

Relaxed Connected Dominating Set Problem for Power System Cyber-Physical Security

Kin Cheong Sou and Jie Lu

Abstract—Motivated by power system cyber-physical security applications, this paper investigates a variant of connected dominating set (CDS) problem called relaxed connected dominating set (RCDS) problem, whereby between every two nodes in the solution dominating set there exists a path not traversing two consecutive nodes not in the set. The RCDS problem is NP-hard. To facilitate solving, presolving and problem reduction techniques are developed. In addition, three integer program formulations are obtained using single commodity flow, Dantzig-Fulkerson-Johnson and Miller-Tucker-Zemlin ideas to enforce graph connectedness. For a realistic power network repository (MATPOWER 7) including graphs with 70k nodes, our presolving and reduction procedures simplify problem instances to on average 30% of the original sizes. All reduced instances, when modeled by our formulations, can be solved with optimality gap less than 0.5% within 20 minutes using a PC with Gurobi 9.

I. INTRODUCTION

A. Problem description

Given a connected undirected graph $G = (V, E)$, we define $D \subseteq V$ to be a *relaxed connected dominating set* (RCDS) if the following two conditions (RCDS-D1) hold:

- 1) D is a dominating set of G , meaning that every node in V is either in D or a neighbor of a member of D ;
- 2) For any $i, j \in D$, a path exists in G that cannot traverse two consecutive nodes not in D .

In sequel, we refer to the paths in 2) of definition (RCDS-D1) as D -paths associated with D . For example, $1, 2, 4, 6$ is a D -path in the left of Fig. 1 with RCDS $D = \{1, 2, 6\}$. In contrast, a connected dominating set is $\{1, 2, 4, 6\}$ as shown in the right.

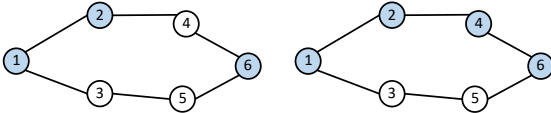


Fig. 1: Left: an example RCDS. Right: an example CDS.

Given $G = (V, E)$, $\mathcal{B} \subseteq V$ and $\mathcal{W} \subseteq V$, we consider the (minimum) RCDS optimization problem defined as

$$\begin{aligned} & \underset{D \subseteq V}{\text{minimize}} && |D| \\ & \text{subject to} && D \text{ is a RCDS of } G \\ & && \mathcal{B} \subseteq D \\ & && \mathcal{W} \cap D = \emptyset \end{aligned} \quad (1)$$

where \mathcal{B} and \mathcal{W} specify the subsets of V that must be included and excluded in the solution set D , respectively.

K.C. Sou is with the Department of Electrical Engineering, National Sun Yat-sen University, Taiwan. J. Lu is with the School of Information Science and Technology, ShanghaiTech University, China. E-mail: soul2@mail.nsysu.edu.tw, lujie@shanghaitech.edu.cn

Problem (1) is motivated from power system cyber-physical security applications. The connection, however, is better seen with an equivalent definition of RCDS as follows: For a graph $G = (V, E)$ and $U \subseteq V$, we denote $E_G[U] = \{\{i, j\} \in E \mid \{i, j\} \cap U \neq \emptyset\}$ being the set of edges “incident to” U . We may omit the subscript G in $E_G[\cdot]$ when appropriate. Then, the following condition, denoted (RCDS-D2), is equivalent to (RCDS-D1):

Proposition 1: Given a connected undirected graph $G = (V, E)$ with $|V| \geq 2$, a set $D \subseteq V$ is a RCDS of G if and only if graph $(V, E[D])$ is connected.

Proof: In the proof we abbreviate (RCDS-D1) and (RCDS-D2) as (D1) and (D2), respectively. First, we establish that (D2) \implies (D1). The assumptions that $|V| \geq 2$ and $(V, E[D])$ is connected by (D2) imply that every $i \in V$ is incident to some edge in $E[D]$ denoted, for instance, by $\{i, j\}$. The definition of $E[D]$ requires that at least one of i and j is in D . This concludes that D is a dominating set, satisfying condition 1) in (D1). In addition, (D2) specifies that for any pair of nodes $i, j \in D \subseteq V$, there exists a path $i = i_0, i_1, \dots, i_p = j$ such that $\{i_k, i_{k+1}\} \in E[D]$ and at least one of $i_k, i_{k+1} \in D$, for all $0 \leq k < p$. Therefore, the path cannot traverse two consecutive nodes not in D and hence condition 2) in (D1) is satisfied. Next, we establish (D1) \implies (D2). Consider any two nodes $i, j \in V$. By condition 1) in (D1), only four possibilities are allowed: (a) $i \in D, j \in D$; (b) $i \in D, j \notin D$ but adjacent to some $t \in D$; (c) $j \in D, i \notin D$ but adjacent to some $s \in D$; (d) $i \notin D, j \notin D$ but i is adjacent to $s \in D$ and j is adjacent to $t \in D$. Consider case (d), by condition 2) in (D1) a D -path s, \dots, t exists traversing no two consecutive nodes not in D . Hence, the combined path i, s, \dots, t, j exists traversing no two consecutive nodes not in D and hence every edge on the path is in $E[D]$. This establishes (D2) for case (d). The other three cases (a), (b) and (c) can be analyzed similarly. ■

In sequel, we consider only graphs with at least two nodes and hence problem (1) can be interpreted with (RCDS-D1) or (RCDS-D2) depending on the context. Given an instance of dominating set problem specified by graph $G = (V, E)$, we can construct an instance of (1) specified by graph $G_0 := (V \cup \{v_0\}, E \cup \{\{v_0, i\} \mid i \in V\})$, $\mathcal{B} = \emptyset$ and $\mathcal{W} = \{v_0\}$ such that $D \subseteq V$ is a dominating set of G if and only if D is a RCDS of G_0 with $v_0 \notin D$ (according to (RCDS-D1)). Since the decision version of dominating set problem is NP-complete [1], the RCDS problem in (1) is NP-hard. In fact, even if G is planar and $\mathcal{B} = \mathcal{W} = \emptyset$, (1) is still NP-hard. The proof is omitted.

B. Application motivation

Cyber-physical security has evolved into an important topic in power system research since the initiation of this direction (e.g., [2]–[4]). The analysis and evaluation of data integrity

of state estimation (with SCADA and PMU measurements) is a topic of great concern (e.g., [5]–[9]). To counter cyber-physical security threat, attack detection and SCADA system strengthening are investigated (e.g., [10]–[14]). Strengthening SCADA system typically involves the strategic deployment of limited security resources (e.g., secure PMU, secure communication, etc.) in order to maximize the security and resilience of the system. This is the main concern of this paper and we study the *perfect protection problem* formulated in [4] seeking to eliminate the threat of false data injection attacks on state estimation [3] by protecting the minimum number of remote terminal units (RTU). In this paper, a power system is modeled as an undirected graph where a node represents a bus and an edge models a line, transformer, etc. In addition to assuming that each bus is equipped with a RTU subjected to protection decision as in [4], this paper also assumes that certain pre-specified buses are equipped with PMUs that we need to decide whether to protect or not. More details about the application setup can be found in [15]. Based on [5], [16], [17], perfect protection can be shown to be equivalent to the condition that a certain subgraph of a modified power system graph is connected. The modified graph contains one additional node denoted the “ground” and it has an edge connecting to each node with PMU in the original graph. The subgraph required to be connected for perfect protection has the same node set of the modified graph, while edges incident to nodes with protected RTUs or PMUs form the edge set. Thus, the set of protected buses for perfect protection is a RCDS according to (RCDS-D2), with G interpreted as the modified graph and the ground must be in \mathcal{W} (since the ground does not have any RTU or PMU to protect). Therefore, the perfect protection problem is the same as (1) with appropriately defined G and \mathcal{W} . The set \mathcal{B} can describe the RTUs and PMUs that must be protected or free to protect. However, we note that (1) cannot model the situation where different RTUs or PMUs have different costs to protect.

C. Contributions

The RCDS problem in (1) has a dual interpretation of equipment (e.g., RTU, PMU, FACTS, etc.) placement problem to guarantee certain desirable properties (e.g., observability). This is also an important problem (e.g., [18]–[21]). However, rather than focusing on problem modeling (e.g., a contingency-constrained extension as in [19]) or suboptimal solution procedures (e.g., [20], [21]), this paper investigates the structure and properties of the underlying graph combinatorial optimization problem. Our intention is to obtain close-to-optimal solutions with meaningful optimality gap even for very large problem instances, as we may afford significantly more resources in memory and time (e.g., hours, days, etc.) in power system planning than in real-time operations commonly considered.

To the best of our knowledge, the combinatorial RCDS problem in (1) has not appeared in the literature before our preliminary conference work in [15]. In this paper, we propose

- a collection of presolving (or data reduction, kernelization [22], etc.) techniques to simplify problem (1), to be detailed in Sections II and III;

- a condensation method, leading to an equivalent reduction of (1) by coalescing connected components of $(V, E_G[D])$ guaranteed by presolving, to be detailed in Section II;
- integer program formulations of the reduced problem based on ideas including single commodity flow (SCF), Dantzig-Fulkerson-Johnson (DFJ) and Miller-Tucker-Zemlin (MTZ), to be detailed in Section IV.

Presolving is a standard technique to simply integer program instances to facilitate its solution process. For example, if 0-1 integer variables x and y satisfy $x + y \leq 0.7$ then we can fix $x = y = 0$. Presolving is effective practically (e.g., [23]–[25]) and theoretically (e.g., [26]–[30]). Solvers including Gurobi and CPLEX employ elaborate presolving routines. Most relevant to this paper are the presolving results for dominating set problem in [26] and for planar connected dominating set problem in [30]. However, it should be emphasized that “relaxed connectedness” for general (not necessarily planar) graphs poses additional challenges demanding further investigations beyond the previous work.

Most previous presolving ideas for domination-like problems are “local” in that the neighborhood(s) of one or two nodes are considered at any time. However, for a “global” property such as graph connectedness (cf. (RCDS-D2)) global analysis may yield deeper problem reduction. In particular, presolving restricts the choices of D in (1) to include certain nodes denoted by D^1 , implying that $E[D^1]$ always contribute to satisfy condition (RCDS-D2) in (1). If $(V, E[D^1])$ is connected then (1) is solved by presolving alone. This does not hold in general, but the information of the connected components of $(V, E[D^1])$ can help to greatly simplify (1) as will be demonstrated subsequently. This global presolving technique is another distinguishing feature of this paper.

While graph connectedness and the related tree requirements (spanning tree, Steiner tree, etc.) often appear in the analysis of (radial) power distribution systems, its modeling appears to be somewhat ad-hoc or specialized in power system literature [31]–[34]. In an earlier work in [17] we proposed using standardized constraints such as SCF and MTZ formulations, both requiring only linearly many constraints with respect to problem instance size. In this paper, we also study the DFJ formulation with exponentially many constraints requiring “cutting-plane-like” problem modeling (e.g., using lazy constraints in Gurobi). The effect of this formulation will be demonstrated later in this paper.

The focus of this paper is the practical performance of our approach applied to power system applications. For this, we evaluate our approach with all applicable benchmarks in MATPOWER 7 [35], including the 70k-node ACTIVSg70k example. For all applicable cases, our proposed approach is able to obtain highly accurate solutions ($< 0.5\%$ optimality gap) in reasonable time (< 20 minutes). To the best of our knowledge, demonstration on this scale for integer program solving in power systems applications has not been common.

II. PROPOSED SOLUTION APPROACH

Since problem (1) is NP-hard, it is unrealistic to expect an efficient polynomial-time algorithm for all cases. With (RCDS-D2), it is possible to model (1) as an integer linear program

with appropriate graph connectedness constraints (e.g., SCF, DFJ, MTZ formulations). However, for larger instances this approach is insufficient even with state-of-the-art solvers (e.g., Gurobi 9). Instead, in this paper we propose to first simplify (1) by a series of *equivalent presolving updates* – including extra constraints in (1) to eliminate parts of feasible (or even optimal) solutions, while guaranteeing that the updated problem has an optimal solution which can be used to construct an optimal solution to (1). Only after presolving is branch-and-bound (e.g., Gurobi) called to solve the remaining problem.

A. Presolving

For the rest of this paper, the symbol $G = (V, E)$ is reserved for the input graph to (1). The effect of presolving is encoded by the triple $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ where \mathcal{L} is a node-labeling function $\mathcal{L} : V \mapsto \{-2, -1, 0, 1, 2, 3\}$, \mathcal{P} is a set of unordered pairs of V (i.e., $\mathcal{P} \subseteq V^2$). We initialize \mathcal{G} as G but the effect of subsequent presolving is reflected in \mathcal{G} . The meaning of $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ is as follows: every node $u \in V$ is designated a “label” $\mathcal{L}(u)$ so that

- $\mathcal{L}(u) = -2$ means u is in the optimal solution set of (1) but it can be “merged” with some neighboring node v in the solution set. To reflect the graph structure change, \mathcal{G} is updated as follows: u and all its incident edges are removed from \mathcal{G} . Also, for every $w \in N(u) \setminus \{v\}$ we update $E(\mathcal{G}) \leftarrow E(\mathcal{G}) \cup \{w, v\}$, where $N(u)$ denotes the open neighborhood of u . Node u disappears subsequently until in the end it is included in the final solution;
- $\mathcal{L}(u) = -1$ means $u \in V$ is excluded from the solution of (1) and u and its incident edges are removed from \mathcal{G} ;
- $\mathcal{L}(u) = 0$ means $u \in V$ is excluded from the solution of (1) but it should remain in \mathcal{G} for certain reasons (e.g., u needs to be dominated, u is needed to construct D - paths);
- $\mathcal{L}(u) = 1$ means $u \in V$ should be in the solution of (1);
- $\mathcal{L}(u) = 2$ means u is none of above and there is some $v \in V$ such that at least one of u and v is included in the solution set of (1). The set \mathcal{P} contains all such pairs;
- $\mathcal{L}(u) = 3$ means the status of $u \in V$ is none of above.

In sequel, for any $U \subseteq V$ we use the shorthand $\mathcal{L}(U) = x$ to express $\mathcal{L}(v) = x$ for all $v \in U$. Note also that since nodes with negative \mathcal{L} values are removed, $V(\mathcal{G}) = \{v \in V \mid \mathcal{L}(v) \geq 0\}$. By convention, we assume that if $\mathcal{L}(u) \leq 1$ for any $u \in V$ then $\{u, v\} \notin \mathcal{P}$ for any $v \in V$. For instance, if $\{u, v\} \in \mathcal{P}$ originally but then $\mathcal{L}(u) \leftarrow 0$, then we also update $\mathcal{L}(v) \leftarrow 1$ and $\mathcal{P} \leftarrow \mathcal{P} \setminus \{u, v\}$. Similarly, suppose $\{u, v\} \in \mathcal{P}$. If $\mathcal{L}(u) \leftarrow 1$ or -2 , then we update $\mathcal{P} \leftarrow \mathcal{P} \setminus \{u, v\}$ and update $\mathcal{L}(v)$ to 3 depending on whether v pairs only with u .

Assuming the meaning of $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ is justified (detail in Section III), problem (1) can be updated to the *annotated RCDS problem* as follows:

$$\begin{aligned}
& \underset{D \subseteq V(\mathcal{G})}{\text{minimize}} && |D| \\
& \text{subject to} && D \text{ is a RCDS of } \mathcal{G} \\
& && \mathcal{L}(u) = 0 \implies u \notin D \\
& && \mathcal{L}(u) = 1 \implies u \in D \\
& && \{u, v\} \in \mathcal{P} \implies \text{at least one of } u, v \text{ in } D
\end{aligned} \tag{2}$$

The original RCDS problem in (1) described by G, \mathcal{B} and \mathcal{W} is a case of (2) with $\mathcal{L}(\mathcal{B}) = 1$, $\mathcal{L}(\mathcal{W}) = 0$, $\mathcal{L}(V \setminus (\mathcal{B} \cup \mathcal{W})) = 3$, $\mathcal{P} = \emptyset$ and $\mathcal{G} = G$. Starting from the initial triple $(\mathcal{L}, \mathcal{P}, \mathcal{G})$, we sequentially apply the presolving procedures to be discussed in Section III to update $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ to a more manageable but equivalent instance of (2). For the examples in MATPOWER 7 the proposed presolving alone can typically decide the solution set membership of over 70% of nodes.

B. Condensation

Problem (2) can be further simplified by replacing the RCDS condition (over graph \mathcal{G}) with the connectedness of a much smaller graph. Let $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ specify a presolving update. Define $E^1 = E^1(\mathcal{L}, \mathcal{P}, \mathcal{G}) \subseteq E(\mathcal{G})$ by

$$E^1 := E_{\mathcal{G}}[\mathcal{L}^{-1}(1)] \cup \{\{u, v\} \in E(\mathcal{G}) \mid \{u, v\} \in \mathcal{P}\} \tag{3}$$

where $\mathcal{L}^{-1}(1) = \{v \in V(\mathcal{G}) \mid \mathcal{L}(v) = 1\}$. In addition, define

$$\mathcal{D}(\mathcal{L}, \mathcal{P}) := \{D \subseteq V(\mathcal{G}) \mid D \text{ satisfies all but the first (i.e., RCDS) constraint in (2)}\} \tag{4}$$

Then, any D feasible to (2) is in \mathcal{D} and hence $E^1 \subseteq E[D]$. In the “lucky” special case where $(V(\mathcal{G}), E^1)$ is connected, then by (RCDS-D2) problem (2) is reduced to its relaxation

$$\begin{aligned}
& \underset{D \subseteq V(\mathcal{G})}{\text{minimize}} && |D| \\
& \text{subject to} && D \in \mathcal{D}(\mathcal{L}, \mathcal{P})
\end{aligned} \tag{5}$$

(5) is easier to solve than (2) because of the absence of the RCDS constraint. Suppose D^* is optimal to (5). Then $D^* \cup \mathcal{L}^{-1}(-2)$ is optimal to (2). However, in general $(V(\mathcal{G}), E^1)$ is not connected and solving (5) does not necessarily solves (2). In this case, we need to enlarge D so that edges in $E[D] \setminus E^1$ connect the connected components of $(V(\mathcal{G}), E^1)$ to make $(V(\mathcal{G}), E[D])$ connected (so that D is a RCDS of \mathcal{G}). To describe this remaining problem we consider a derived graph called *condensation* defined as follows: Let H_1, H_2, \dots, H_K denote the connected components of $(V(\mathcal{G}), E^1)$. The condensation is denoted H and its node set is $V(H) = \{H_1, H_2, \dots, H_K\}$. To define the edge set $E(H)$, we need the following notations: $\bar{E}^2 := E(\mathcal{G}) \setminus \cup_i E(H_i)$ and $E^2 := \bar{E}^2 \setminus \{\{u, v\} \in \bar{E}^2 \mid \mathcal{L}(u) = \mathcal{L}(v) = 0\}$. Then, $E(H) = \{\{H_u, H_v\} \mid \exists \{u, v\} \in E^2, u \in V(H_u), v \in V(H_v)\}$. For $D \subseteq V(\mathcal{G})$, we denote

$$\mathcal{E}_H(D) := \{\{H_u, H_v\} \in E(H) \mid \exists \{u, v\} \in E^2 \cap E_{\mathcal{G}}[D], u \in V(H_u), v \in V(H_v)\} \tag{6}$$

as the set of edges in H “enabled” by D . The following statement provides the desired characterization of RCDS of \mathcal{G} in (2) in the context of condensation H .

Proposition 2: Let $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ be given and H be defined by E^1 in (3). Then, for any $D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ in (4), D is a RCDS of \mathcal{G} if and only if the graph $(V(H), \mathcal{E}_H(D))$ is connected.

Proof: Suppose D is a RCDS of \mathcal{G} . For any two $H_u, H_v \in V(H)$, let $u \in V(H_u)$ and $v \in V(H_v)$. Then, by (RCDS-D2) there is a path with edges in $E[D]$ between u and v . Suppose the path traverses $H_u = H_{i_1}, H_{i_2}, \dots, H_{i_p} = H_v$. Then, for $1 \leq k < p$, $E[D]$ includes an edge $\{i_k, i_{k+1}\}$ such that $i_k \in V(H_{i_k})$ and

$i_{k+1} \in V(H_{i_{k+1}})$. Since i_k and i_{k+1} belongs to different components H_{i_k} and $H_{i_{k+1}}$, $\{i_k, i_{k+1}\} \in \bar{E}^2$. Further, it is impossible that $\mathcal{L}(i_k) = \mathcal{L}(i_{k+1}) = 0$ since otherwise $\{i_k, i_{k+1}\} \notin E[D]$ as $D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$. Hence, $\{i_k, i_{k+1}\} \in E^2 \cap E[D]$ and consequently $\{H_{i_k}, H_{i_{k+1}}\} \in \mathcal{E}_H(D)$. The aforementioned argument holds for all $1 \leq k < p$, and hence a path exists in H between H_i and H_j . This concludes that $(V(H), \mathcal{E}_H(D))$ is connected.

Conversely, suppose $D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ and $(V(H), \mathcal{E}_H(D))$ is connected. Let $u, v \in V(\mathcal{G})$ and let $u \in V(H_u)$, $v \in V(H_v)$. There exists a path $H_u = H_{i_1}, H_{i_2}, \dots, H_{i_p} = H_v$ in $(V(H), \mathcal{E}_H(D))$. For $1 \leq k < p$, by definition of $\mathcal{E}_H(D)$ there exist $i_k \in V(H_{i_k})$ and $i_{k+1} \in V(H_{i_{k+1}})$ such that $\{i_k, i_{k+1}\} \in E[D]$. In addition, $D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ implies that, for any $1 \leq k \leq p$, H_{i_k} is indeed a connected component of $(V(\mathcal{G}), E^1)$ and hence a path with edges in $E[D]$ exists between any two nodes $s, t \in V(H_{i_k})$. Therefore, a path exists between any $u, v \in V(\mathcal{G})$ with edges in $E[D]$. This implies that $(V(\mathcal{G}), E[D])$ is connected and hence D is a RCDS of \mathcal{G} by definition (RCDS-D2). ■

According to Proposition 2, the annotated RCDS problem in (2) can be rewritten as the *condensed RCDS problem* defined on condensation H as follows:

$$\begin{aligned}
& \underset{D \subseteq V(\mathcal{G})}{\text{minimize}} && |D| \\
& \text{subject to} && (V(H), \mathcal{E}_H(D)) \text{ is connected} \\
& && \mathcal{L}(u) = 0 \implies u \notin D \\
& && \mathcal{L}(u) = 1 \implies u \in D \\
& && \{u, v\} \in \mathcal{P} \implies \text{at least one of } u, v \text{ in } D
\end{aligned} \tag{7}$$

The main advantage of (7) is the dimension reduction due to H . For instance, for the original graph of the ACTIVSg70k benchmark $|V| = 70000$ and $|E| = 83318$. With the proposed presolving and condensation, $|V(H)| = 12592$ and $|E(H)| = 19361$. This is approximately reduction of 82% nodes and 77% in edges, respectively. This can significantly speed up solution computation and will be demonstrated in Section V.

C. Summary of proposed solution procedure

In Section III, a collection of presolve procedures will be discussed. Some are local requiring only $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ to compute, while others are global in that condensation H (as a function of $(\mathcal{L}, \mathcal{P}, \mathcal{G})$) needs to be computed before the presolving can be determined. Starting from the initial $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ of the original problem in (1) which is assumed to be feasible, the presolving procedures can be applied (after finding the corresponding H if necessary) in any order with or without repetition to update $(\mathcal{L}, \mathcal{P}, \mathcal{G})$. After presolving, the final version of H is once again computed and the corresponding problem (7) is modeled as an integer program and solved using a branch-and-bound type algorithm by calling Gurobi.

III. EQUIVALENCE-PRESERVING PRESOLVING

Given a triple $(\mathcal{L}, \mathcal{P}, \mathcal{G})$, presolving updates it to $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$ so that problem (2) (resp., (7)) remains equivalent with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ and $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$. It is reminded that equivalence means that the updated problem has an optimal solution which can be used to easily construct an optimal solution to the problem before the update. The following

describes the proposed presolving procedures and shows their correctness. In the following we implicitly assume that if some $u \in V(\mathcal{G})$ has its label $\mathcal{L}(u)$ updated to -2 or -1 then u and its incident edges will be removed from \mathcal{G} , conforming to the convention of $\mathcal{L}(\cdot)$ discussed in Section II. The following preparatory statement will be useful in this section, justifying when we can “strengthen” the \mathcal{L} -values of certain nodes in \mathcal{G} from 0 to -1 and hence removing them from \mathcal{G} .

Proposition 3: Let $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ be given to define (2). Let $\mathcal{D}(\mathcal{L}, \mathcal{P})$ be defined in (4). Suppose $U \subseteq V(\mathcal{G})$ satisfies all three below:

- 1) $\mathcal{L}(U) = 0$;
- 2) For every $D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$, for all $u \in U$ there exists some $v \in D$ such that $u \in N[v]$, where $N[v]$ denotes the closed neighborhood of v (in \mathcal{G}). Intuitively, U is dominated in \mathcal{G} by every $D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$;
- 3) For every $D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$, every D -path in \mathcal{G} between $s, t \in D$ can be replaced by another D -path in \mathcal{G} between s and t not traversing any node in U .

Define $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$ by letting $\mathcal{L}'(v) = \mathcal{L}(v)$ for all $v \in V \setminus U$ and $\mathcal{L}'(U) = -1$. In addition, $\mathcal{G}' = \mathcal{G} - U$ (i.e., \mathcal{G} with U and $E[U]$ removed) and $\mathcal{P}' = \mathcal{P}$. Then, D is feasible to (2) with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ if and only if D is feasible to (2) with $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$.

Proof: Suppose D is feasible to (2) with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$. Then $D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ and hence $U \cap D = \emptyset$ by proposition assumption 1). This implies that $D \subseteq V(\mathcal{G} - U)$ (a necessary condition to become a feasible solution of (2) with $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$). In addition, feasibility of D with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ implies that $u \notin D$ for all $\mathcal{L}'(u) = 0$ since $\mathcal{L}'^{-1}(0) \subseteq \mathcal{L}^{-1}(0)$, meaning that D satisfies the second constraint of (2) with $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$. Further, D satisfies the third and fourth constraints with $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$ by the definitions of \mathcal{L}' and \mathcal{P}' . The rest of this part argues for the first (RCDS) constraint of (2). D being a feasible solution with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ implies that D is a RCDS of \mathcal{G} . By (RCDS-D1) D is a dominating set of \mathcal{G} . Together with the fact that $U \cap D = \emptyset$, it holds that for all $w \in V(\mathcal{G}) \setminus U$, either (a) $w \in D$ or (b) there exists some $v \in D$ such that $v \notin U$ and $\{v, w\} \in E(\mathcal{G})$. Conditions (a) and (b) hold even if \mathcal{G} is replaced with $\mathcal{G} - U$. In particular, (b) remains true since $v, w \notin U$ and hence $\{v, w\} \in E(\mathcal{G} - U)$. This implies that D is also a dominating set of $\mathcal{G} - U$, establishing the first condition that D is a RCDS of $\mathcal{G} - U$ by (RCDS-D1). Next, consider any $s, t \in D$. By (RCDS-D1) there exists a D -path between s and t in \mathcal{G} . By proposition assumption 3), $D \in \mathcal{D}(\mathcal{G}, \mathcal{P})$ implies that the D -path can be replaced by a replacement D -path not traversing any node in U and hence is in $\mathcal{G} - U$. Therefore, D is a RCDS of $\mathcal{G} - U$. This concludes that D is feasible to (2) with $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$.

Now suppose D' is feasible to (2) with $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$. Then $D' \subseteq V(\mathcal{G} - U) \subseteq V(\mathcal{G})$. In addition, $D' \cap \mathcal{L}'^{-1}(0) = \emptyset$ due to the second constraint in (2). Then, $D' \cap \mathcal{L}^{-1}(0) = D' \cap (\mathcal{L}'^{-1}(0) \cup U) = (D' \cap \mathcal{L}'^{-1}(0)) \cup (D' \cap U) = \emptyset$ since $D' \subseteq V(\mathcal{G} - U)$ implies $D' \cap U = \emptyset$. Hence, D' satisfies the second constraint of (2) with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$. Namely, $\mathcal{L}(u) = 0$ implies $u \notin D'$. Also, D' satisfies the third and fourth constraints in (2) with $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$ because of the definitions of \mathcal{L}' and \mathcal{P}' . This establishes that $D' \in \mathcal{D}(\mathcal{L}, \mathcal{P})$. To conclude that D' is

feasible to (2) with $(\mathcal{L}', \mathcal{P}', \mathcal{G}')$, it remains to show D' is a RCDS of \mathcal{G} . D' is a RCDS of $\mathcal{G} - U$ and this implies for all $w \in V(\mathcal{G} - U) = V(\mathcal{G}) \setminus U$ there exists some $v \in V(\mathcal{G}) \setminus U$ such that $v \in D'$ and $w \in N[v]$ (in $\mathcal{G} - U$ and hence in \mathcal{G} as well). Since $D' \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ (as D' satisfies all but the first constraint in (2) with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$), proposition assumption 2) implies that for all $w \in U$ there exists some $v \in D'$ such that $w \in N[v]$ (in \mathcal{G}). This implies that D' is a dominating set of \mathcal{G} (the first condition of RCDS-D1). Finally, note that D' being a RCDS of $\mathcal{G} - U$ implies that for any $s, t \in D$ there exists in D -path in $\mathcal{G} - U \subseteq \mathcal{G}$ between s, t . This establishes that D' is a RCDS of \mathcal{G} . Hence D' is feasible to (2) with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$. ■

The remaining part of this section lists the proposed pre-solving procedures and argue for their correctness. It can be shown that the runtime of these procedures are bounded by $O(d(G)^2|V(G)|^2)$ where $d(G)$ is the maximum degree of G . The detail of the complexity analysis is omitted.

A. Degree-one node in \mathcal{G}

Prerequisite: $\deg(u) = 1$ with $v \in N(u)$, $\mathcal{L}(u) = 3$.

Rule:

- 1) If $\mathcal{L}(v) = 0$ then update $\mathcal{L}(u) \leftarrow 1$.
- 2) If $\mathcal{L}(v) > 0$, then update $\mathcal{L}(u) \leftarrow -1$ and $\mathcal{L}(v) \leftarrow 1$.

Proof: Any $D \subseteq V(\mathcal{G})$ feasible to (2) is a RCDS of \mathcal{G} . By (RCDS-D1), u must be dominated by D . This implies that $D \cap \{u, v\} \neq \emptyset$ since $N[u] = \{u, v\}$. This implies 1). For 2), we note that if D is feasible to (2) and $u \in D$ then $D' := D \cup \{v\} \setminus \{u\}$ is also feasible to (2) with $|D'| = |D|$. This is because $E[D] \subseteq E[D']$ (hence D' is also a RCDS of \mathcal{G} by (RCDS-D2)) and the other three constraints in (2) are not violated by D' . This implies $\mathcal{L}(u) \leftarrow 0$ as it is always possible to replace D with D' as optimal solution to (2). In turn, since u needs to be dominated and $N[u] = \{u, v\}$, update $\mathcal{L}(v) \leftarrow 1$ is required. Consider $\mathcal{D}(\mathcal{L}, \mathcal{P})$ as defined in (4) with the updated \mathcal{L} and \mathcal{P} . Assumptions 1) and 2) of Proposition 3 are satisfied with $U = \{u\}$ because of \mathcal{L} . Let $s, t \in D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ and consider any D -path between s and t in \mathcal{G} . The D -path cannot traverse u since $\mathcal{L}(u) = 0$ and $\deg(u) = 1$. Hence, assumption 3) of Proposition 3 is also satisfied. Thus, we can update $\mathcal{L}(u) \leftarrow -1$ according to Proposition 3.

B. Degree-two-triangle

Prerequisite: $\deg(u) = 2$ with $N(u) = \{v, w\}$, v and w are adjacent (i.e., u, v, w form a triangle in \mathcal{G}), $\mathcal{L}(u) = 3$.

Rule:

- 1) If $\mathcal{L}(v) = \mathcal{L}(w) = 0$, then update $\mathcal{L}(u) \leftarrow 1$.
- 2) If $\mathcal{L}(v) > 0$ but $\mathcal{L}(w) = 0$, set $\mathcal{L}(v) \leftarrow 1$, $\mathcal{L}(u) \leftarrow -1$. Update analogously if $\mathcal{L}(w) > 0$ but $\mathcal{L}(v) = 0$ instead.
- 3) If $\mathcal{L}(v) > 0$, $\mathcal{L}(w) = 3$ or 0 , $\deg(v) > \deg(w) = 2$, then update $\mathcal{L}(v) \leftarrow 1$, $\mathcal{L}(w) \leftarrow -1$ and $\mathcal{L}(u) \leftarrow -1$. Update analogously if v and w switch roles instead.
- 4) If $\mathcal{L}(v) > 0$, $\mathcal{L}(w) > 0$, $\deg(v) > 2$ and $\deg(w) > 2$.
 - If $\mathcal{L}(v) = 1$ or $\mathcal{L}(w) = 1$, then update $\mathcal{L}(u) \leftarrow -1$.
 - If $\mathcal{L}(v) > 1$, $\mathcal{L}(w) > 1$, set $\mathcal{L}(u) \leftarrow -1$, add $\{v, w\}$ to \mathcal{P} and change $\mathcal{L}(v)$ and $\mathcal{L}(w)$ to 2 if necessary.

Proof: All cases can be understood intuitively by ‘‘combining’’ v and w to become the only neighbor of u . The detail argument is as follows: 1) is due to the need to dominate u . For 2), we consider only the case where $\mathcal{L}(w) = 0$ and omit the symmetric case where $\mathcal{L}(v) = 0$. If D containing u is feasible to (2) then it is claimed that $D' := D \setminus \{u\} \cup \{v\}$ is also feasible to (2) with $|D'| = |D|$. On one hand $\mathcal{L}(v) > 0$ and $\mathcal{L}(u) = 3$ imply that D' satisfies all but the first constraint in (2). On the other hand, by (RCDS-D2) D' is a RCDS of \mathcal{G} since the triangle structure involving u, v, w implies that $(V(\mathcal{G}), E[D'])$ is connected if and only if $(V(\mathcal{G}), E[D])$ is connected (the latter is since D is feasible to (2)). Thus, D' is feasible to (2) and this together with the triangle structure imply that $\mathcal{L}(u) \leftarrow 0$ and $\mathcal{L}(v) \leftarrow 1$. Next, let $\mathcal{D}(\mathcal{L}, \mathcal{P})$ be defined in (4) with the updated \mathcal{L} and \mathcal{P} . Assumptions 1) and 2) in Proposition 3 hold with $U = \{u\}$ because of \mathcal{L} and the triangle structure. For $s, t \in D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ consider any D -path between s and t traversing U . The segment of the D -path v, u, w can be replaced with v, w since $\mathcal{L}(v) = 1$, and hence a replacement D -path not traversing U is always possible. Therefore, assumption 3) of Proposition 3 holds and it can be concluded that $\mathcal{L}(u) \leftarrow -1$. 3) holds because u and w (resp., u and v) can be interpreted as a single node with degree one. For 4) a similar argument as 2) suffices and the detail is omitted.

C. Degree-two-path

Prerequisite: $\deg(u) = 2$ with $N(u) = \{v, w\}$ but v, w are not adjacent (v, u, w form a path in \mathcal{G}), $\mathcal{L}(u) = 3$.

Rule:

- 1) If $\mathcal{L}(v) = 1$ and $\mathcal{L}(w) > 0$, update $\mathcal{L}(u) \leftarrow 0$. Similarly, if $\mathcal{L}(w) = 1$ and $\mathcal{L}(v) > 0$, update $\mathcal{L}(u) \leftarrow 0$.
- 2) If $\{v, w\} \in \mathcal{P}$, then $\mathcal{L}(u) \leftarrow 0$.

Proof: For both 1) and 2), we consider the case with D feasible to (2) such that $v \in D$ (the case where $w \in D$ is analyzed similarly). Under the assumption that $v \in D$, if D containing u is feasible to (2) then $D' := D \setminus \{u\} \cup \{w\}$ is also feasible to (2) with $|D'| = |D|$ since $E[D] \subseteq E[D']$. This implies $\mathcal{L}(u) \leftarrow 0$. However, unlike degree-one case $\mathcal{L}(u) \leftarrow -1$ cannot be imposed. Removing u from \mathcal{G} may disconnect \mathcal{G} .

D. Surrounded by label-1 neighbors

Prerequisite: $\mathcal{L}(u) = 3$, $\deg(u) > 1$.

Rule:

- 1) If $\mathcal{L}(N(u)) = 1$, ‘‘merge’’ $N[u]$ into a single node as follows: identify (arbitrarily) $v \in N(u)$ and update $\mathcal{L}(v) \leftarrow 1$. For any $w \in N(u) \setminus \{v\}$ and $s \in N(w) \setminus N[u]$ such that $\{w, s\} \in E(\mathcal{G})$, we update $E(\mathcal{G}) \leftarrow E(\mathcal{G}) \cup \{s, v\}$ (combine multi-edges if necessary). In addition, set $\mathcal{L}(u) \leftarrow -1$ and $\mathcal{L}(w) = -2$ for all $w \in N(u) \setminus \{v\}$.
- 2) If some $v \in N(u)$ satisfies $\mathcal{L}(v) > 1$ and for the rest of $N(u)$, it holds that $\mathcal{L}(N(u) \setminus \{v\}) = 1$, then $\mathcal{L}(u) \leftarrow 0$.

Proof: For 1), note that $E[N[u]] \subseteq E[D]$ for all D feasible to (2) whether $u \in D$ or not. Since $\mathcal{L}(u) = 3 \neq 2$, it is not restrictive to update $\mathcal{L}(u) \leftarrow 0$. To argue the rest of 1), let \mathcal{G}' denote the graph after merging $N[u]$. In addition, let \mathcal{L}' be the same as \mathcal{L} except $\mathcal{L}'(u) = -1$ and $\mathcal{L}'(N(u) \setminus \{v\}) =$

–2. We need to show that $D' \subseteq V(\mathcal{G}')$ is feasible to (2) with $(\mathcal{L}', \mathcal{P}, \mathcal{G}')$ if and only if $D = D' \cup (N(u) \setminus \{v\})$ is feasible to (2) with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$. First, note that $\mathcal{L}^{-1}(1) = \mathcal{L}'^{-1}(1) \cup (N(u) \setminus \{v\})$. Then,

$$\begin{aligned} & \mathcal{L}^{-1}(1) \subseteq D \\ \iff & (\mathcal{L}'^{-1}(1) \cup (N(u) \setminus \{v\})) \subseteq (D' \cup (N(u) \setminus \{v\})) \\ \iff & \mathcal{L}'^{-1}(1) \subseteq D' \end{aligned}$$

since $\mathcal{L}'^{-1}(1) \cap (N(u) \setminus \{v\}) = \emptyset$ and $D' \cap (N(u) \setminus \{v\}) = \emptyset$. Second, note that $\mathcal{L}^{-1}(0) = \mathcal{L}'^{-1}(0) \cup \{u\}$. Then,

$$\begin{aligned} & \mathcal{L}^{-1}(0) \cap D \\ \iff & (\mathcal{L}'^{-1}(0) \cup \{u\}) \cap (D' \cup (N(u) \setminus \{v\})) \\ \iff & \mathcal{L}'^{-1}(0) \cup D' \end{aligned}$$

Hence, with \mathcal{D} defined in (4) with $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ and \mathcal{D}' analogously defined in (4) with $(\mathcal{L}', \mathcal{P}, \mathcal{G}')$, it holds that $D' \in \mathcal{D}'$ if and only if $D = D' \cup (N(u) \setminus \{v\}) \in \mathcal{D}$. To complete the proof of 1), it remains to show that $D' \in \mathcal{D}'$ is a RCDS of \mathcal{G}' if and only if $D = D' \cup (N(u) \setminus \{v\})$ is a RCDS of \mathcal{G} . This condition, according to (RCDS-D2), is equivalent to

$$(V(\mathcal{G}'), E_{\mathcal{G}'}[D']) \text{ connected} \iff (V(\mathcal{G}), E_{\mathcal{G}}[D]) \text{ connected} \quad (8)$$

Since $\mathcal{L}'(v) = 1$, $D' \in \mathcal{D}'$ implies that $v \in D'$. In addition, $N(u) \subseteq D = D' \cup (N(u) \setminus \{v\})$ as $v \in D'$. Hence,

$$E_{\mathcal{G}'}[\{v\}] \subseteq E_{\mathcal{G}'}[D'] \quad \text{and} \quad E_{\mathcal{G}}[N(u)] \subseteq E_{\mathcal{G}}[D] \quad (9)$$

In addition, it can be verified that

$$E_{\mathcal{G}'}[D'] \cap E_{\mathcal{G}'}(\mathcal{G}' - \{v\}) = E_{\mathcal{G}}[D] \cap E_{\mathcal{G}}(\mathcal{G} - N[u]) \quad (10)$$

Then, (9) and (10) together imply (8). This concludes the argument for 1). For 2), if $D \subseteq V(\mathcal{G})$ is feasible to (2) then $D' := D \setminus \{u\} \cup \{v\}$ is also feasible to (2). Hence, $\mathcal{L}(u) \leftarrow 0$. Note that u may not be removed from graph (i.e., $\mathcal{L}(u) \neq -1$) as its removal may disconnect graph \mathcal{G} .

E. Surrounded by label-0 neighbors

Prerequisite: $\mathcal{L}(u) \geq 2$, $\deg(u) > 1$.

Rule: If $\mathcal{L}(N(u)) = 0$. Then update $\mathcal{L}(u) \leftarrow 1$.

Proof: The need to dominate u requires $\mathcal{L}(u) \leftarrow 1$.

F. Bridge in \mathcal{G}

For connected graph \mathcal{G} , an edge $e \in E(\mathcal{G})$ is a bridge (or edge-cut) if $(V(\mathcal{G}), E(\mathcal{G}) \setminus \{e\})$ is not connected. The bridges of \mathcal{G} can be identified in $O(|V(\mathcal{G})| + |E(\mathcal{G})|)$ time (e.g., [36]).

Prerequisite: $\{v, w\} \in E(\mathcal{G})$ is a bridge.

Rule:

- 1) If $\mathcal{L}(v) = 0$ then set $\mathcal{L}(w) \leftarrow 1$. Conversely, if $\mathcal{L}(w) = 0$ then set $\mathcal{L}(v) \leftarrow 1$.
- 2) If $\mathcal{L}(v) \geq 2$ and $\mathcal{L}(w) \geq 2$ and $\{v, w\} \notin \mathcal{P}$, set $\mathcal{P} \leftarrow \mathcal{P} \cup \{v, w\}$ and set $\mathcal{L}(v) \leftarrow 2$, $\mathcal{L}(w) \leftarrow 2$ if necessary.

Proof: For any $D \subseteq V(\mathcal{G})$ to be a RCDS, $\{v, w\}$ must be in $E[D]$ for $(V(\mathcal{G}), E[D])$ to be connected (RCDS-D2). Therefore, at least one of v and w is in the optimal solution of (2). Situations of 1) and 2) call for the corresponding updates.

G. Singleton component of $(V(\mathcal{G}), E^1)$

Recall $E^1 \subseteq E(\mathcal{G})$ defined in (3) in Section II.

Prerequisite: $u \in V(\mathcal{G})$ forming a singleton connected component of $(V(\mathcal{G}), E^1)$, with $\mathcal{L}(u) \geq 2$.

Rule:

- 1) If $\mathcal{L}(N(u)) = 0$, then update $\mathcal{L}(u) \leftarrow 1$
- 2) If $\mathcal{L}(u) = 3$ and $N(u) \subseteq V(H_k)$ for some connected component H_k of $(V(\mathcal{G}), E^1)$ and at least one $w \in N(u)$ satisfies $\mathcal{L}(w) \geq 1$, then update $\mathcal{L}(u) \leftarrow 0$.
- 3) Same as 2) except $\mathcal{L}(u) = 2$ with only one z such that $\{u, z\} \in \mathcal{P}$, then update $\mathcal{L}(u) \leftarrow 0$.

Proof: 1) is necessary because u needs to be dominated. For 2), since $N(u) \subseteq V(H_k)$, the only possible benefit to include u in the solution of (2) is to dominate u . However, including w can also dominate u . Therefore, $\mathcal{L}(u) \leftarrow 0$. However, $\mathcal{L}(u)$ cannot be updated to -1 , since u has not been guaranteed dominated by any node in $\mathcal{L}^{-1}(1)$ of pair v and w such that $\{v, w\} \in \mathcal{P}$. 3) is similar except that $\mathcal{L}(u) \leftarrow 0$ is not guaranteed to be profitable if u pairs with more than one node.

H. Dominated enabler

Recall H being the condensation defined by $(\mathcal{L}, \mathcal{P}, \mathcal{G})$ (through E^1) in Section II. Also recall for $D \subseteq V(\mathcal{G})$, $\mathcal{E}_H(D)$ defined in (6) is the set of edges in $E(H)$ enabled by D .

Rule: If $L(u) = 3$, $L(v) > 1$ and $\mathcal{E}_H(\{u\}) \subseteq \mathcal{E}_H(\{v\})$ then update $\mathcal{L}(u) \leftarrow 0$.

Proof: If u is to be included in the optimal solution of (7) then replacing u with v is always no-worse.

I. Bridge in condensation H

Prerequisite: $\{H_k, H_\ell\} \in E(H)$ is a bridge of H .

Rule: If $\{H_k, H_\ell\} \in \mathcal{E}_H(\{u\})$ and all other w such that $\{H_k, H_\ell\} \in \mathcal{E}_H(\{w\})$ have label 0 (i.e., $\mathcal{L}(w) = 0$). Then update $\mathcal{L}(u) \leftarrow 1$.

Proof: For any D feasible to (7), $\{H_k, H_\ell\}$ must be in $\mathcal{E}_H(D)$. Since u is the only choice to enable $\{H_k, H_\ell\}$, we must update $\mathcal{L}(u) \leftarrow 1$.

J. Neighborhood of a single node in \mathcal{G}

This presolving procedure is modified from the one in [26] for dominating set problem. However, the different definition of RCDS and the -1 label require the modifications and the proof herein. For any $v \in V(\mathcal{G})$, let

$$\begin{aligned} N_1(v) &:= \{u \in N(v) \mid N(u) \setminus N[v] \neq \emptyset, \\ &\quad \text{or } \mathcal{L}(u) = 1, \text{ or } \{u, w\} \in \mathcal{P}, \text{ for } w \neq v\} \\ N_2(v) &:= \{u \in N(v) \setminus N_1(v) \mid N(u) \cap N_1(v) \neq \emptyset\} \\ \tilde{N}_2(v) &:= \{u \in N_2(v) \mid \mathcal{L}(N(u) \cap N_1(v)) = 0\} \\ N_3(v) &:= N(v) \setminus (N_1(v) \cup N_2(v)) \end{aligned} \quad (11)$$

Prerequisite: $v \in V(\mathcal{G})$ and $\mathcal{L}(v) = 3$.

Rule: If $N_3(v) \cup \tilde{N}_2(v) \neq \emptyset$, then $\mathcal{L}(N_2(v) \cup N_3(v)) \leftarrow -1$ and $\mathcal{L}(v) \leftarrow 1$.

Proof: See Appendix A.

K. Neighborhood of a pair of nodes in \mathcal{G}

This presolving procedure is modified from the one in [26] for dominating set problem. However, the different definition of RCDS and the -1 label require the modifications and the proof herein. For any pair $v, w \in V(\mathcal{G})$, let $N(v, w) := N(v) \cup N(w) \setminus \{v, w\}$ and $N[v, w] := N[v] \cup N[w]$. In addition, denote

$$\begin{aligned} N_1(v, w) &:= \{u \in N(v, w) \mid N(u) \setminus N[v, w] \neq \emptyset\} \\ &\quad \text{or } \mathcal{L}(u) = 1, \text{ or } \{u, z\} \in \mathcal{P}, \text{ for } z \notin \{v, w\} \\ N_2(v, w) &:= \{u \in N(v, w) \setminus N_1(v, w) \mid N(u) \cap N_1(v, w) \neq \emptyset\} \\ \tilde{N}_2(v, w) &:= \{u \in N_2(v, w) \mid \mathcal{L}(N(u) \cap N_1(v, w)) = 0\} \\ N_3(v, w) &:= N(v, w) \setminus (N_1(v, w) \cup N_2(v, w)) \end{aligned} \quad (12)$$

Prerequisite: For any $v, w \in V(\mathcal{G})$, $v \neq w$ such that

- $\mathcal{L}(v) = \mathcal{L}(w) = 3$
- $N[v] \cap N[w] \neq \emptyset$ (i.e., $\{v, w\} \notin E(\mathcal{G}) \implies N(v) \cap N(w) \neq \emptyset$)
- $\tilde{N}_2(v, w) \cup N_3(v, w) \neq \emptyset$
- $\nexists z \in N_2(v, w) \cup N_3(v, w)$ such that $\mathcal{L}(z) = 3$ and $(\tilde{N}_2(v, w) \cup N_3(v, w)) \subseteq N[z]$

Rule:

- 1) **(case 1.1)** If $(\tilde{N}_2(v, w) \cup N_3(v, w)) \subseteq (N(v) \cap N(w))$, then
 - $\mathcal{L}(v) \leftarrow 2$, $\mathcal{L}(w) \leftarrow 2$, $\mathcal{P} \leftarrow \mathcal{P} \cup \{v, w\}$
 - If $\{v, w\} \in E(\mathcal{G})$, set $\mathcal{L}((N_2(v, w) \cap N(v) \cap N(w)) \cup N_3(v, w)) \leftarrow -1$ and $\mathcal{L}(N_2(v, w) \setminus (N(v) \cap N(w))) \leftarrow 0$
 - Denote $\tilde{U} := (\tilde{N}_2(v, w) \cap N(v) \cap N(w)) \cup N_3(v, w)$. If $\{v, w\} \notin E(\mathcal{G})$ and $(N(v) \cap N(w)) \not\subseteq \tilde{U}$, then $\mathcal{L}(\tilde{U}) \leftarrow -1$ and $\mathcal{L}(N_2(v, w) \setminus \tilde{U}) \leftarrow 0$
 - Denote $\tilde{U} := (\tilde{N}_2(v, w) \cap N(v) \cap N(w)) \cup N_3(v, w)$. If $\{v, w\} \notin E(\mathcal{G})$ and $N(v) \cap N(w) \subseteq \tilde{U}$, pick arbitrarily $z \in \tilde{U}$ and set $\mathcal{L}(\tilde{U} \setminus \{z\}) \leftarrow -1$, $\mathcal{L}((N_2(v, w) \setminus \tilde{U}) \cup \{z\}) \leftarrow 0$
- 2) **(case 1.2)** If $(\tilde{N}_2(v, w) \cup N_3(v, w)) \subseteq N(v)$ but $(\tilde{N}_2(v, w) \cup N_3(v, w)) \not\subseteq N(w)$, then
 - $\mathcal{L}(v) \leftarrow 1$
 - If $\{v, w\} \in E(\mathcal{G})$, then $\mathcal{L}((N_2(v, w) \cap N(v)) \cup N_3(v, w)) \leftarrow -1$ and $\mathcal{L}(N_2(v, w) \setminus N(v)) \leftarrow 0$
 - Denote $\tilde{U} := (\tilde{N}_2(v, w) \cap N(v)) \cup N_3(v, w)$. If $\{v, w\} \notin E(\mathcal{G})$ and $(N(v) \cap N(w)) \not\subseteq \tilde{U}$, then $\mathcal{L}(\tilde{U}) \leftarrow -1$ and $\mathcal{L}(N_2(v, w) \setminus \tilde{U}) \leftarrow 0$
 - Denote $\tilde{U} := (\tilde{N}_2(v, w) \cap N(v)) \cup N_3(v, w)$. If $\{v, w\} \notin E(\mathcal{G})$ and $N(v) \cap N(w) \subseteq \tilde{U}$, pick arbitrarily $z \in \tilde{U}$ and set $\mathcal{L}(\tilde{U} \setminus \{z\}) \leftarrow -1$, $\mathcal{L}((N_2(v, w) \setminus \tilde{U}) \cup \{z\}) \leftarrow 0$
- 3) **(case 1.3)** If $(\tilde{N}_2(v, w) \cup N_3(v, w)) \not\subseteq N(v)$ but $(\tilde{N}_2(v, w) \cup N_3(v, w)) \subseteq N(w)$, then
 - $\mathcal{L}(w) \leftarrow 1$
 - If $\{v, w\} \in E(\mathcal{G})$, then $\mathcal{L}((N_2(v, w) \cap N(w)) \cup N_3(v, w)) \leftarrow -1$ and $\mathcal{L}(N_2(v, w) \setminus N(w)) \leftarrow 0$
 - Denote $\tilde{U} := (\tilde{N}_2(v, w) \cap N(w)) \cup N_3(v, w)$. If $\{v, w\} \notin E(\mathcal{G})$ and $(N(v) \cap N(w)) \not\subseteq \tilde{U}$, then $\mathcal{L}(\tilde{U}) \leftarrow -1$ and $\mathcal{L}(N_2(v, w) \setminus \tilde{U}) \leftarrow 0$
 - Denote $\tilde{U} := (\tilde{N}_2(v, w) \cap N(w)) \cup N_3(v, w)$. If $\{v, w\} \notin E(\mathcal{G})$ and $N(v) \cap N(w) \subseteq \tilde{U}$, pick arbitrarily $z \in \tilde{U}$ and set $\mathcal{L}(\tilde{U} \setminus \{z\}) \leftarrow -1$, $\mathcal{L}((N_2(v, w) \setminus \tilde{U}) \cup \{z\}) \leftarrow 0$
- 4) **(case 2)** If $(\tilde{N}_2(v, w) \cup N_3(v, w)) \not\subseteq N(v)$ and $(\tilde{N}_2(v, w) \cup N_3(v, w)) \not\subseteq N(w)$, then

- $\mathcal{L}(v) \leftarrow 1$, $\mathcal{L}(w) \leftarrow 1$
- If $\{v, w\} \in E(\mathcal{G})$ or $(N(v) \cap N(w)) \not\subseteq (N_2(v, w) \cup N_3(v, w))$, then $\mathcal{L}(N_2 \cup N_3) \leftarrow -1$
- If $\{v, w\} \notin E(\mathcal{G})$ and $(N(v) \cap N(w)) \subseteq (N_2(v, w) \cup N_3(v, w))$, arbitrarily pick $z \in N_2(v, w) \cup N_3(v, w)$. Update $\mathcal{L}(z) \leftarrow 0$, $\mathcal{L}((N_2(v, w) \cup N_3(v, w)) \setminus \{z\}) \leftarrow -1$.

Proof: See Appendix B.

IV. INTEGER PROGRAM FORMULATIONS

In this section we present the integer program formulations of the condensed RCDS problem in (7). The annotated RCDS problem in (2) is a special case of (7) with H interpreted as \mathcal{G} . To describe the integer program formulations, we need to define for any $\{H_i, H_j\} \in E(H)$, the symbol

$$X_{ij} = \{v \in V(\mathcal{G}) \mid \{H_i, H_j\} \in \mathcal{E}_H(\{v\})\} \quad (13)$$

where $\mathcal{E}_H(\cdot)$ is defined in (6). X_{ij} denotes the set of “enablers” of $\{H_i, H_j\} \in E(H)$. Also, we define the “bidirectional” version of $E(H)$ as $A(H) := \{(H_i, H_j), (H_j, H_i) \mid \{H_i, H_j\} \in E(H)\}$.

A. Single commodity flow (SCF) formulation

The decision variables are $x_v \in \{0, 1\}$ for $v \in V(\mathcal{G})$ and $f_{ij} \in \mathbb{R}_+$ for $(H_i, H_j) \in A(H)$. $v \in V(\mathcal{G})$ is included in the solution set D if and only if $x_v = 1$. f are auxiliary “flow variables” required to impose the connectedness constraint in (7). Let s corresponds to an arbitrarily designated “source” $H_s \in V(H)$. The proposed SCF formulation is as follows:

$$\begin{aligned} \min_{x, f} \quad & \sum_{v \in V(\mathcal{G})} x_v \\ \text{s.t.} \quad & \sum_{j: H_j \in N(H_i)} f_{ij} - \sum_{j: H_j \in N(H_i)} f_{ji} = -1, \quad \forall i \neq s \quad (14a) \\ & f_{ij} \leq (|V(H)| - 1) \sum_{v \in X_{ij}} x_v, \quad \forall \{H_i, H_j\} \in E(H) \quad (14b) \\ & f_{ji} \leq (|V(H)| - 1) \sum_{v \in X_{ij}} x_v, \quad \forall \{H_i, H_j\} \in E(H) \quad (14c) \\ & x_u = 0, \quad \forall u \in \mathcal{L}^{-1}(0) \quad (14d) \\ & x_v = 1, \quad \forall v \in \mathcal{L}^{-1}(1) \quad (14e) \\ & x_v + x_w \geq 1, \quad \forall \{v, w\} \in \mathcal{P} \quad (14f) \end{aligned}$$

(14b) and (14c) together are “flow enabling” constraints, specifying that the flow (forward or reverse direction) on edge $\{H_i, H_j\} \in E(H)$ can be nonzero if and only if the edge is enabled by including at least one enabler in X_{ij} in the solution set. Constraint (14a) specifies that exactly one unit of flow needs to be delivered from source H_s to each non-source node. This is possible if and only if a spanning tree of H can be formed using only the enabled edges (i.e., $\mathcal{E}_H(D)$ where $v \in D$ if and only if $x_v = 1$). Constraints (14d) to (14f) enforce the presolving outcome specified by \mathcal{L} and \mathcal{P} .

The SCF formulation (14) is simple but the “big- M ” constant $|V(H)| - 1$ is usually undesired. Solving (14) to optimality may not be easy. However, the simplicity of (14) often means good suboptimal solutions can be found early in solution process. This benefit will be shown in Section V.

B. Dantzig-Fulkerson-Johnson (DFJ) formulation

Similar to SCF, the DFJ formulation involves binary decision variables $x_v \in \{0, 1\}$ for $v \in V(\mathcal{G})$ to encode the inclusion of nodes in the solution set D . In addition, auxiliary binary “edge” decision variables are needed: $y_{ij} \in \{0, 1\}$ for $(H_i, H_j) \in A(H)$. The y -variables characterize a spanning tree of $(V(H), \mathcal{E}_H(D))$ to certify its connectedness. Similar to SCF, we denote s such that $H_s \in V(H)$ is an arbitrarily designated “root”. The proposed DFJ formulation is as follows:

$$\min_{x, y} \sum_{v \in V(\mathcal{G})} x_v \quad (15a)$$

$$\text{s.t.} \quad \sum_{j: H_j \in N(H_s)} y_{sj} = 0 \quad (15b)$$

$$\sum_{j: H_j \in N(H_s)} y_{js} \geq 1 \quad (15c)$$

$$\sum_{j: H_j \in N(H_i)} y_{ij} = 1, \quad \forall i \neq s \quad (15d)$$

$$\sum_{(H_i, H_j) \in A(H)} y_{ij} = |V(H)| - 1, \quad (15e)$$

$$y_{ij} + y_{ji} \leq \sum_{v \in X_{ij}} x_v, \quad \forall \{H_i, H_j\} \in E(H) \quad (15f)$$

$$y_{ij} + y_{ji} \leq 1, \quad \forall \{H_i, H_j\} \in E(H) \quad (15g)$$

$$\sum_{\{H_i, H_j\} \in E(S)} (y_{ij} + y_{ji}) < |S|, \quad \forall S \subseteq V(H), |S| \geq 3 \quad (15h)$$

$$x_u = 0, \quad \forall u \in \mathcal{L}^{-1}(0) \quad (15i)$$

$$x_v = 1, \quad \forall v \in \mathcal{L}^{-1}(1) \quad (15j)$$

$$x_v + x_w \geq 1, \quad \forall \{v, w\} \in \mathcal{P} \quad (15k)$$

Constraints (15e) and (15f) specify that for every $\{H_i, H_j\} \in E(H)$ at most one direction (H_i, H_j) or (H_j, H_i) can be used to construct an in-root tree to H_s if and only if $\{H_i, H_j\} \in \mathcal{E}_H(D)$ where $D = \{v \in V(\mathcal{G}) \mid x_v = 1\}$. Then, constraints (15a) to (15d) are necessary conditions of an in-root spanning tree to H_s in $(V(H), \mathcal{E}_H(D))$: there is no outgoing edge from H_s and at least one incoming edge to H_s . For all other nodes there is exactly one outgoing edge. Constraint (15d) specifies that the tree contains $|V(H)| - 1$ edges. However, subtours (i.e., disjoint directed cycles) may exist in $(V(H), \mathcal{E}_H(D))$. To exclude subtours, the subtour elimination constraints (15f) and (15g) are imposed since a graph is acyclic if and only if it has no induced subgraph with $|S|$ nodes but at least $|S|$ edges. In conclusion, $(V(\mathcal{G}), \mathcal{E}_H(D))$ is an in-root spanning tree of H if and only if constraints (15a) through (15g) are satisfied. However, (15g) involves exponentially many constraints because of the quantifier $\forall S \subseteq V(H)$. Modern integer programming solvers (e.g., Gurobi, CPLEX) can model these as “lazy constraints” – a user-defined callback function is specified when calling Gurobi to add the relevant subtour elimination constraints only when needed (cf. “cutting-plane”) [37]. In our implementation, the callback function returns the S corresponding to the minimum length cycle to augment (15g).

While the DFJ formulation in (15) may require exponentially many constraints, it is usually “strong” in that its linear program relaxations are typically more “useful” (cf. [38] for travelling salesman problem). Our result in Section V supports this observation for the RCDS problem in (7).

C. Miller-Tucker-Zemlin (MTZ) formulation

The MTZ formulation is the same as DFJ except for the subtour elimination constraints. The decision variables of the MTZ formulation include $x_v \in \{0, 1\}$ for $v \in V(\mathcal{G})$ and $y_{ij} \in \{0, 1\}$ for $(H_i, H_j) \in A(H)$ as with DFJ. In addition, continuous decision variables $u_v \in [1, |V(H)| - 1]$ for $v \in V(H) \setminus \{H_s\}$ are needed to describe the MTZ’s variant of subtour elimination constraints. The proposed MTZ formulation is as follows:

$$\min_{x, u, y} \sum_{v \in V(\mathcal{G})} x_v \quad (16a)$$

$$\text{s.t.} \quad \sum_{j: H_j \in N(H_s)} y_{sj} = 0 \quad (16b)$$

$$\sum_{j: H_j \in N(H_s)} y_{js} \geq 1 \quad (16c)$$

$$\sum_{j: H_j \in N(H_i)} y_{ij} = 1, \quad \forall i \neq s \quad (16d)$$

$$\sum_{(H_i, H_j) \in A(H)} y_{ij} = |V(H)| - 1, \quad (16e)$$

$$y_{ij} + y_{ji} \leq \sum_{v \in X_{ij}} x_v, \quad \forall \{H_i, H_j\} \in E(H) \quad (16f)$$

$$y_{ij} + y_{ji} \leq 1, \quad \forall \{H_i, H_j\} \in E(H) \quad (16g)$$

$$(|V(H)| - 1)y_{ij} + (|V(H)| - 3)y_{ji} + u_i - u_j \leq |V(H)| - 2, \quad \forall (H_i, H_j) \in A(H), i \neq s, j \neq s \quad (16h)$$

$$1 \leq u_v \leq |V(H)| - 1, \quad \forall v \in V(H) \setminus \{H_s\} \quad (16i)$$

$$x_u = 0, \quad \forall u \in \mathcal{L}^{-1}(0) \quad (16j)$$

$$x_v = 1, \quad \forall v \in \mathcal{L}^{-1}(1) \quad (16k)$$

$$x_v + x_w \geq 1, \quad \forall \{v, w\} \in \mathcal{P} \quad (16l)$$

Comparing to (15), formulation (16) differs only in (16g) and (16h) which form the MTZ’s version of subtour elimination constraints. Let (x, u, y) be a feasible solution of (16). Suppose y leads to a subtour in $(V(H), \mathcal{E}_H(D))$. Let the subtour be $H_{i_0}, H_{i_1}, \dots, H_{i_p} = H_{i_0}$. Since (16a) specifies that root H_s does not have any outgoing edge in $(V(H), \mathcal{E}_H(D))$, H_s cannot be part of the subtour. Thus, constraint (16g) applies to all subtour directed edges $(H_{i_k}, H_{i_{k+1}})$ for $k = 0, \dots, p - 1$, leading to

$$|V(H)| - 1 + u_{i_k} - u_{i_{k+1}} \leq |V(H)| - 2, \quad k = 0, \dots, p - 1$$

Adding all inequalities above yields $-1 \leq -2$. Thus, y cannot lead to any subtour. Conversely, let T_s be a directed in-root spanning tree of H with roots H_s and for $v \in V(H) \setminus \{H_s\}$ let d_v denote the (edge) distance from v to H_s along $E(T_s)$. Then, $u_v := |V(H)| - d_v$ satisfies (16h) since $1 \leq u_v \leq |V(H)| - 1$ for all $v \in V(H) \setminus \{H_s\}$. Also, let $y_{ij} = 1$ if and only if $(H_i, H_j) \in E(T_s)$. Then, for each $(H_i, H_j) \in A(H)$ with $i, j \neq s$, u_i and u_j are defined and the left-hand-side of (16g) evaluates to

$$|V(H)| - 1 - d_j + d_i = |V(H)| - 2 \quad \text{if } (H_i, H_j) \in E(T_s)$$

$$|V(H)| - 3 - d_j + d_i = |V(H)| - 2 \quad \text{if } (H_j, H_i) \in E(T_s)$$

$$u_i - u_j \leq |V(H)| - 2 \quad \text{otherwise}$$

Thus, (16g) is satisfied by u and y . Together with a properly chosen x , y satisfies all other constraints in (16) as well.

In our case study, MTZ is on par with DFJ. However, for the largest example ACTIVSg70k, MTZ is much better.

V. NUMERICAL CASE STUDIES

In this section, the RCDS problem defined on power network examples in MATPOWER 7 [35] are considered. The effect of presolving and condensation in Section III are demonstrated. In addition, the three different integer program formulations in Section IV are evaluated. All instances of RCDS problem assume $\mathcal{B} = \mathcal{W} = \emptyset$ initially. All computations are performed on a PC with a CPU at 4.2 GHz (4 cores) and 16 GB of RAM. The presolving procedures are implemented using MATLAB and C. The integer programs are solved using Gurobi 9, called either from MATLAB or from C.

For the IEEE 14-bus example, by solving (1) an optimal RCDS is $D^* = \{2, 6, 7, 9\}$ including four buses. Fig. 2 shows

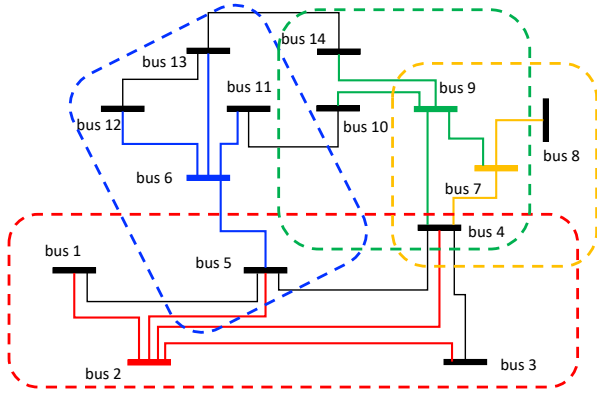


Fig. 2: A minimum RCDS of IEEE 14-bus benchmark

the layout of the RCDS. The color-coded edges of $E_G[D^*]$ indicate that the graph $(V, E_G[D^*])$ is connected, satisfying definition (RCDS-D2). On the other hand, Fig. 2 shows that D^* is a dominating set of G and together with D^* -path 6, 5, 2, 4, 7, 9 it can be seen that (RCDS-D1) is satisfied as well.

Next, we examine the effect of presolving and condensation in Section III on instance size reduction. Let deg1, deg2-T, deg2-P, Nb1, Nb0, brG, CC1, domX, brH, scan1, scan2 denote the presolving procedures in Section III, in the order they appear in the section. Table I shows the cumulative progress of applying the presolving procedures to example 13659pegase with 13659 buses. In Table I, the columns < 0 ,

TABLE I: Cumulative presolving progress, 13659pegase

method	< 0 (%)	$= 0$ (%)	$= 1$ (%)	$= 2$ (%)	fixed (%)
scan2	46.1	8.4	17.0	0.5	71.5
scan1	46.4	9.0	17.3	0.5	72.6
deg2-T	46.4	9.0	17.6	0.5	72.9
deg2-P	46.4	9.3	17.6	0.5	73.2
Nb1	47.2	10.3	17.0	0.5	74.5
Nb0	47.2	10.3	17.2	0.4	74.8
brG	47.2	10.3	17.6	0.5	75.2
CC1	47.2	11.1	17.6	0.5	75.9
domX	47.2	12.9	17.6	0.5	77.8
brH	47.2	12.9	18.2	0.4	78.3

$= 0$, $= 1$ and $= 2$ correspond to the percentage of nodes in G (i.e., the original graph) with the corresponding \mathcal{L} values. For example, after scan2 and scan1 46.4% of nodes are removed from G , 9% of nodes are excluded from the

optimal RCDS while 17.3% of nodes must be included in the optimal RCDS, etc. The last column in Table I shows the percentage of nodes whose RCDS membership are determined because of presolving. An analogous result for the 70k-bus benchmark ACTIVSg70k is shown in Table II, also suggesting significant RCDS membership determination. It is noted that

TABLE II: Cumulative presolving progress, ACTIVSg70k

method	< 0 (%)	$= 0$ (%)	$= 1$ (%)	$= 2$ (%)	fixed (%)
scan2	32.7	11.2	25.0	0.1	68.9
scan1	33.1	11.3	25.5	0.1	69.9
deg2-T	33.1	11.3	25.5	0.1	69.9
deg2-P	33.1	11.4	25.5	0.1	70.0
Nb1	35.4	14.5	23.9	0.1	73.9
Nb0	35.4	14.5	24.3	0.0	74.3
brG	35.4	14.5	24.4	0.0	74.4
CC1	35.4	15.1	24.4	0.0	75.0
domX	35.4	17.5	24.4	0.0	77.3
brH	35.4	17.5	24.9	0.0	77.8

scan2 is found to be most effective and hence it is run first. However, the presolving procedures can be run in any order with or without repetition. Our initial assessment is that more aggressive batch run improves the determination, but the improvement does not seem to be significant.

A summary of the effect of presolving and condensation for selected benchmarks in MATPOWER 7 is shown in Table III. The number of nodes of the benchmarks are indicated in

TABLE III: Summary of presolving and condensation

example	$ E(G) $	time (s)	presolved	$\frac{ V(H) }{ V(G) }$	$\frac{ E(H) }{ E(G) }$
IEEE 300	409	0.0162	65.3%	30%	35.0%
2848rte	3442	0.1914	88.9%	7.5%	9.8%
6515rte	8104	0.8935	82.3%	14.1%	18.0%
1354pegase	1710	0.0451	88.8%	7.8%	9.4%
2869pegase	3968	0.246	68.8%	29.2%	32.2%
9241pegase	14207	2.666	57.2%	41.3%	44.8%
13659pegase	18625	3.396	78.3%	18.8%	23.1%
2383wp	2886	0.1812	61.5%	35.7%	41.6%
3120sp	3684	0.3286	53.6%	43.6%	48.9%
ACTIVSg10k	12217	2.1089	76.0%	19.1%	25%
ACTIVSg25k	30111	10.912	81.6%	13.3%	18.2%
ACTIVSg70k	83318	83.04	77.8%	17.8%	23%

the names. The third column (i.e., time) lists the time to presolve and to find the condensations. The fourth column (i.e., presolved) is the percentage of nodes in G whose RCDS membership is determined during presolving (cf. Tables I and II). It can be seen that with negligible computation effort (comparing to solving integer programs), presolving can determine the RCDS membership of significant portions of the nodes in G before solving the remaining problem in (7). Also, the dimension reduction from G to H is substantial (e.g., ACTIVSg70k). This indicates that the structure of power system graphs can be utilized for computation speedup while solving combinatorial problems on these graphs.

To illustrate the effect of presolving and condensation on solution computation speedup, the original version in (1), the annotated version in (2) and the condensed version in (7) are solved for the larger networks of 13659pegase, ACTIVSg25k and ACTIVSg70k. Each of these nine combinations is modeled with the SCF formulation in (14), the DFJ formulation in

(15) and MTZ formulation in (16) respectively. Since Gurobi 9 has difficulty obtaining feasible solutions when solving (15) and (16), initial feasible solutions are specified when solving these two formulations. The initial feasible solutions initialize all possible x variables to one (except for those whose \mathcal{L} values are 0) and initialize the remaining y and u variables based on the in-root spanning trees obtained using breadth-first search. Each time Gurobi is called, a time limit of 15 minutes is imposed. The result of the computation experiment is shown in Table IV, which clearly indicates the condensed RCDS problem in (7) should be solved instead of others. Table IV also indicates that presolving alone without condensation may not result in sufficiently accurate solution to the RCDS problem. Further, we note that the DFJ formulation

TABLE IV: Percentage optimality gap obtained when solving RCDS problem with different levels of presolving (15 min)

case	problem	SCF	DFJ	MTZ
13659pegase	(1)	0.48%	199%	3.75%
13659pegase	(2)	0.45%	71.1%	1.52%
13659pegase	(7)	0.31%	0.11%	0.17%
ACTIVSg25k	(1)	0.77%	99.9%	2.92%
ACTIVSg25k	(2)	0.76%	25.0%	0.55%
ACTIVSg25k	(7)	0.14%	0.036%	0%
ACTIVSg70k	(1)	9.95%	133%	9.28%
ACTIVSg70k	(2)	2.94%	43.7%	12.9%
ACTIVSg70k	(7)	0.63%	25.2%	0.087%

may not lead to acceptable solution in the ACTIVSg70k case.

It turns out that specifying good initial feasible solutions could significantly speed up the solution process. In the following experiment, for selected benchmarks problem (7) modeled in the SCF formulation in (14) is solved using Gurobi 9 for 3 minutes. The resulted suboptimal solutions are then used as initial feasible solutions for the DFJ and MTZ formulations in (15) and (16). The later two formulations are solved using Gurobi 9 for another 15 minutes each. Table V shows the statistics of the computation experiment. The second column

TABLE V: Statistics for mixed SCF/DFJ/MTZ (3 + 15 min)

case	SCF (gap)	DFJ (U/L/gap)	MTZ (U/L/gap)
IEEE 300	0%	93/93/ 0%	93/93/ 0%
2848rte	0%	971/971/ 0%	971/971/ 0%
6515rte	0.14%	2085/2085/ 0%	2085/2085/ 0%
1354pegase	0%	407/407/ 0%	407/407/ 0%
2869pegase	0.48%	841/839/ 0.24%	841/838/0.36%
9241pegase	1.38%	2780/2769/ 0.40%	2776/2763/0.47%
13659pegase	0.48%	3548/3546/ 0.056%	3551/3544/0.20%
2383wp	0.49%	821/820/ 0.12%	821/820/ 0.12%
3120sp	2.60%	1138/1138/ 0%	1138/1137/0.088%
ACTIVSg10k	0.46%	3304/3303/ 0.03%	3304/3302/0.061%
ACTIVSg25k	0.77%	8370/8370/ 0%	8370/8369/0.012%
ACTIVSg70k	9.95%	24387/24040/1.44%	24062/24035/ 0.11%

in Table V (i.e., SCF (gap)) shows the percentage optimality gap of solving problem (1) in SCF formulation in (14) with 15-minute time limit, without specifying any initial feasible solution to Gurobi. This serves as the “baseline” approach. The third and fourth columns show the objective upper bound, lower bound and percentage optimality gap for the mixed approaches described earlier: (a) 3-minute SCF plus 15-minute

DFJ and (b) 3-minute SCF plus 15-minute MTZ. The mixed approaches result in very accurate results within 18 minutes.

For the ACTIVSg70k example the experiment in Table V is extended. The main computations to solve ((1), SCF), ((7), DFJ) and ((7), MTZ) are extended to 4 hours. Also, to find the initial feasible solutions the computation for ((7), SCF) is extended to 5 minutes. Fig. 3 shows the percentage optimality gaps resulted from SCF, DFJ and MTZ formulations respectively. For the ACTIVSg70k example, MTZ has the

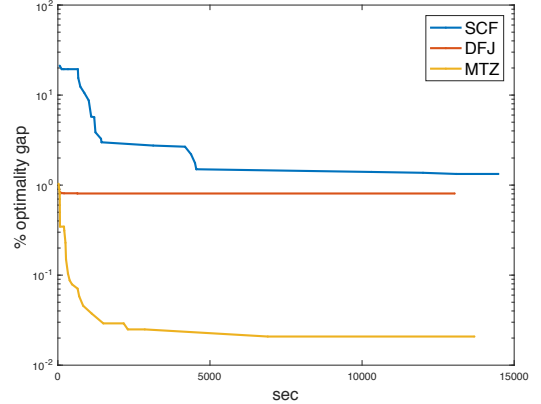


Fig. 3: Percentage optimality gap, ACTIVSg70k (4 hours)

best performance with objective upper bounded by 24050 and lower bounded by 24045. Though optimality gap progress slows down significantly during the process. Additional measures may be required if exact solution is necessary.

VI. CONCLUSION

Solving integer programs (using branch-and-bound) is likely to be the method of choice to obtain high-quality solutions to power system planning and operation problems and presolving is indispensable for this effort. However, generic presolving routines (as available in Gurobi) may not be sufficient and should be enhanced by problem-specific variants as shown in this study for the RCDS problem. This is especially relevant for larger instances (e.g., ACTIVSg70k). Despite our effort, the list of presolving options presented in this paper is unlikely to be exhaustive and the best combination of presolving runs may require more extensive effort to find. It is also noted that the presolving results in this paper pertain only to the unweighted RCDS problem. Non-uniform weights introduce more complicated global dependency between the nodes. Presolving due to feasibility will go through (e.g., bridge), while others based on objective value comparison may need an overhaul. The SCF formulation is simple and tends to be robust, while the DFJ and MTZ formulations may lead to better results if care is taken (e.g., choice of initial feasible solution). Though the systematic study of these algorithmic choices are outside the scope of this paper. Due to power and information flow, graph connectedness and domination remain vital to power system planning and operation modelling. It is hoped that the results in this study can benefit other related problems such as the (connected) power dominating set problem or other equipment placement problems.

VII. APPENDIX A

In the proof, we abbreviate $N_1(v)$ as N_1 and other notations in (11) are treated similarly. First we justify the update $\mathcal{L}(N_2 \cup N_3) \leftarrow 0$. Suppose $D \in \mathcal{V}(\mathcal{G})$ with $D \cap (N_2 \cup N_3) \neq \emptyset$ is feasible to (2). We show that $\hat{D} := D \setminus (N_2 \cup N_3) \cup \{v\}$ is also feasible to (2) with $|\hat{D}| \leq |D|$. \hat{D} is feasible to (2) if (a) $\hat{D} \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ in (4) and (b) \hat{D} is a RCDS of \mathcal{G} . Since $\mathcal{L}(v) \neq 2$ definition (11) implies that $\mathcal{L}(u) = 0$ or 3 for all $u \in N_2 \cup N_3$. This, together with $\mathcal{L}(v) = 3$, implies that $\hat{D} \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ establishing (a). For (b), first note that $N[u] \subseteq N[v]$ for all $u \in N_2 \cup N_3$ and hence \hat{D} is a dominating set of \mathcal{G} (since D is). This establishes the first part of (RCDS-D1) for \hat{D} . The second part of (RCDS-D1) is argued as follows. Let $s, t \in (\hat{D} \setminus \{v\}) \subseteq D$, then there exists a D -path in \mathcal{G} between s and t since D is a RCDS of \mathcal{G} . If the D -path contains a part in $N_2 \cup N_3$, namely x, u_1, \dots, u_p, y where $u_1, \dots, u_p \in N_2 \cup N_3$ but $x, y \notin N_2 \cup N_3$. Then, replace the part with x, v, y . By removing possible loops, a \hat{D} -path can be found in \mathcal{G} between s and t . In addition, for any $w \in \hat{D} \setminus \{v\}$, a D -path exists between w and u for some $u \in N_2 \cup N_3$ since $D \cap (N_2 \cup N_3) \neq \emptyset$. Since $N[z] \subseteq N[v]$ for all $z \in N_2 \cup N_3$, a \hat{D} -path can be constructed between w and v by replacing all parts of the D -path involving $N_2 \cup N_3$ (including u) by v . This concludes that between any two nodes in \hat{D} there is a \hat{D} -path in \mathcal{G} between them. Hence, \hat{D} is a RCDS of \mathcal{G} and then \hat{D} is feasible to (2). This justifies $\mathcal{L}(N_2 \cup N_3) \leftarrow 0$. Next, we justify $\mathcal{L}(v) \leftarrow 1$. According to definition (11), $\tilde{N}_2 \cup N_3$ (assumed nonempty) can be dominated only by nodes in $N_2 \cup N_3 \cup \{v\}$. The update $\mathcal{L}(N_2 \cup N_3) \leftarrow 0$ then necessitates $\mathcal{L}(v) \leftarrow 1$. Finally, $\mathcal{L}(N_2 \cup N_3) \leftarrow 0$, $\mathcal{L}(v) \leftarrow 1$ and $N_2 \cup N_3 \subseteq N[v]$ imply assumptions 1) and 2) of Proposition 3. In addition, assumption 3) holds since for every $D \in \mathcal{D}(\mathcal{G}, \mathcal{P})$ any part of a D -path involving $N_2 \cup N_3$ can be replaced by v as $N(u) \subseteq N[v]$ for all $u \in N_2 \cup N_3$. Therefore, the update $\mathcal{L}(N_2 \cup N_3) \leftarrow -1$ is justified.

VIII. APPENDIX B

In the proof, we simplify the notations in (12). For instance, N_2 is shorthand for $N_2(v, w)$. We prove case 1.1 only. This proof can be specialized to show the other (easier) cases. Similar to the proof of the presolving procedures discussed earlier, case 1.1 is established in three steps: (a) $\mathcal{L}(N_2 \cup N_3) \leftarrow 0$, (b) $\mathcal{P} \leftarrow \mathcal{P} \cup \{v, w\}$ and (c) further labelling of part of $N_2 \cup N_3$ to -1 where appropriate. Suppose $D \subseteq \mathcal{V}(\mathcal{G})$ is feasible to (2). Then D is a RCDS of \mathcal{G} . By (RCDS-D1), D is a dominating set of \mathcal{G} . In addition, the prerequisite (item 3) specifies that $\tilde{N}_2 \cup N_3 \neq \emptyset$ and since $\tilde{N}_2 \cup N_3$ can be dominated only by nodes in $M := \{v, w\} \cup N_2 \cup N_3$, it holds that $(D \cap M) \neq \emptyset$. If $|D \cap M| = 1$ then the prerequisite (item 4) implies that $(D \cap M) \subseteq \{v, w\}$. In other words, $D \cap (N_2 \cup N_3) = \emptyset$ when $|D \cap M| = 1$. On the other hand, suppose $|D \cap M| \geq 2$ and $D \cap (N_2 \cup N_3) \neq \emptyset$. Then it is claimed that $\hat{D} := (D \setminus (N_2 \cup N_3)) \cup \{v, w\}$ is also feasible to (2) with $|\hat{D}| \leq |D|$. To see the claim, first note that $\mathcal{L}(u) = 0$ or 3 for all $u \in N_2 \cup N_3$ and $\mathcal{L}(v) = \mathcal{L}(w) = 3$ imply that $\hat{D} \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ defined in (4). Since $N[u] \subseteq N[v, w]$ for all $u \in N_2 \cup N_3$, \hat{D} is a dominating set of \mathcal{G} (since D is), establishing that \hat{D} satisfies the first condition of (RCDS-D1). For the second condition of (RCDS-D1), note that for any $s, t \in \hat{D} \setminus \{v, w\}$, then $s, t \in D$

and hence there exists a D -path between s and t . We argue that a \hat{D} -path can be constructed between s and t as follows. Consider (if any) part of D -path involving $N_2 \cup N_3$. Namely, x, u_1, \dots, u_p, y such that $x, y \notin N_2 \cup N_3$ but $u_1, \dots, u_p \in N_2 \cup N_3$. If $\{v, w\} \in E(\mathcal{G})$, replace the part with one of the following depending on adjacency: x, v, w, y or x, w, v, y or x, v, y or x, w, y . On the other hand, if $(v, w) \notin E(\mathcal{G})$ then the prerequisite (item 2) $N(v) \cap N(w) \neq \emptyset$ implies some $z \in N(v) \cap N(w)$. The part of D -path x, u_1, \dots, u_p, y can then be replaced with one of the following depending on adjacency: x, v, z, w, y or x, w, z, v, y or x, v, y or x, w, y . Whether $\{v, w\} \in E(\mathcal{G})$ or not, a \hat{D} -path can be constructed between s and t since $v, w \in \hat{D}$. Since $D \cap (N_2 \cup N_3)$ a D -path can be found between any $s \in \hat{D} \setminus \{v, w\}$ and some $u \in N_2 \cup N_3$. This D -path can be modified into a \hat{D} -path between s and v (resp., w) as argued above. Consequently, \hat{D} is a RCDS of \mathcal{G} and is hence feasible to (2). Therefore, regardless of $|D \cap M|$ it is always possible to assume that $D \cap (N_2 \cup N_3) = \emptyset$. This justifies $\mathcal{L}(N_2 \cup N_3) \leftarrow 0$. This further implies that at least one of v or w must be in the optimal solution of (2) since $D \cap M \neq \emptyset$ for any D feasible to (2). This justifies the update $\mathcal{L}(v) \leftarrow 2$, $\mathcal{L}(w) \leftarrow 2$ and $\mathcal{P} \leftarrow \mathcal{P} \cup \{v, w\}$. The remaining part of the proof establishes that certain subset of $N_2 \cup N_3$ can be re-labeled to be -1 depending on the situations specified in the last three bullets in the description of case 1.1. First, suppose $\{v, w\} \in E(\mathcal{G})$. Let $D \in \mathcal{D}(\mathcal{L}, \mathcal{P})$ in (4) with the updated \mathcal{L} and \mathcal{P} (NB: not the same D earlier in the proof). Denote $U := (N_2 \cap N(v) \cap N(w)) \cup N_3$. Then, $\mathcal{L}(U) = 0$ since $U \subseteq (N_2 \cup N_3)$ and $\mathcal{L}(N_2 \cup N_3) = 0$. This is assumption 1) of Proposition 3. In addition, $\{v, w\} \in \mathcal{P}$, the definition of U and the assumption of case 1.1 (i.e., $N_3 \subseteq N(v) \cap N(w)$) together imply that U is dominated by both v and w , one of which is in $D \in \mathcal{D}$. This is assumption 2) of Proposition 3. Consider a D -path between $s, t \in D$ denoted by $s \sim t$. If $s \sim t$ contains a part traversing U of the form x, u_1, \dots, u_p, y where $u_1, \dots, u_p \in U$ but $x, y \notin U$, then it must be of the form x, u, y for $u \in U$ (i.e., $p = 1$) since $\mathcal{L}(U) = 0$ and no D -path traverses two consecutive nodes not in D . Also, $x, y \in D \cap N[v, w]$. Thus, at least one of $x, v, y, x, w, y, x, v, w, y$ and x, w, v, y is a path (after removing repeated nodes if necessary) in \mathcal{G} since $x, y \in N[v, w]$ and $\{v, w\} \in E(\mathcal{G})$. The path is in fact a D -path since $x, y \in D$ and $\{v, w\} \cap D \neq \emptyset$ as $\{v, w\} \in \mathcal{P}$. Therefore, the part x, u, y in D -path $s \sim t$ can be replaced by a D -path not traversing U . Repeat this process to replace all parts of the $s \sim t$ traversing U and we can obtain a D -path between s and t without traversing U . Therefore, when $\{v, w\} \in E(\mathcal{G})$ Proposition 3 implies that $\mathcal{L}(U) \leftarrow -1$. Next, we let $\tilde{U} := (\tilde{N}_2 \cap N(v) \cap N(w)) \cup N_3$ and consider the case where $\{v, w\} \notin E(\mathcal{G})$ and $N(v) \cap N(w) \not\subseteq \tilde{U}$. \tilde{U} satisfies assumptions 1) and 2) in Proposition 3 since $\tilde{U} \subseteq U$. For assumption 3) of Proposition 3, let $s \sim t$ be a D -path between $s, t \in D$. As argued before, if $s \sim t$ contains a part traversing \tilde{U} then the part must be of the form x, u, y where $u \in \tilde{U}$ and $x, y \in D \setminus \tilde{U}$ since $\mathcal{L}(\tilde{U}) = 0$. By definition, $N[\tilde{u}] \subseteq \{v, w\} \cup (N_1 \setminus D) \cup N_2 \cup N_3$ for all $\tilde{u} \in \tilde{U}$. Since $\mathcal{L}(N_2 \cup N_3) = 0$, $x, y \in D$ implies that $x, y \in \{v, w\}$. If $x = y = v$ then x, u, y can be replaced by v . Similarly, if $x = y = w$ then x, u, y can be replaced by w . If $x = v$ and $y = w$, then let $z \in N(v) \cap N(w)$ (and $z \notin \tilde{U}$) and x, z, y is a valid replacement D -path since $x, y \in D$. The case with $x = w$ and $y = v$ is similar. In summary,

the part x, u, y of D -path $s \sim t$ can be replaced by another part not traversing \tilde{U} . Repeat the process and a D -path between s and t without traversing \tilde{U} can be found from $s \sim t$. This establishes assumption 3) of Proposition 3 and justifies the update $\mathcal{L}(\tilde{U}) \leftarrow -1$. Note that contrary to the case where $\{v, w\} \in E(\mathcal{G})$, we may update the labels to -1 for \tilde{U} instead of the entire U . This is because when considering D -paths traversing $U \setminus \tilde{U}$ the candidate replacement part v, z, w or w, z, v need not be a D -path since only one of v, w is guaranteed in D and z may not be in D . Lastly, we note that the case where $\{v, w\} \notin E(\mathcal{G})$ and $N(v) \cap N(w) \subseteq \tilde{U}$ can be argued similarly. The only exception is that $z \in N(v) \cap N(w)$ is in \tilde{U} and hence $\mathcal{L}(z) \leftarrow 0$ instead of -1 as the rest of \tilde{U} .

REFERENCES

- [1] M. R. Garey and D. S. Johnson, *Computers and intractability*. wh freeman, 2002, vol. 29.
- [2] C.-W. Ten, C.-C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for scada systems," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1836–1846, 2008.
- [3] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, pp. 1–33, 2011.
- [4] G. Dán and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 2010, pp. 214–219.
- [5] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 2010, pp. 220–225.
- [6] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, K. Poolla et al., "Smart grid data integrity attacks: characterizations and countermeasures," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*. IEEE, 2011, pp. 232–237.
- [7] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Worst-case innovation-based integrity attacks with side information on remote state estimation," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 1, pp. 48–59, 2018.
- [8] Q. Dai, L. Shi, and Y. Ni, "Risk assessment for cyberattack in active distribution systems considering the role of feeder automation," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 3230–3240, 2019.
- [9] S. A. Desilva, J. Kim, E. Cotilla-Sanchez, and T. Hagan, "On pmu data integrity under gps spoofing attacks: A sparse error correction framework," *IEEE Transactions on Power Systems*, 2021.
- [10] T. T. Kim and H. V. Poor, "Strategic protection against data injection attacks on power grids," *Smart Grid, IEEE Transactions on*, vol. 2, no. 2, pp. 326–333, 2011.
- [11] S. Bi and Y. J. Zhang, "Graphical methods for defense against false-data injection attacks on power system state estimation," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1216–1227, 2014.
- [12] A. Chattopadhyay and U. Mitra, "Security against false data-injection attack in cyber-physical systems," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 1015–1027, 2019.
- [13] M. Ghafouri, M. Au, M. Kassouf, M. Debbabi, C. Assi, and J. Yan, "Detection and mitigation of cyber attacks on voltage stability monitoring of smart grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5227–5238, 2020.
- [14] S. V. S. Chauhan and G. X. Gao, "Synchrophasor data under gps spoofing: Attack detection and mitigation using residuals," *IEEE Transactions on Smart Grid*, 2021.
- [15] K. C. Sou and J. Lu, "Relaxed connected dominating set problem with application to secure power network design," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 4657–4662.
- [16] G. Krumpolz, K. Clements, and P. Davis, "Power system observability: a practical algorithm using network topology," *IEEE Transactions on Power Apparatus and Systems*, pp. 1534–1542, 1980.
- [17] K. C. Sou, "Protection placement for power system state estimation measurement data integrity," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 638–647, 2019.
- [18] B. Gou and A. Abur, "An improved measurement placement algorithm for network observability," *Power Systems, IEEE Transactions on*, vol. 16, no. 4, pp. 819–824, 2001.
- [19] F. Aminifar, A. Khodaei, M. Fotuhi-Firuzabad, and M. Shahidepour, "Contingency-constrained pmu placement in power networks," *Power Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 516–523, 2010.
- [20] V. Frolov, P. G. Thakurta, S. Backhaus, J. Bialek, and M. Chertkov, "Operations-and uncertainty-aware installation of facts devices in a large transmission system," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 961–970, 2019.
- [21] L. Ye, S. Roy, and S. Sundaram, "Resilient sensor placement for kalman filtering in networked systems: Complexity and algorithms," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1870–1881, 2020.
- [22] R. G. Downey and M. R. Fellows, *Parameterized complexity*. Springer Science & Business Media, 1999.
- [23] M. W. Savelsbergh, "Preprocessing and probing techniques for mixed integer programming problems," *ORSA Journal on Computing*, vol. 6, no. 4, pp. 445–454, 1994.
- [24] L. Gurobi Optimization, "Advanced gurobi algorithms (2016)."
- [25] T. Achterberg, R. E. Bixby, Z. Gu, E. Rothberg, and D. Weninger, "Presolve reductions in mixed integer programming," *INFORMS Journal on Computing*, vol. 32, no. 2, pp. 473–506, 2020.
- [26] J. Alber, M. R. Fellows, and R. Niedermeier, "Polynomial-time data reduction for dominating set," *Journal of the ACM (JACM)*, vol. 51, no. 3, pp. 363–384, 2004.
- [27] J. Guo and R. Niedermeier, "Invitation to data reduction and problem kernelization," *ACM SIGACT News*, vol. 38, no. 1, pp. 31–45, 2007.
- [28] H. L. Bodlaender, "Kernelization: New upper and lower bound techniques," in *International Workshop on Parameterized and Exact Computation*. Springer, 2009, pp. 17–37.
- [29] H. L. Bodlaender, B. M. Jansen, and S. Kratsch, "Preprocessing for treewidth: A combinatorial analysis through kernelization," *SIAM Journal on Discrete Mathematics*, vol. 27, no. 4, pp. 2108–2142, 2013.
- [30] Q. Gu and N. Imani, "Connectivity is not a limit for kernelization: Planar connected dominating set," in *Latin American Symposium on Theoretical Informatics*. Springer, 2010, pp. 26–37.
- [31] E. Romero-Ramos, J. Riquelme-Santos, and J. Reyes, "A simpler and exact mathematical model for the computation of the minimal power losses tree," *Electric Power Systems Research*, vol. 80, no. 5, pp. 562–571, 2010.
- [32] M. Lavorato, J. F. Franco, M. J. Rider, and R. Romero, "Imposing radiality constraints in distribution system optimization problems," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 172–180, 2011.
- [33] H. Ahmadi and J. R. Martí, "Mathematical representation of radiality constraint in distribution system reconfiguration problem," *International journal of electrical power & energy systems*, vol. 64, pp. 293–299, 2015.
- [34] S. Lei, C. Chen, Y. Song, and Y. Hou, "Radiality constraints for resilient reconfiguration of distribution systems: Formulation and application to microgrid formation," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 3944–3956, 2020.
- [35] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 12–19, 2011.
- [36] R. E. Tarjan, "A note on finding the bridges of a graph." 1974.
- [37] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021. [Online]. Available: <http://www.gurobi.com>
- [38] G. Pataki, "Teaching integer programming formulations using the traveling salesman problem," *SIAM review*, vol. 45, no. 1, pp. 116–123, 2003.