"/"-ve effect" columns represent the percentage of instances for which the optimality gap was enhanced/degraded, respectively. The "no effect" column shows the percentage of instances where there was no noticeable effect on the optimality gap. Table VI compares the positive effect performance of the proposed formulations for different network sizes (represented by node count). Figure 1 shows CPLEX's behavior under Formulation 11 for all instances. From these results, we can deduce the following:

- Overall, for all the proposed formulations, we obtained significant improvement between $50-65\%$ of the instances, slight degradation for about 20%, and no noticeable effect on the remaining set of instances.
- The proposed formulations were most effective for networks of size between 30-45 nodes. When the number of nodes is less than 30, the problem size is small enough so that every instance can be solved to optimality using the original formulation. When the number of nodes exceeds 45, the increased problem difficulty suggests that further model or algorithmic enhancements are needed

Form. Index	Description			+ve effect	-ve effect	no effect
	p1	p2	p3			
2	VIs			65.45%	21.82%	12.73%
6		✓		60%	23.64%	16.36%
7			✓	61.82%	21.82%	16.36%
8		✓	✓	60%	21.82%	18.18%
3	VIs w/ binary vars			58.18%	12.73%	29.09%
9		✓		60%	21.82%	18.18%
10			✓	60%	20%	20%
11		✓	✓	50.91%	18.18%	30.91%

TABLE V: Overall performance of the formulations.

Form. Index	Node count				
	30	35	40	45	50
2	45.45%	90.91%	72.73%	72.73%	45.45%
6	72.73%	54.55%	81.82%	45.45%	45.45%
7	63.64%	90.91%	81.82%	45.45%	27.27%
8	45.45%	63.64%	72.73%	72.73%	45.45%
3	72.73%	63.64%	63.64%	54.55%	36.36%
9	45.45%	72.73%	63.64%	63.64%	54.55%
10	54.55%	54.55%	72.73%	45.45%	72.73%
11	63.64%	54.55%	72.73%	36.36%	27.27%

TABLE VI: Percentage of instances exhibiting enhanced performance.

in this case. For networks of this size, the branch-and-bound tree gets relatively huge when approaching small values of optimality gap despite the tightening effect of the proposed cuts.

In general, adding cuts helps by tightening relaxations, but also influences branching strategies and the performance of the solver's internal heuristics, which can have unpredictable effects if the augmented model representation is not aligned with the software's built-in algorithmic strategies. Likewise, introducing auxiliary binary variables to the formulation does enhance the performance by providing alternative constraint-based branching options, but adds some difficulty to the overall problem and causes "improvement degradation" in a few instances (where the imposed extra variables burden the formulation more that they assist it). However, overall, the proposed combination of strategies provide a significant impetus to resolving challenging problem instances that were otherwise hopelessly impossible to solve.

VI. Conclusions and Future Work

In this paper, we discussed different approaches to tackle the problem of excessive memory consumption when solving MILP problems. Generic formulations are often not sufficiently attractive from the problem-solving perspective. We demonstrated that generating special cuts through exploiting the structure of the problem offers a better strategy. In most cases, combining different kinds of special cuts outperformed the performance of the formulations that use these cuts individually (or not at all). Moreover, introducing auxiliary binary variables to provide partitioning opportunities based on these cuts, when applicable, significantly enhanced the performance for some instances. Overall, this work demonstrates how the use of proper combinations of model enhancement techniques can help optimize (or further reduce the optimality gap) for challenging instances that were hopelessly unsolvable using traditional formulations.

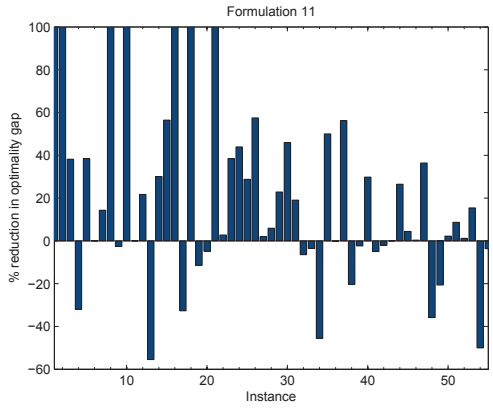

Fig. 1: CPLEX behavior under Formulation 11.

This work can be extended in several directions. The general multi-hop network cuts can be applied to different problems to study their relative effect. On the other hand, additional approaches following a like philosophy can be explored to obtain better performance. For example, we could specify specialized branching priorities within CPLEX for binary and integer variables, or introduce partitioning based on different types of disjunctive constraints according to our understanding of the network structure.

References

[1] Y. T. Hou, Y. Shi, and H. D. Sherali, *Applied Optimization Methods for Wireless Networks,* Cambridge University Press, Cambridge, UK, 2014.

[2] H. D. Sherali and W. P. Adams, "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems," *SIAM J. Discrete Math.,* vol. 3, no. 3, pp. 411–430, Aug. 1990.

[3] http://www.ilog.com/products/cplex/

[4] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization,* John Wiley & Sons, New York, NY, 1999.

[5] Y. T. Hou, Y. Shi, and H. D. Sherali, "Spectrum sharing for multi-hop networking with cognitive radios," *IEEE J. Sel. Areas Commun.,* vol. 26, no. 1, pp. 146–155, Jan. 2008.

[6] M. Johansson and L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation," *IEEE Trans. Wireless Commun.,* vol. 5, no. 2, pp. 435–445, Feb. 2006.

[7] M. Pan, C. Zhang, P. Li and Y. Fang, "Joint routing and link scheduling for cognitive radio networks under uncertain spectrum supply," in *Proc. IEEE INFOCOM,* Shanghai, China, pp. 2237–2245, April 2011.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization,* Cambridge University Press, Cambridge, UK, 2004.

[9] P. Gupta and P. R. Kumar, "The capacity of wireless networks." *IEEE Trans. Inf. Theory,* vol. 46, no. 2, pp. 388–404, March 2000.

[10] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Elsevier Computer Networks,* vol. 50, no. 13, pp. 2127–2159, Sept. 2006.

[11] O. Simone, I. Stanojev, S. Savazzi, Y. Bar-Ness, U. Spagnolini, and R. Pickholtz, "Spectrum leasing to cooperating secondary ad hoc networks," *IEEE J. Sel. Areas Commun.,* vol. 26, no. 1, pp. 203–213, Jan. 2008.

[12] X. Yuan, Y. Shi, Y. T. Hou, W. Lou, and S. Kompella, "UPS: A united cooperative paradigm for primary and secondary networks," in *Proc. IEEE MASS,* Hangzhou, Zhejiang, P.R.C., pp. 78–85, Oct. 2013.

[13] http://www.arc.vt.edu/resources/hpc/blueridge.php